



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TÉCNICA

**TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN**

Verificación de Competencias Académicas en base a niveles de habilidad
mediante elementos semánticos.

TRABAJO DE TITULACIÓN

AUTOR: Ortiz Sánchez, Corina Lizbeth.

DIRECTORA: González Eras, Alexandra Cristina, Ing.

LOJA – ECUADOR

2016



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2016

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniera.

Alexandra Cristiana González Eras.

DOCENTE DE TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Verificación de Competencias Académicas en base a niveles de habilidad mediante elementos semánticos realizado por Ortiz Sánchez Corina Lizbeth, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, abril de 2016

f).....

DECLARACIÓN DE AUDITORÍA Y CESIÓN DE DERECHOS

“Yo Ortiz Sánchez Corina Lizbeth declaro ser autor (a) del presente trabajo de titulación: Verificación de Competencias Académicas en base a niveles de habilidad mediante elementos semánticos, de la Titulación de Sistemas Informáticos y Computación, siendo González Eras Alexandra Cristiana directora del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f.....

Ortiz Sánchez Corina Lizbeth

1104448327

DEDICATORIA

Con inmenso amor dedico este proyecto a mis padres Luis Ortiz y Corina Sánchez, quienes con su esfuerzo, sacrificio, confianza e infinito amor han permitido que llegue a concluir con esta etapa de mi vida y por ser el motor para levantarme y seguir adelante.

A mis hermanos Willan y Luis quienes me han impulsado a lograr mis metas y por ser un ejemplo a seguir. A mis sobrinos Sol María, Scarlet, Marcela, Luis Israel y Shashita quienes han alegrado mi vida desde el inicio de su existencia.

A Dante quien con su paciencia y amor me incentivo a seguir luchando por mis metas.

Este título va con mucho cariño para ustedes, por estar presente en todo momento a lo largo de mi carrera profesional. A parte de la vida son lo mejor que Dios me regalo, los amo con todo mi ser.

Corina Lízbeth Ortiz Sánchez

AGRADECIMIENTO

Agradezco primeramente a Dios, por regalarme la vida y ser mi guía en todo momento para forjar mi futuro.

A mis padres y hermanos por confiar en mí, y por la paciencia que han tenido en ver cumplida esta meta. Mil gracias, Dios les pague y les multiplique todo lo que han hecho por mí.

A mi tutora de Tesis Ing. Alexandra González quien con sus conocimientos y experiencia ha sabido guiarme para culminar el presente trabajo de fin de titulación.

A la titulación de Sistemas Informáticos y Computación, y a mis profesores quienes me impartieron sus conocimientos en el transcurso de la carrera.

Muchas gracias a todos, por ser parte de mi formación profesional.

Corina Lizbeth Ortiz Sánchez

INDICE DE CONTENIDOS

CARÁTULA	i
PROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN	ii
DECLARACIÓN DE AUDITORÍA Y CESIÓN DE DERECHOS	iii
INDICE DE CONTENIDOS	vi
INDICE DE FIGURAS	viii
INDICE DE TABLAS	x
RESUMEN	11
ABSTRACT	12
INTRODUCCIÓN	13
CAPÍTULO I: PLANTEAMIENTO DEL PROYECTO	15
1.1 Análisis del problema	16
1.2 Causas del problema	18
1.3 Consecuencias del problema	19
1.4 Solución del problema	19
1.5 Objetivos	20
CAPÍTULO II: MARCO TEÓRICO	21
2.1 Definición de Competencia	22
2.1.1 Competencia	22
2.1.2 Competencias Académicas	23
2.2 Taxonomías de dominios de aprendizaje	24
2.2.1 Taxonomía de Marzano y Kendall	25
2.2.2 Taxonomía de Simpson	25
2.2.3 Taxonomía de Bloom	25
2.3 Ontologías	28
2.4 Lenguajes para la representación de ontologías	29
2.4.1 XML (eXtensible Markup Language)	29
2.4.2 RDF (Resource Description Framework)	29
2.4.3 RDF-S (RDF Schema)	29
2.4.4 OWL (Ontology Web Language)	29
2.4.5 SPARQL	30
2.5 Metodologías de desarrollo de ontologías	31

2.5.1	<i>Metodología Cyc</i>	31
2.5.2	<i>Metodología Uschold y King</i>	31
2.5.3	<i>Metodología Grüninger y Fox</i>	32
2.5.4	<i>Metodología Kactus</i>	32
2.5.5	<i>Metodología Methontology</i>	32
2.5.6	<i>Metodología de Noy & McGuinness</i>	32
2.6	<i>Herramientas para la construcción de ontologías</i>	34
2.6.1	<i>Protégé</i>	34
2.6.2	<i>OntoEdit</i>	35
2.6.3	<i>OntoStudio</i>	35
2.6.4	<i>Apollo</i>	35
CAPITULO III: CONSTRUCCIÓN DE LA ONTOLOGÍA DE VERIFICACIÓN DE COMPETENCIAS ACADÉMICAS		38
3.1	<i>Determinación de requerimientos</i>	39
3.2	<i>Reutilizar las ontologías y metadatos existentes</i>	40
3.3	<i>Elaboración del Modelo Conceptual</i>	41
3.3.1	<i>Definición de términos de la ontología</i>	41
3.3.2	<i>Definición de las clases y de la jerarquía</i>	42
3.3.3	<i>Definición de las propiedades de clases</i>	44
3.3.4	<i>Definición de las restricciones de las propiedades</i>	46
3.3.5	<i>Definición de axiomas formales</i>	47
3.3.6	<i>Creación de Instancias</i>	48
3.4	<i>Implementación de la Ontología verificación de competencias académicas</i>	49
3.4.1	<i>Definición de los conceptos</i>	50
3.4.2	<i>Definición de las relaciones entre clases</i>	51
3.4.3	<i>Definición de atributos</i>	51
3.4.4	<i>Definición de axiomas</i>	52
3.4.5	<i>Declaración de Instancias</i>	52
3.4.6	<i>Definición de consultas</i>	53
3.4.7	<i>Validación de la Ontología</i>	55
3.5	<i>Evaluación de versiones de la ontología</i>	58
3.6	<i>Métricas de Evaluación</i>	60
3.6.1	<i>Compleción de clases</i>	60

3.6.1.1	<i>Riqueza de clases</i>	60
3.6.1.2	<i>Riqueza de propiedades</i>	61
3.6.2	<i>Compleción de relaciones</i>	61
3.6.2.1	<i>Riqueza de relaciones</i>	61
CAPITULO IV: APLICACIÓN DE VERIFICACIÓN DE COMPETENCIAS ACADÉMICAS. ...		63
4.1	<i>Metodología de la Aplicación</i>	64
4.1.2	<i>Caso de uso</i>	66
4.1.3	<i>Modelo visto desde la aplicación web</i>	67
4.2	<i>Casos de pruebas Manuales y de la Aplicación</i>	70
4.2.1	<i>Caso Exitoso</i>	70
4.2.2	<i>Caso sobre dimensionado</i>	72
4.2.3	<i>Caso sub dimensionado</i>	73
4.2.4	<i>Caso de no concordancia</i>	74
ANÁLISIS DE RESULTADOS		76
CONCLUSIONES		77
RECOMENDACIONES		78
BIBLIOGRAFÍA		79
ANEXOS		82
ANEXO 1		83
ANEXO 2		85
ANEXO 3		94
ANEXO 4		96
ANEXO 5		98
ANEXO 5		100

INDICE DE FIGURAS

<i>Figura 1. Modelo del sistema de créditos UTPL/ECTS.</i>	17
<i>Figura 2. Taxonomía de Bloom. (Bloom's Taxonomy of learning domians).</i>	26
<i>Figura 3. Verbos y actividades de la Taxonomía de Bloom</i>	27
<i>Figura 4. Definición y Componentes de una Ontología.</i>	28
<i>Figura 5. Lenguajes de marcado de ontologías.</i>	30
<i>Figura 6. Metodología propuesta para el Modelado de Competencias Académicas.</i>	34
<i>Figura 7. Creación de Corpus.</i>	39

<i>Figura 8. Taxonomía de Verificación de competencias académicas.</i>	43
<i>Figura 9. Diagrama de relaciones binarias.</i>	45
<i>Figura 10. Modelo Conceptual de la ontología.</i>	45
<i>Figura 11. Interfaz Protégé.</i>	50
<i>Figura 12. Definición de clases en Protégé.</i>	50
<i>Figura 13. Definición de las relaciones entre clases.</i>	51
<i>Figura 14. Definición de Atributos.</i>	51
<i>Figura 15. Definición de axiomas.</i>	52
<i>Figura 16. Definición de instancias de clases.</i>	52
<i>Figura 17. Definición de instancias de atributos.</i>	53
<i>Figura 18. Consulta de materias troncales de titulación.</i>	53
<i>Figura 19. Consulta de competencias específicas.</i>	54
<i>Figura 20. Consulta verbos que pertenecen a la taxonomía de Bloom.</i>	54
<i>Figura 21. Escoger el razonador (Pellet) Protégé 3.4.8.</i>	55
<i>Figura 22. Verificación de inconsistencia de datos, Protégé 3.4.8.</i>	55
<i>Figura 23. Verificación de Taxonomía de clases, Protégé 3.4.8.</i>	56
<i>Figura 24. Verificación de inferencias de clases, Protégé 3.4.8.</i>	56
<i>Figura 25. Inferencia realizada por el razonador Protégé 3.4.8.</i>	57
<i>Figura 26. Escoger el razonador e iniciarlo. Protégé 4.3.</i>	57
<i>Figura 27. Instancias de nivel de habilidad inferidas por el razonador. Protégé 4.3.</i>	58
<i>Figura 28. Primera versión de la ontología Protégé 4.3.</i>	59
<i>Figura 29. Segunda versión de la ontología Protégé 4.3.</i>	59
<i>Figura 30. Segunda versión de la ontología.</i>	60
<i>Figura 31. Ciclo de Vida de la metodología XP.</i>	64
<i>Figura 32. Arquitectura de la aplicación.</i>	65
<i>Figura 33. Casos de uso de verificación de competencias académicas.</i>	66
<i>Figura 34. Página de inicio.</i>	67
<i>Figura 35. Seleccionar componente académico.</i>	68
<i>Figura 36. Seleccionar competencia específica.</i>	68
<i>Figura 37. Ingresar la competencia del componente.</i>	69
<i>Figura 38. Presentación de resultados.</i>	69
<i>Figura 39. Caso exitoso.</i>	71
<i>Figura 40. Caso exitoso de la aplicación.</i>	71
<i>Figura 41. Caso Sobre dimensionado.</i>	72
<i>Figura 42. Caso sobre dimensionado de la aplicación.</i>	73
<i>Figura 43. Caso Sub dimensionado Manual.</i>	74
<i>Figura 44. Caso Sub dimensionado de la aplicación.</i>	74
<i>Figura 45. Caso de no concordancia manual.</i>	75
<i>Figura 46. Caso de no concordancia de la aplicación.</i>	75
<i>Figura 47. Carpeta donde se debe copiar la plantillas.</i>	95
<i>Figura 48. Ingresar las plantillas COE.</i>	95
<i>Figura 49. Plantillas COE agregadas.</i>	96

INDICE DE TABLAS

<i>Tabla 1. Comparación de las Taxonomías.....</i>	<i>27</i>
<i>Tabla 2. Cuadro comparativo de metodologías de construcción de ontologías.....</i>	<i>33</i>
<i>Tabla 3. Análisis comparativo de las diferentes herramientas de diseño de ontologías.</i>	<i>36</i>
<i>Tabla 4. Taxonomía de Bloom.....</i>	<i>40</i>
<i>Tabla 5. Glosario de términos.....</i>	<i>41</i>
<i>Tabla 6. Lista de clases y subclases.....</i>	<i>42</i>
<i>Tabla 7. Lista de propiedades de las clases.....</i>	<i>44</i>
<i>Tabla 8. Lista de las restricciones de las propiedades.....</i>	<i>46</i>
<i>Tabla 9. Axiomas de la Ontología de verificación de competencias académicas.....</i>	<i>47</i>
<i>Tabla 10. Creación de instancias.....</i>	<i>48</i>
<i>Tabla 11. Primera versión de la ontología.....</i>	<i>58</i>
<i>Tabla 12. Segunda versión de la ontología.....</i>	<i>59</i>
<i>Tabla 13. Segunda versión de la ontología.....</i>	<i>59</i>
<i>Tabla 14. Evaluación de riqueza de clases.....</i>	<i>60</i>
<i>Tabla 15. Evaluación de riqueza de propiedades.....</i>	<i>61</i>
<i>Tabla 16. Evaluación de riqueza de relaciones.....</i>	<i>62</i>
<i>Tabla 17. Caso de prueba exitoso.....</i>	<i>70</i>
<i>Tabla 18. Caso de prueba Sobre dimensionado.....</i>	<i>72</i>
<i>Tabla 19. Caso de prueba Sub dimensionado Manual.....</i>	<i>73</i>
<i>Tabla 20. Caso de no Concordancia.....</i>	<i>74</i>
<i>Tabla 21. Competencias específicas de la titulación de Sistemas Informáticos y Computación.....</i>	<i>83</i>
<i>Tabla 22. Materias Troncales de Titulación.....</i>	<i>84</i>
<i>Tabla 23. Corpus del nivel habilidad Crear.....</i>	<i>85</i>
<i>Tabla 24. Corpus del nivel habilidad Crear.....</i>	<i>88</i>
<i>Tabla 25. Corpus del nivel habilidad Aplicar.....</i>	<i>90</i>
<i>Tabla 26. Corpus del nivel habilidad Comprender.....</i>	<i>92</i>
<i>Tabla 27. Corpus del nivel habilidad Recordar.....</i>	<i>93</i>

RESUMEN

El siguiente trabajo de titulación tiene como finalidad la Verificación de Competencias Académicas en base a niveles de habilidad mediante elementos semánticos; para ello, se desarrolló un modelo ontológico que permite la comparación del nivel de habilidad de la competencia del componente versus la competencia específica de la titulación de acuerdo a una taxonomía de habilidades predefinida, estableciendo la dimensión que ocupan las habilidades comparadas. De esta manera, el modelo permite determinar si la habilidad de la competencia de componente corresponde o no al nivel cognitivo de habilidad indicado por la competencia específica, dando la oportunidad al docente de recibir la retroalimentación correspondiente en caso de error.

Para el desarrollo del modelo ontológico que es la base principal de este proyecto se usó la metodología híbrida de Noy & McGuinness y Methontology; se toma la taxonomía de Bloom para la verificación de los niveles de habilidad. Además, para el desarrollo se usaron herramientas como: CmapTools COE (Mapa Conceptual), Protégé (Ontología), y finalmente el API Jena para la demostración del modelo ontológico.

Palabras Claves: Ontología, taxonomía, competencias académicas, modelado.

ABSTRACT

The following work aims degree Verification academic competitions based on skill levels by semantic elements; for this, an ontological model that allows comparison of the skill level of competition component versus the specific competence of the degree according to a predefined taxonomy developed skills, setting the dimension occupying skills compared. Thus, the model can determine whether the ability of competition component or not to the cognitive skill level indicated by the specific competition, giving teachers the opportunity to receive feedback on error.

To develop the ontological model which is the main basis of this project the hybrid methodology Noy & McGuinness and Methontology was used; Bloom's taxonomy for verification of skill levels is taken. CmapTools COE (Concept Map), Protégé (Ontology), and finally the Jena API to show ontological model: In addition, development tools as were used.

Keywords: Ontology, taxonomy, academic competitions, modeling.

INTRODUCCIÓN

Este proyecto tiene como objetivo la construcción de un modelo ontológico para la verificación de competencias académicas en base a los niveles de habilidad que expresa la misma. Para el modelado se tomó como punto de referencia la matriz de competencias específicas y los planes académicos de la Titulación de Sistemas Informáticos y Computación, la cual corresponde a la modalidad presencial de la UTPL.

El modelo ontológico de verificación de competencias en base a los niveles de habilidad, será de ayuda para el docente, porque le permitirá verificar si la habilidad de la competencia que ingresó está o no en el mismo nivel de habilidad de la competencia específica, ya que debe haber una coherencia entre las competencias definidas para una titulación y las competencias que el docente asigna, porque este será el indicador de que las competencias se están cumpliendo dentro de un plan de estudios. El modelo ontológico también servirá como base para la implementación en diferentes escuelas y para la construcción de un modelo que evalúe no solo el nivel de habilidad sino también el dominio de conocimiento en una competencia.

Para el desarrollo del proyecto de tesis en el capítulo I, se realiza el análisis del problema; en el cual se identifican las características principales de la gestión de las competencias específicas de la titulación y sus transferencias hacia los planes de componente académico, así como la problemática existente en cuanto a definir competencias. Al final se realiza el planteamiento del proyecto.

El Capítulo II, presenta el marco teórico de la tesis en el cual se realiza una definición de conceptos esenciales como son: competencia, competencias académicas y competencias en ciencias de la computación. Dado que el proyecto se enmarca hacia el estudio de la habilidad en la competencia, se realiza una revisión general de diferentes taxonomías de habilidad en el dominio cognitivo.

El Capítulo III, describe la elaboración del modelo conceptual y la implementación del modelo ontológico para la verificación de competencias en base a la metodología seleccionada. También se presenta una interfaz que permite al usuario la interacción con el modelo, además se muestra la evaluación de modelo ontológico, a través de indicadores como: riqueza de clases, propiedades y relaciones;

En el Capítulo IV, se detalla el desarrollo de aplicación de demostración de la ontología de verificación de competencias académicas, también casos enfocados en la detección de cuatro estados: caso exitoso, caso de sobre dimensión, caso de sub dimensión, y caso no acorde.

Finalmente, se presentan los resultados obtenidos conclusiones y recomendaciones obtenidas con el desarrollo del proyecto de tesis.

El modelo ontológico de verificación de competencias en base a los niveles de habilidad, será de mucha ayuda para el docente, ya que le facilitará la verificación de las competencias de componente que quiera asignar a una competencia específica, también servirá como base para la implementación en diferentes titulaciones de la UTPL.

Para la construcción de este modelo se hizo uso de la malla curricular de Sistemas Informáticos y Computación de la cual se extrajeron solo las materias troncales de titulación, la matriz de competencias y planes de estudios, además se elaboró un corpus de sinónimos de la taxonomía de Bloom y sin lugar a duda el uso de las herramientas fueron de ayuda para facilitar la construcción del modelado.

Al momento del desarrollo del presente trabajo de fin de titulación, se presentaron algunos inconvenientes el más importante fue la escasa información que existe en base a trabajos relacionados con la verificación de competencias académicas.

CAPÍTULO I: PLANTEAMIENTO DEL PROYECTO

En este capítulo, se realiza el análisis del problema; mediante el cual se logran conocer la función de la gestión de las competencias específicas de la titulación y su traspaso hacia los planes de componente académico, la asignación de competencias de componente por parte del docente, así como la problemática existente. Finalmente se plantea una solución al problema.

1.1 Análisis del problema.

La Universidad Técnica Particular de Loja, sigue el modelo de aprendizaje por competencias, en el cual las competencias se constituyen en la base de los procesos de enseñanza aprendizaje. Para ello, la universidad tiene definida un cierto número de competencias genéricas que contienen las actitudes (emociones y valores) que cada alumno tiene que alcanzar y desarrollar sin importar la carrera que curse, y además debe cumplir con las competencias específicas, que son propias de cada titulación (UTPL, 2012).

Hoy por hoy la tendencia de las universidades es representar o definir en los perfiles de egreso un conjunto de competencias, que el estudiante debe alcanzar al finalizar sus estudios (Soto-Fuentes, Reynaldos-Grandón, Martínez-Santana, & Jerez-Yáñez, 2014), esto hace que sea necesaria la verificación del cumplimiento de competencias a nivel de los componentes académicos.

La Universidad Técnica Particular de Loja, se acoplo a un nuevo modelo académico fundado en el Sistema Europeo de Transferencia y Acumulación de Créditos, European Credit Transfer System (ECTS)¹, el mismo que toma en cuenta el esfuerzo del estudiante por cumplir los objetivos de un programa académico que se puntualizan en los resultados del aprendizaje y de las competencias alcanzadas (UTPL, 2011).

El modelo ECTS-UTPL², fue implementado en un comienzo en la modalidad presencial en septiembre del 2007, en vista de que se consideraba a este un ambiente favorable para poder consolidar el nuevo modelo académico, más adelante en el periodo académico octubre 2009–febrero 2010 se pone en práctica en la modalidad a distancia (UTPL, 2011). El modelo del sistema de créditos ECTS es el siguiente:

¹ http://ec.europa.eu/education/ects/ects_es.htm

² <http://www.utpl.edu.ec/>

Sistemas de Créditos UTPL-ECTS modelo académico y de investigación se refleja en la adopción del sistema académico UTPL–ECTS que se desarrolla con la colaboración permanente de escuelas y CITTE's en donde profesores investigadores comparten su experiencia de vida académica y personal con sus estudiantes, los cuales aprenden, en el aula, en el CITTE's, en la vida de la universidad. Para valorar todas las actividades que realizan los estudiantes en adquirir y desarrollar las competencias necesarias para su formación, nos basamos en el modelo del sistema de créditos ECTS (UTPL, 2012, p.10).



Figura 1. Modelo del sistema de créditos UTPL/ECTS.
Fuente. (UTPL, 2012).

La UTPL dentro de su modelo académico, plantea una formación basada en competencias, en el mismo que se ha tomado como referencia el proyecto Tuning³.(UTPL, 2012). Actualmente, las competencias se las ve como acciones integrales para poder “identificar, interpretar, argumentar y resolver problemas del contexto con idoneidad y ética” (Tobón, Pimienta, & García, 2010). Estas competencias a su vez son propuestas en un plan de estudios, el cual indica que el estudiante debe adquirir estas competencias para poder aprobar una materia.

El Plan Docente de la materia, refleja los lineamientos generales bajo los cuales se va a desarrollar la materia; está diseñado en función del alumno como protagonista de su aprendizaje, en el cual se especifican elementos claves como objetivos, competencias específicas, competencias del componente, indicadores, contenidos y actividades a realizar. (UTPL, 2012, p.14).

³ <http://tuning.unideusto.org/tuningal/>

El Plan de estudios o plan docente “es un conjunto de asignaturas y créditos académicos organizados en componentes de formación, que un estudiante debe cursar para alcanzar los objetivos de formación de un programa académico”.(UCC, 2014).

Tomando en cuenta el diseño y la gestión de objetos de aprendizaje, el Plan Docente se convierte en el eje principal con el cual el docente se encarga de decidir cuando y donde crea, selecciona y organiza los diferentes elementos necesarios para una determinada materia.(Cabrera & Mas, n.d.)

Cada titulación de la UTPL crea un perfil de profesional, en el cual están inmersas las competencias que el estudiante debe desarrollar en el transcurso de su carrera, aquí los expertos escogen un cierto número de competencias genéricas y definen las competencias específicas.

La titulación de Sistemas Informáticos y Computación, crea el perfil de profesional y este a su vez es validado tanto externa como internamente la validación se basa principalmente en la competencias: La validación externa se procede realizando entrevistas al sector privado y a instituciones del estado con la finalidad de tener diferentes puntos de vista, y de esta manera determinar el peso que tiene cada una de las competencias para el desarrollo del Ingeniero de Sistemas Informáticos y Computación y poder retroalimentar las competencias existentes. La validación interna se la realiza involucrado a los docentes quienes se encargan de validar y aprobar las competencias genéricas propuestas por la EGA⁴, establecer niveles de dominio y asignar al componente académico las competencias específicas y genéricas; a partir de esto se crea la matriz de cruce de las competencias con cada una de las materias.

El docente puede establecer en el plan de componente las competencias específicas que crea conveniente en base a la matriz de cruce, es aquí cuando el docente debe asignar las competencias del componente.

1.2 Causas del problema.

En el modelo académico ECTS, los créditos no pueden ser el único indicador para valorar el desempeño del estudiante, la forma ideal es evaluar los resultados del aprendizaje y competencias que las instituciones ofrecen en los diferentes planes de estudio (TUNING, 2002 citado por Reina, 2008).

⁴ Dirección General Académica UTPL

El docente debe asignar competencias del componente a cada una de las competencias específicas en el plan de estudios, las mismas que el estudiante debe alcanzar para aprobar una materia; estas competencias son determinados por el docente en base a su conocimiento en la asignatura; pero sin embargo en ocasiones no domina el fundamento de las competencias y por eso se crean inconsistencias entre los niveles de habilidad y dominio de conocimiento propuestos para el desarrollo del componente académicos y los determinados en las competencias específicas.

Actualmente esta verificación se la realiza en el plan de estudios, a través de un proceso de revisión por pares en base a una rúbrica de indicadores, pero el éxito de este proceso depende en gran medida de la experiencia del par en el manejo de competencias.

1.3 Consecuencias del problema.

Debido a que no existe una correspondencia clara para el docente en la definición de competencias, no se llega a cumplir a cabalidad las competencias específicas porque al proponer niveles de habilidad diferentes no se establecen las actividades necesarias para alcanzar las competencias específicas. Pueden existir algunos casos que son:

- Sobre dimensión: cuando el nivel de habilidad planteado por el docente este en un nivel superior al que propone la competencia específica.
- Sub dimensión: este caso es cuando el nivel de habilidad expuesto por el docente en el plan de estudios es inferior a la competencia específica.
- No concordancia: cuando el nivel de habilidad que plantea el docente no pertenece a una habilidad.

Lo mismo puede pasar con el dominio del conocimiento el cual necesita de un estudio más exhaustivo.

1.4 Solución del problema.

Como solución del problema, se plantea la creación de un modelo ontológico para la verificación de competencias; concretamente cuando se intenta definir el nivel de habilidad que debe estar inmersa en una competencia.

A partir de esto, se propone la verificación de las habilidades entre las competencias, con el propósito de establecer el indicador de cumplimiento de éstas en relación a las competencias específicas del perfil versus las competencias del componente académico en

base a una taxonomía de aprendizaje. Para ello se usará niveles de habilidades como punto de referencia y reglas que definen las relaciones de las habilidades.

En el caso de producirse la sobre dimensión, sub dimensión, o no existencia de la habilidad, se debe retroalimentar al docente con habilidades o verbos de acción que puedan ser usados para describir la competencia específica que se quiere definir a través de las competencias del componente.

1.5 Objetivos

Objetivo General:

- Establecer un modelo ontológico con el cual se logre la verificación de los niveles de habilidad de las competencias, con el propósito de establecer el indicador de cumplimiento de éstas en relación a las competencias específicas del perfil y las competencias del componente académico.

Objetivos Específicos:

- Investigar metodologías y herramientas necesarias para la elaboración del modelado.
- Definir la metodología y herramientas a utilizar para modelar las competencias académicas.
- Implementar el modelo ontológico que nos permita la representación de las competencias académicas.
- Realizar la validación y evaluación de la ontología.

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se detalla la definición de: competencia, competencias académicas. En vista de que el proyecto centra su estudio en los verbos de acción o habilidad que se expresan en la competencia, se realiza una revisión general de diferentes taxonomías de habilidad en el dominio cognitivo. Además se mencionan algunas metodologías, herramientas y lenguajes para la construcción de modelos ontológicos realizando cuadros comparativos para definir cuál de ellas se usará para el proyecto.

2.1 Definición de Competencia

Para hablar de competencias académicas primero se debe tener clara la definición de competencia, la cual se ha desarrollado a partir de la década de los veinte en las diferentes partes del mundo, pero ha prestado su mayor atención en los países de Alemania, Inglaterra, Estados Unidos, Australia y Argentina (Huerta, Pérez & Castellanos, 2000 citado por (Charria Ortiz, Sarsosa Prowesk, Uribe Rodríguez, López Lesmes, & Arenas Ortiz, 2011). En la revisión de la bibliografía se han encontrado algunas definiciones de competencias de diferentes autores.

2.1.1 Competencia

(Chomsky, 1957) y (Hymes, 1972); nombrado en (Charria Ortiz et al., 2011) analiza a las competencias desde un punto de vista lingüístico en donde el concepto de competencia es separado en competencia que es el conocimiento y dominio que el hablante u oyente tiene de su lenguaje; y actuación que es el uso que se da al lenguaje en situaciones específicas. Desde una perspectiva sociolingüístico. Además menciona que según su origen gramatical “la competencia nace del verbo ‘competere’, que significa ‘pertener a’ o ‘incumbir’, lo que da lugar al sustantivo competencia, y al adjetivo competente para indicar apto o adecuado”.

En la década actual, (Lévy-Leboyer, 2000) manifiesta que la persona tiene un sinnúmero de conductas por las cuales crea competencias que se manifiestan dependiendo la situación en la que se encuentren. En cambio (Sternberg, 2000) enunciado en (Charria Ortiz et al., 2011), entiende la competencia como un conocimiento implícito que, más que ser una inteligencia académica o general, está más bien relacionado con la habilidad para resolver problemas específicos de la vida diaria. (Alzina, 2003) y (Muñoz, Quintero y Munévar, 2001) citado por (Charria Ortiz et al., 2011) menciona que la competencia es el conjunto de conocimientos, habilidades y actitudes que se aplican en una tarea específica. Otra definición es la de (Maldonado, 2006) publicado por (Rafael, 2011) en donde afirma que “La

competencia es la integración dinámica del saber ser, saber conocer y saber hacer, que impulsa un desempeño idóneo y superior en la solución de situaciones y problemas reales dentro del contexto cambiante donde se desarrolla la vida de la persona humana”(p.4).

(Paquette, 2007) menciona que “las competencias son declaraciones que alguien, y más generalmente algún recurso, puedan demostrar la aplicación de una habilidad genérica para algunos conocimientos, con un cierto grado de rendimiento”.

En cuanto a lo manifestado anteriormente por los autores acerca del concepto o definición de competencias se resume que competencia es el conjunto de conocimientos, enfoques, metodologías, actitudes, habilidades y valores; las mismas que son dadas por el desempeño eficiente en soluciones de problemas dados en un determinado contexto o dominio particular.

2.1.2 Competencias Académicas

(Charria Ortiz et al., 2011) asevera que las competencias académicas están estrechamente relacionadas con el aprendizaje escolar ya que empiezan a desarrollarse desde los primeros años de vida; y que las competencias académicas “están asociadas a conocimientos fundamentales que se adquieren en la formación general” (p.141), lo que nos aclara que las competencias académicas no solo se las adquiere educación básica sino que son aprendidas en el transcurso de la vida y las clasifica en habilidades básicas: hablar, escuchar, leer, comprender, razonar, hablar, escuchar.

(Salas, 1996) proyecta que las competencias académicas son un conjunto de capacidades pero en un área específica, las mismas que son desarrolladas a través de una educación formal, citados por (Charria Ortiz et al., 2011).

Otras definiciones basadas en función de la responsabilidad de la educación y de la persona que está en formación profesional. Es la de (Pinilla, 1999), quien plantea que las competencias académicas son el grupo de capacidades que se manifiesta a través de un proceso de aprendizaje continuo, con el cual se lleva a la persona responsable a ser competente para realizar múltiples acciones tales como: sociales, cognitivas, culturales, afectivas, laborales y productivas; con las cuales puede mostrar como evidencia su habilidad para resolver los diferentes tipos de problemas en un entorno específico. (Monreal, citado por Arias, González, Gómez., 2011) afirma que el “aprendizaje por competencias es el

conjunto de conocimientos, actitudes y destrezas necesarias para desempeñar una ocupación dada” (p.1). Citados por (Charria Ortiz et al., 2011).

(Aquilino et al., n.d.) Menciona que “a partir de la definición del concepto de competencia surgen catalogaciones de las mismas: competencias específicas, vinculadas a cada área de estudio; competencias genéricas, transversales a las titulaciones, y relacionadas con la formación integral de la persona”. (p.3)

(Cideal, 2012) Indica que “los elementos de competencia se redactan como una oración, siguiendo la regla de iniciar con un verbo en infinitivo preferiblemente; a continuación describir el objeto y; finalmente, aunque no es obligatorio en todos los casos, incluir la condición que debe tener la acción sobre el objeto” (p.45).

Las definiciones de competencias coinciden mucho; por lo que se concluye que competencia académica es el conjunto de capacidades y habilidades que se desarrollan durante todo el proceso de aprendizaje para estar aptos al momento de resolver un problema específico.

La competencia como lo menciona Cideal, está formada por un verbo de acción en infinitivo, el mismo que vendría a ser la habilidad, la cual puede ser evaluada en base a taxonomías de aprendizaje, ya que estas expresan niveles de habilidad y un sinnúmero de verbos relacionados a ellas.

2.2 Taxonomías de dominios de aprendizaje.

Hoy en día existen un sin número de taxonomías en los diferentes dominios que son: *psicomotor* habilidades físicas o motoras, *afectivo* habilidades actitudinales y valores que la persona adecua a las situaciones que vive, *cognitivo* desempeño del conocimiento y uso de las habilidades intelectuales o de pensamiento, *meta-cognitivo* implantar las metas a lograr a y diseñar estrategias para su logro, interno relación mutua entre las creencias y metas. Entre las taxonomías que se basan en el aprendizaje tenemos: Marzano y Kendall, Simpson y Bloom.

2.2.1 Taxonomía de Marzano y Kendall.

Esta taxonomía propuesta por Robert Marzano y John Kendall se basa en la idea de Benjamín Bloom en 1956. Pero sin embargo presenta algunas variaciones entre ellas es el problema al momento de ejecutar un proceso mental, el mismo que a su vez se ajusta a dos factores: primero, la complejidad inherente del proceso en términos de los pasos o fases, lo que implica que debe existir una sincronía de pasos; y segundo el nivel de familiaridad que uno tiene con respecto al proceso, el cual depende de la experiencia que tenga la persona.

Este modelo está compuesto por tres sistemas mentales: *el interno, el meta-cognitivo y el cognitivo* y toma como un cuarto componente al conocimiento. El sistema interno maneja la correspondencia entre las metas y las creencias que tiene la persona en el transcurso de su aprendizaje, el meta-cognitivo plantea las metas que deben ser alcanzadas en base a las tareas propuestas y el cognitivo es importante para terminar las tareas planteadas a través de acciones como: comparar, analizar, calificar, etc. (Gallardo, 2009).

2.2.2 Taxonomía de Simpson.

Esta taxonomía interpretada por Elizabeth Simpson en 1966, se centra específicamente en uno de los dominios de los objetos de aprendizaje, que es el dominio *psicomotor*. Esta taxonomía se encuentra conformada por siete niveles los cuales son:

- **Precepción:** Percibir los objetos a través de los sentidos.
- **Preparación:** Prepararse para una acción física específica
- **Respuesta dirigida:** Forma de actuar ante una actividad requerida.
- **Mecanismo:** Alcanzar a través de aprendizaje un cierto nivel de desempeño en cuanto a su respuesta usual.
- **Respuesta abierta compleja:** Desempeñar un acto motor considerado complejo de acuerdo al tipo de movimiento requerido.
- **Adaptación:** Aplicar actividades motoras para satisfacer exigencias de nuevas situaciones problemáticas.
- **Producción:** Crear nuevas acciones motoras o modos de manipular los materiales.

2.2.3 Taxonomía de Bloom.

Benjamín Bloom en el año de 1956, creó su taxonomía de Objetivos Educativos; la misma que fue de mucha ayuda para organizar y comprender el proceso de aprendizaje. Más

adelante en los años 90, Lorin Anderson y David R. Krathwohl, dos estudiantes de Bloom realizaron la revisión de la taxonomía y la publicaron en diciembre del 2000.

Los cambios que se realizaron en la revisión de Bloom fue de sustantivos que se manejaban anteriormente a los verbos que describen una acción. Además otro modificación que realizaron fue cambiar la síntesis con un criterio más amplio como el de crear, tomando en cuenta que la síntesis como tal es una creación, también se hizo una cambio en el orden de los niveles y además se cambió conocimiento por recordar. (Churches, n.d.).

Para que quede un poco más clara la explicación anterior, se muestra la taxonomía de Bloom y la revisión de Bloom en la figura 2.



Figura 2. Taxonomía de Bloom. (Bloom's Taxonomy of learning domians).
Fuente: (Churches, n.d.).

(Perez & Sepulved, 2008) Señala que la taxonomía de Bloom, enfatiza su estudio en los objetivos educativos y, plantea que las actividades a realizar pueden estar dentro de tres dominios psicológicos: cognitivo, afectivo, o psicomotor. El dominio cognitivo trata de la habilidad para procesar y usar la información en un determinado contexto, dominio afectivo son las cualidades y las sensaciones que se pueden lograr en el proceso de aprendizaje y el dominio psicomotor involucra las destrezas físicas.

La taxonomía de Bloom, cuenta con un conjunto de verbos en cada uno de los niveles, y además actividades relacionadas a estos; que se muestra en la siguiente figura 3.

La taxonomía de Bloom maneja los niveles de habilidad en el dominio cognitivo, los mismos que están formados por verbos relacionados que ayudan a mayor retroalimentación de habilidades, además ha sido imprentado en algunas ontologías de competencias lo cual verifica que uso es pertinente dentro del problema a resolver de verificación de competencias académicas en base a niveles de habilidad.

2.3 Ontologías

(Fermoso, Salvador, & Sicilia, 2008) “Señala que una ontología es una jerarquía de conceptos con atributos y relaciones, que define una terminología consensuada para definir redes semánticas de unidades de información interrelacionadas” (p.3).

Una ontología tiene un vocabulario conformado por clases, propiedades o atributos, relaciones e instancias, el mismo que ayuda compartir la información en base a un dominio específico (Fermoso et al., 2008).

Las ontologías son una conceptualización de un dominio, es decir, un modelo abstracto de un área en particular, en el que, se describen conceptos y relaciones entre los mismos, por lo que son generalmente conocidas como procedimientos de representación del conocimiento. El concepto más difundido es el (Gruber, 1993), quien hace referencia a las ontologías como: “Una ontología es la especificación de una conceptualización”, es decir, un modelo conceptual donde se especifican los conceptos y sus relaciones.

Según (Guzmán, Bonilla, & Durley, 2012), las ontologías benefician la comunicación entre personas, organizaciones y aplicaciones porque proporcionan una conocimiento común de un dominio, de esta forma se pueden eliminar las confusiones tanto conceptuales como terminológicas.

Definición y componentes de la Ontologías representado en un mapa conceptual en la figura 4.



Figura 4. Definición y Componentes de una Ontología.
Fuente: (Silva-Sprock et al., 2007).

(Gascueña, Fernández-caballero, & González, 2005) Menciona que las ontologías es una de las mejores herramientas para explicar los conceptos en un determinado dominio dentro de la Web Semántica.

Se usará ontologías en el proyecto, porque tienen mayor flexibilidad para describir el conocimiento, permite la creación de restricciones y axiomas, y sobretodo porque son fácil de consumir al momento de querer ser reutilizadas.

2.4 Lenguajes para la representación de ontologías

Existen algunos lenguajes de representación de ontologías, A continuación se presentan algunos de ellos:

2.4.1 XML (eXtensible Markup Language)

XML es un lenguaje de marcas de etiquetas, el cual se lo utiliza para almacenar datos de forma legible, este proviene del lenguaje SGML. Un documento XML permite estructurar documento de forma jerárquica para luego ser almacenados.

2.4.2 RDF (Resource Description Framework)

RDF consiente la representación de datos en la web, además es un modelo de datos que puede ser representado mediante XML a través de una semántica básica. RDF está formado en tripletas (sujeto, predicado y objeto).

2.4.3 RDF-S (RDF Schema)

RDFS es una extensión de RDF, el cual es un lenguaje primitivo de ontologías que facilita los elementos básicos para la descripción de vocabularios. RDFS tiene la misma estructura y sintaxis que RDF, pero define nuevos conceptos introduciendo el uso de propiedades y clases dentro de la descripción de los enunciados.

2.4.4 OWL (Ontology Web Language)

OWL es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. Permite definir el significado de términos en vocabularios y las relaciones entre ontologías. OWL expresa mejor el significado y la semántica que los otros lenguajes nombrados, y, de este modo, OWL va más allá de estos lenguajes en su capacidad para

representar contenido interpretable por un ordenador en la Web. Además puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre los términos, a esta representación de términos y sus interrelaciones se denominan ontología. OWL es el lenguaje extendido y para esto ha tomado como base lenguajes ya existentes como son XML, XML Schema, RDF y RDF Schema, además agrega otras características para crear ontologías: que son: definición de clases, relaciones, propiedades, instancias, restricciones, cardinalidad, etc.

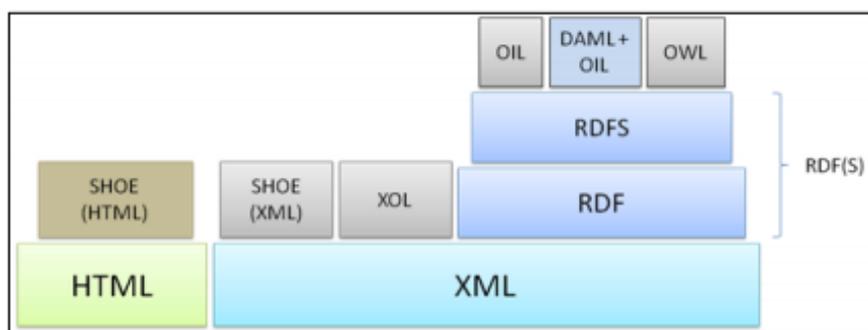


Figura 5. Lenguajes de marcado de ontologías.
Fuente: (Ñauta & Escandón, 2011)(p.62).

2.4.5 SPARQL

SPARQL es un lenguaje estándar para la consulta de grafos RDF. En un principio SPARQL únicamente incorpora funciones para la recuperación sentencias RDF, sin embargo algunas propuestas también incluyen operaciones para el mantenimiento (creación, modificación y borrado) de datos. Actualmente esta tecnología está siendo considerada como una recomendación del W3C para crear un lenguaje de consulta dentro de la Web semántica

(Fermoso et al., 2008) “El lenguaje OWL ha sido diseñado para que las aplicaciones puedan procesar e integrar automáticamente el contenido de la información en la Web, en vez de simplemente presentarla para “consumo humano”” (p.3).

Después del estudio realizado a los lenguajes para la representación de ontologías, se cree conveniente el uso de OWL, ya que tienen mayor capacidad de expresión; y con esto queremos lograr la representación del modelo ontológico a través de clases, relaciones, propiedades e individuos, los cuales serán usados para definir el dominio escogido.

Se introduce además el tema de SPARQL, ya que es un lenguaje para consultas dentro de la web semántica, con el mismo que se podrá extraer la información necesaria para presentar los resultados.

2.5 Metodologías de desarrollo de ontologías.

En la investigación realizada para la verificación de Competencias Académicas, he encontrado algunas metodologías para el desarrollo de ontologías entre ellas tenemos: Cyc, Uschold y King, Grüninger y Fox, Kactus, Methontology, Noy&McGuinness.

2.5.1 Metodología Cyc.

Cyc nace el año de 1984 en la Corporación de Tecnología en Computación y Microelectrónica, como un proyecto de inteligencia artificial que busca la construcción de una ontología comprensible para habilitar el razonamiento humano. Cyc tiene una base de conocimiento propia, pero también existe OpenCyc que es una versión libre.

Los pasos para la construcción de la ontología usando esta metodología son:

- Extraer manualmente el conocimiento común que está implícito en diferentes fuentes
- Usar herramientas de procesamiento de lenguaje natural o aprendizaje computacional. (Cartuche, 2008)

2.5.2 Metodología Uschold y King.

(Silva-Sprock et al., 2007); menciona que esta metodología permite la creación de ontologías en base a otras ya existentes. Recomienda los siguientes pasos:

- Identificar el propósito y el alcance.
- Construir la Ontología.
- Capturar.
- Codificar.
- Integrar Ontologías existentes.
- Evaluación – Documentación.
- Pautas iniciales para el diseño de Ontologías. (p.3)

2.5.3 Metodología Grüninger y Fox.

En esta metodología se proponen los siguientes pasos:

1. Definición de los escenarios motivadores.
2. Formulación de preguntas en lenguaje natural.
3. Especificación de la terminología.
4. Formalización de las interrogantes.
5. Especificación de axiomas formales
6. Verificación de la ontología. (Cartuche, 2008).

2.5.4 Metodología Kactus.

(Guzmán et al., 2012) Kactus es una metodología con la cual se construye la ontología sobre una base de conocimiento por medio de un proceso de abstracción. Kactus define los siguientes pasos:

- Especificación de la aplicación
- Diseño preliminar basado en categorías ontológicas toplevel relevantes
- Refinamiento y estructuración de la ontología.(p.4)

2.5.5 Metodología Methontology.

(Corcho, Fernandez, Gomez, & Lopez-Cima, 2005) Menciona que METHONTOLOGY ha sido desarrollada por el Grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid. Esta metodología permite construir ontologías en el nivel de conocimientos, METHONTOLOGY proporciona guías sobre cómo llevar a cabo el desarrollo de la ontología a través de las actividades:

- Especificación.
- Conceptualización.
- Formalización.
- Implementación.
- Mantenimiento.

2.5.6 Metodología de Noy & McGuinness.

Metodología creada por la Universidad de Stanford, la cual cumple los siguientes pasos:

- Determinar el dominio y alcance de la Ontología.

- Considerar la reutilización de Ontologías existentes.
- Enumerar términos importantes para la Ontología.
- Definir las clases y las jerarquías de clases.
- Definir las propiedades de las clases: slots.
- Definir las facetas de los slots - Crear instancias.

Tabla 2. Cuadro comparativo de metodologías de construcción de ontologías.

Características Metodologías	Ciclo de vida propuesto	Herramientas de soporte	Modelado	Nivel de Abstracción	Uso de ontología base
Cyc	Secuencial	No especificada	Extracción manual	Alto	No
Uschold y King	No especificada	No especificada	Identificación directa de conceptos	Normal	No
Grüninger y Fox	Secuencial	No especificada	Preguntas de competencia	Alto	No
Kactus					
Methontology	Secuencial con prototipos de desarrollo y actividades de administración y apoyo paralelo al desarrollo	ODE WebODE OntoEdit Protégé 2000	Identificación directa de conceptos y Organización taxonómica.	Elevado	NO
Noy & McGuinness.	Secuencial con prototipos de desarrollo	Protégé 2000	Identificación de conceptos, relaciones, propiedades, individuos	Elevado	No

Fuente: Elaboración propia

Para la definición y construcción de la verificación de competencias académicas, primeramente se realizó un estudio comparativo de algunas metodologías de diseño de ontologías, de donde se concluye que para este proyecto se hará uso de una metodología híbrida, la misma que se componga de aspectos que sean relevantes que para efectos del proyecto. (Alvarado, 2010); La Metodología de Noy & McGuinness y Methontology, pone principal énfasis en la construcción de un modelo conceptual robusto y en la determinación clara y concisa de los requerimientos de la ontología a construirse. Los pasos a seguir en esta metodología son:

1. Determinar los requerimientos de la ontología.
2. Reutilizar las ontologías o metadatos existentes.
3. Elaboración del modelo conceptual.

4. Implementación del modelo conceptual.
5. Evaluación de la ontología.

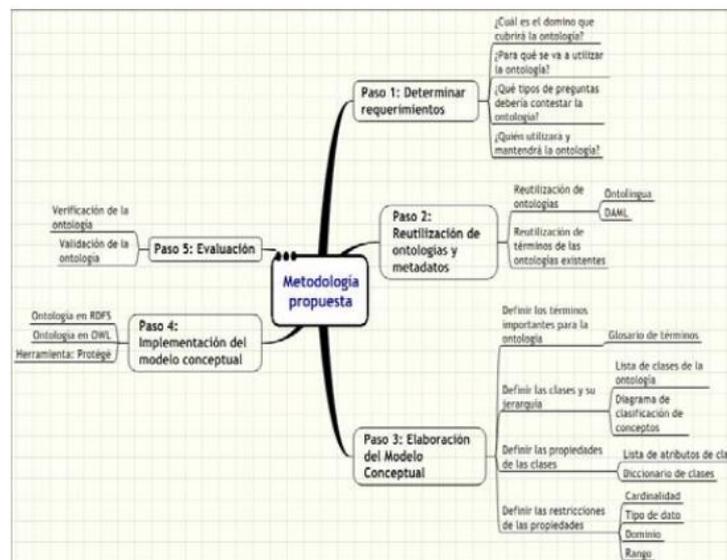


Figura 6. Metodología propuesta para el Modelado de Competencias Académicas.
Fuente: (Sarango, Alvarado, & Villa, n.d.).

2.6 Herramientas para la construcción de ontologías.

García (2008), menciona que las herramientas ayudan a definir las ontologías en un determinado lenguaje, además el uso de editores facilita la tarea de definición de ontologías, para el desarrollo distribuido de ontologías, se pueden usar las siguientes herramientas existentes: Apollo, OntoEdit, OntoStudio, y Protégé, Ontolingua, ect.

2.6.1 Protégé

Protégé es una herramienta de software libre y de código abierto, desarrollada por la Universidad de Stanford. Su plataforma soporta dos formas de modelar ontologías; una es mediante Protégé-Frames, y la otra en el programa usando los editores OWL.

Esta herramienta está basada en Java, y cuenta con varias versiones que han sido creadas para la construcción de ontologías, además sus extensiones ayudan al que el programa sea manejable para el desarrollo de aplicaciones. Protégé tiene una arquitectura cliente-servidor, una ontología en Protégé puede ser exportada en los siguientes formatos RDF, OWL y XML Schema; e importada en XML(S), RDF(S), OWL, RDF,UML, XML, Excel, BioPortal and DataMaster.

2.6.2 *OntoEdit.*

OntoEdit es una herramienta de código abierto, esta herramienta de edición de ontologías permite su desarrollo mediante un entorno gráfico y un entorno Web.

(Ñauta & Escandón, 2011) menciona que “existen varias versiones del programa, desde la 2.0 hasta la 2.6. Cada una de estas con una versión libre y una profesional. Algunas de las características más relevantes de la última versión de OntoEdit son: Importar las estructuras del directorio, importar tablas de Excel, construir reglas gráficas, y; visualizar y editar ontologías en un gráfico” (p.58).

2.6.3 *OntoStudio.*

OntoStudio es una herramienta creado por la compañía Ontoprise, basada en Eclipse y es un entorno de modelado comercial más extendido para el desarrollo y mantenimiento de ontologías, se puede acceder a una licencia de solo 3 meses pero para uso no comercial.

Se destaca por ser una herramienta muy intuitiva, tiene una arquitectura independiente y soporta los siguientes lenguajes: OWL, RDF(S) y F-Logic. (Camacho, 2008) menciona que “su motor de inferencia integrado OntoBroker, OntoStudio permite el procesamiento de ontologías así como de la lógica que ellas representan” (p.16). OntoBroker fue creado para facilitar el proceso de consultas.

2.6.4 *Apollo.*

Apollo fue creado para ser una aplicación de fácil de uso, es un herramienta de software libre tiene una arquitectura independiente, al igual que el resto de herramientas también basa su creación en clases, instancias, funciones, relaciones, etc.

Apollo no está vinculado a ningún lenguaje de representación del conocimiento y puede ser adaptado a través de plug-ins para soportar diferentes formatos de almacenamiento, su modelo interno está construido como un sistema de trama según el modelo interno del protocolo OKBC. Apollo hace una comprobación de coherencia completa durante la edición.

A continuación se presenta un cuadro comparativo en base a las herramientas detalladas anteriormente.

.

Tabla 3. Análisis comparativo de las diferentes herramientas de diseño de ontologías.

CARACTERISTICAS	PROTÉGÉ	ONTOSTUDIO	ONTOEDIT	APOLLO
Arquitectura	Cliente/Servidor Independiente	Independiente	Cliente/Servidor	Independiente
Licencia	Libre	Pagado	Libre	Libre
Motores de Inferencia	Si RACER, FACT, FACT++, F-logic and Pellet	No	Si	No
Extensibilidad	Plug-ins	Plug-ins		Plug-ins
Almacenamiento	File and DBMS (JDBC)	File	JDBC	DBMS
Acoplar con herramientas	PROMPT, OKBC, JESS, FaCT and Jena	No	Si	OKBC
Lenguajes de Axiomas	Si	Si	Si	No
Lenguajes de importación	XML(S), RDF(S), OWL, (RDF,UML, XML)backend, Excel, BioPortal and DataMaster	XML(S), OWL, RDF(S), UML Diagram, Database schemas (Oracle,MSSQL, DB2,MySQL), Outlook, file system Metadata and Remote OntoBroker.	DAML+DOIL , Directory, Excel, Flogic, RDF(s)	Apollo Meta language
Lenguaje de exportación	RDF(S), XML(S) , OWL, MetaAnalysis, Queries	RDF (S), UML, OWL, XML.	DAML+DOIL	OCML and CLOS

Fuente: Elaboración propia citando a (Alatrish, 2012), (Fierro, n.d.).

Para implementación de la ontología se eligió la herramienta Protégé por su portabilidad entre diversas plataformas, su extenso uso y abundante documentación. Esta herramienta posee una interfaz gráfica que facilita el desarrollo de la ontología sin tener que preocuparse por la sintaxis del lenguaje de definición de ontologías escogido (OWL).

Y para la construcción del DEMO, con el cual se demostrará que el mecanismo de verificación de competencias funciona se usará JENA.

(León, Largo, & Tuza, 2014); señala que JENA Apache es un framework Java para construir aplicaciones basadas en ontologías. Jena se desarrolló en HP Labs en el 2000, en 2009 HP cedió el proyecto a la fundación Apache que decidió patrocinarla en noviembre de 2010. Su Arquitectura incluye:

- API para trabajar (leer, procesar, escribir) ontologías RDF y OWL.

- Motor de inferencia para razonar sobre ontologías RDF y OWL.
- Estrategias de almacenamiento flexible para almacenar tripletas RDF en memoria o fichero.
- Motor de queries compatible con especificación SPARQL.

**CAPITULO III: CONSTRUCCIÓN DE LA ONTOLOGÍA DE VERIFICACIÓN DE
COMPETENCIAS ACADÉMICAS**

En este Capítulo, se realizará la construcción del modelo en base a la metodología escogida, primero se revisará el dominio que cubrirá la ontología, a partir de esto se elaborará el mapa conceptual para luego crear la ontología en la herramienta Protégé, el ontología se definirán las clases, relaciones, atributos, axiomas, individuos, ect.

3.1 Determinación de requerimientos.

¿**Qué** dominio cubrirá la ontología? Verificación de Competencias Académicas se modelará el manejo de competencias específicas de la titulación versus las competencias del componente en un plan de estudios, en la Titulación de Sistemas Informáticos y Computación de la UTPL.

¿**Para** qué se va emplear la ontología? Esta ontología se la empleará para realizar la verificación de las competencias del componente académico en cuanto a las competencias específicas del perfil propuestas para cada asignatura, el cual se basará en verificar la concordancia de los niveles de habilidad entre ellas, mediante elementos semánticos, tomando como referencia la taxonomía de Bloom.

La taxonomía de BLOOM, está conformada por 6 niveles de habilidad que son: recordar, comprender, aplicar, analizar, evaluar y crear; los mismo que a su vez tienen verbos relacionados, los cuales pueden ser usados en representación a cada nivel respectivamente; además vale aclarar que hay verbos que se repiten en dos o más niveles de habilidad. Como aportación propia, se cree conveniente agregar los sinónimos de los verbos relacionados, y de los verbos que forman parte de los niveles de habilidad dela taxonomía de Bloom, en vista de que las personas expresamos de diferentes maneras algo en particular, para esto he realizado consultas en diccionarios para la construcción del corpus de sinónimos, ver Anexo 2.

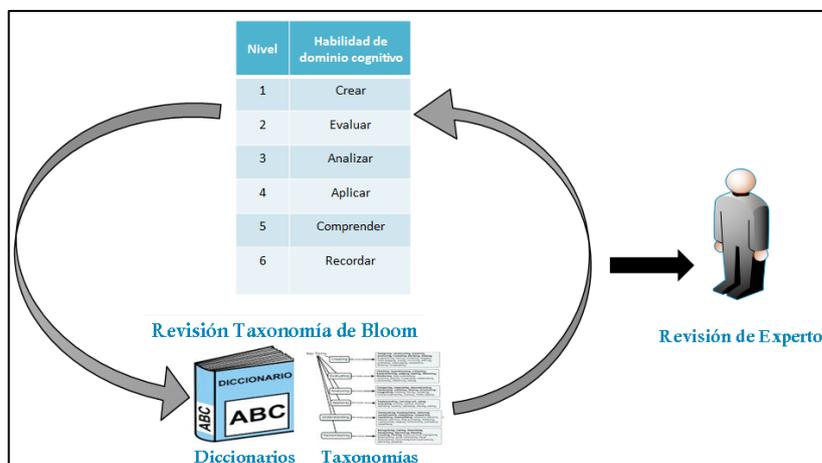


Figura 7. Creación de Corpus.
Fuente: Elaboración propia.

(Farias et al., 2010); señalan que los mecanismos de evaluación del aprendizaje son el medio que ayuda a medir la competencias desarrolladas por el estudiante. De ahí la ocupación en diseñar un sistema de evaluación basado en la taxonomía de Bloom que exige que el participante de un proceso educativo demuestre las habilidades pertinentes.

¿**Qué** preguntas deberá contestar la ontología? En el dominio de Competencias Académicas, las preguntas de mayor importancia son: ¿Las competencias del componente pertenece al mismo nivel de la taxonomía de las competencias específicas?, ¿Los verbos claves de las habilidades se encuentran dentro de los verbos relacionados o sinónimos de las habilidades de las competencias? ¿Cuáles son los verbos que se pueden usar para los niveles de habilidad de las competencias específicas?

¿**Quién** utilizará y mantendrá la Ontología? Los usuarios permitidos en esta ontología son los docentes de la Titulación de Sistemas Informáticos y Computación de la modalidad presencial de la UTPL. Las personas delegadas de actualizar y mantener la ontología serán docentes encargados de cada una de las asignaturas y los coordinadores de la titulación.

3.2 Reutilizar las ontologías y metadatos existentes.

Para la construcción de la ontología de verificación de competencias académicas no reutilizará ninguna ontología, en vista de que la información existente es muy reducida en este.

Sin embargo si se hará uso de los niveles de habilidad del dominio cognitivo de la taxonomía de Bloom que son los siguientes:

Tabla 4. Taxonomía de Bloom

Taxonomía de Bloom
Crear
Evaluar
Analizar
Aplicar
Comprender
Recordar

Fuente: Elaboración propia.

3.3 Elaboración del Modelo Conceptual.

3.3.1 Definición de términos de la ontología.

En este paso primeramente se va a listar todos los términos que tienen relación con el dominio de competencias académicas, los mismos que se han ido eligiendo de acorde al estudio realizado en base al dominio estudiado. En la **tabla 5** se realizará una breve descripción de cada uno de los términos que se van a utilizar para la elaboración del mapa conceptual.

Tabla 5. Glosario de términos.

Nombre	Descripción
Universidad	Institución donde se está realizando el manejo de competencias académicas en este caso la UTPL.
Modalidad	Nombre de la modalidad de estudio con las que cuenta la UTPL, las mismas que son la modalidad abierta y la modalidad presencial que forman parte de la Universidad.
Áreas	Son las áreas de estudio que forman parte de la UTPL, entre ellas esta: Área Técnica, Biológica, SocioHumanística y Administrativa.
Departamentos Área Técnica	Departamentos que conforman el área técnica, entre ellos tenemos: el departamento de Ciencias de la Computación y Electrónica.
Titulación	Titulación que forman parte de cada departamento como: Sistemas Informáticos y Computación
Malla Curricular.	El mismo que contiene las materias o asignaturas que un estudiante debe cursar y aprobar para obtener una titulación.
Materias	Son las materias que se dictan en la UTPL.
Competencias Específicas del Perfil.	Competencias asignadas por un grupo de expertos a la titulación.
Competencia del componente Académico.	Son las competencias que el docente asigna dentro de un plan docente, las cuales el estudiante debe cumplir para medir su rendimiento.
Plan Docente	Se encuentra conformado por las competencias específicas del perfil y las competencias del componente académico.
Nivel Habilidad Complejidad	Niveles de habilidad de la taxonomía de Bloom.
Verbos Relacionados	Son los verbos que forman parte de cada nivel de habilidad complejidad.
Sinónimos	Son los sinónimos de los verbos relacionados.
tieneMod	Se refiere a que la UTPL tiene Modalidad.
perteneceU	Se refiere a que la modalidad pertenece a la UTPL.
tienenA	Se refiere a que la modalidad presencial y abierta tienen Áreas.
formadaDep	Se refiere a que las áreas están formadas por departamentos.
otorgaTi	Se refiere a que los departamentos otorgan titulación.

tieneMalla	Se refiere a que la titulación tiene Malla Curricular.
tieneCE	Se refiere a que la titulación tiene competencias específicas del perfil.
conformadaMate	Se refiere a que la malla curricular está conformada por materias.
pertenecenPD	Se refiere a que las competencias específicas del perfil pertenecen al plan docente.
evaluanPorCC	Se refiere a que las competencias específicas del perfil se evalúan por las Competencias del componente académico
ceContieneH	Se refiere a que las competencias específicas del perfil contienen habilidades.
ccContieneH	Se refiere a que las competencias del componente académico contienen habilidades
poseePD	Se refiere a que las materias posee plan docente
poseeCC	Se refiere a que el plan docente posee competencias del componente académico

Fuente: Elaboración propia.

3.3.2 Definición de las clases y de la jerarquía.

Para la definición de clases y su jerarquía, se toma en cuenta los términos que se describieron en el glosario y a partir de estos se escogen los términos que formaran parte de las clases y subclases.

En la siguiente tabla se puede ver la lista de clases y subclases que pertenecen al modelamiento de verificación de competencias académicas.

Tabla 6. Lista de clases y subclases.

Nombre de clase	Nombre de Subclases
UTPL.	
Modalidad.	Abierta. Distancia.
Áreas.	Técnica. Administrativa. SocioHumanística. Biológica.
Departamentos Área Técnica.	Geología y minas e Ingeniería Civil. Arquitectura y Artes. Ciencias de la Computación y Electrónica.
Titulación.	Sistemas Informáticos y Computación. Ingeniería Informática.
Malla Curricular.	
Materias.	Troncales de Titulación. Formación Básica. Libre Configuración. Complementarias. Genéricos de Titulación. Gestión Productiva.

Competencias Específicas.	
Competencias del Componente Académico.	
Plan Docente.	
Nivel Habilidad Complejidad.	Crear. Evaluar. Analizar. Aplicar. Comprender. Recordar.

Fuente: Elaboración propia.

Una ontología no se necesita especificar más de lo necesario, a continuación en la **figura 8** se muestra la jerarquía de clases de la ontología de verificación de competencias.



Figura 8. Taxonomía de Verificación de competencias académicas.
Fuente: Elaboración Propia.

3.3.3 Definición de las propiedades de clases.

Una vez que ya se tiene las clases, luego se necesita definir los atributos y las relaciones entre ellas. Para ello se define las relaciones binarias (Propiedades), la clase origen (Dominio), la clase destino (rango).

Tabla 7. Lista de propiedades de las clases

Clase Origen	Propiedad	Clase Destino	Relación Inversa
UTPL	tieneMod	Modalidad	perteneceU
SistemasInformaticosYComp utacion	tieneMalla	MallaCurricular	
Titulación	tieneCE	CompetenciasEspecificas	
Modalidad	tieneA	Areas	
CompetenciasEspecificas	poseeCC	CompetenciasComponente Academico	
PlanDeEstudios	peTieneMat	Materias	
PlanDeEstudios	poseeCE	Materias	
CompetenciasComponenteA cademico	pertenecen	Materias	
DepartamentoAreaTecnica	otorgaTi	Titulacion	
CompetenciasComponenteA cademico	iContieneH	NivelHabilidadComplejidad	
Tecnica	formadaDep	DepartamentoAreaTecnica	
MallaCurricular	conformadaM ate	Materias	
CompetenciasEspecificas	ceContieneH	NivelHabilidadComplejidad	
CompetenciasEspecificas	evaluadaPorC C	CompetenciasComponente Academico	

Fuente: Elaboración propia.

También pueden ser representadas por relaciones binarias, las cuales nos ayudan a tener una idea más clara de la relación que existe entre las clases.

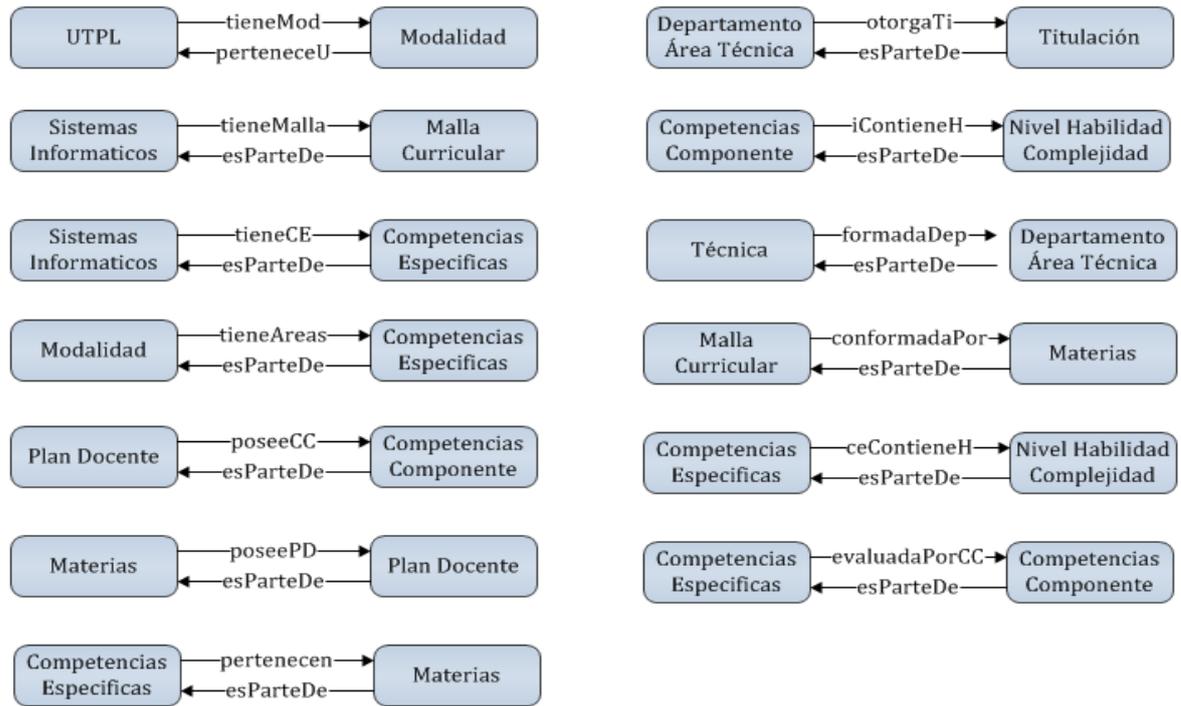


Figura 9. Diagrama de relaciones binarias.
Fuente: Elaboración propia.

Mapa Conceptual del modelado de verificación de competencias académicas.

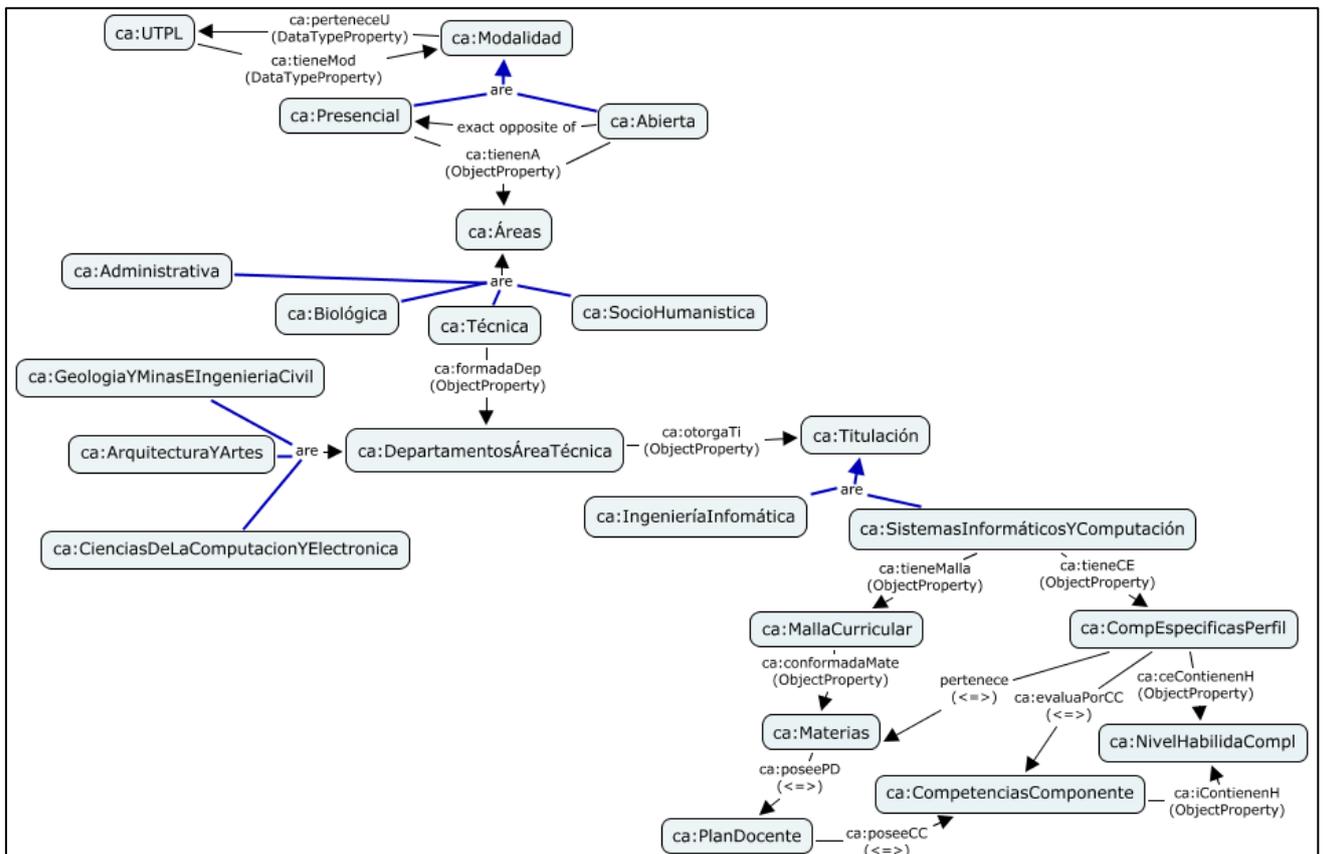


Figura 10. Modelo Conceptual de la ontología.
Fuente: Elaboración propia.

3.3.4 Definición de las restricciones de las propiedades.

Se detallará cada una de las restricciones de las propiedades de cada clase como tipo de dato requerido y la cardinalidad en base a los valores simple y múltiple.

Tabla 8. Lista de las restricciones de las propiedades.

Propiedad	Clase	Tipo de Dato	Cardinalidad
codMod	Modalidad	String	Simple
nombreMod	Modalidad	String	Simple
codArea	Areas	String	Simple
nombreArea	Areas	String	Simple
descripcionArea	Areas	String	Simple
codDep	DepartamentoAreaTecnica	String	Simple
nombreDep	DepartamentoAreaTecnica	String	Simple
numDep	DepartamentoAreaTecnica	Integer	Simple
codTitulacion	Titulacion	String	Simple
nombreTitulacion	Titulacion	String	Simple
codMateria	Materias	String	Simple
nomMateria	Materias	String	Simple
Créditos	Materias	Integer	Simple
nombMalla	MallaCurricular	String	Simple
numMaterias	MallaCurricular	Integer	Simple
descripcionMalla	MallaCurricular	String	Simple
codPlanEs	PlanDocente	String	Simple
nombPlanEs	PlanDocente	String	Simple
codComp	CompetenciasEspecificas	String	Simple
nombreComp	CompetenciasEspecificas	String	Simple
codCC	CompetenciasComponenteAcademico	String	Simple
nombreCC	CompetenciasComponenteAcademico	String	Simple

Sinónimos	Crear Evaluar Analizar Aplicar Comprender Recordar	String	Múltiple
-----------	---	--------	----------

Fuente: Elaboración propia.

3.3.5 Definición de axiomas formales.

En la siguiente **tabla 9**, se presentan los axiomas lógicos en las cuales se definen expresiones lógicas que siempre son verdades. La tabla de cada axioma está conformada por nombre, descripción, concepto, expresión matemática y las relaciones.

Tabla 9. Axiomas de la Ontología de verificación de competencias académicas.

Axioma 1	
Nombre	Competencias del componente en un plan docente
Descripción	Todo plan docente tiene competencias del componente
Concepto	Plan docente, competencias componente.
Expresión Matemática	$\forall (x,y)(\text{Plan docente}(x) \wedge (\text{competencias componente}(y) \rightarrow \text{Tiene}(x,y)))$
Relaciones	poseeCC

Axioma 2	
Nombre	Competencias específicas en una titulación
Descripción	Toda titulación tiene competencias específicas.
Concepto	Titulación, competencias específicas.
Expresión Matemática	$\forall (x,y)(\text{titulación}(x) \wedge (\text{competencias específicas}(y) \rightarrow \text{Tiene}(x,y)))$
Relaciones	tieneCE

Axioma 3	
Nombre	Competencias del componente académico en una competencias específicas
Descripción	Toda competencia específica tiene competencias del componente
Concepto	Competencias específicas, competencia del componente
Expresión Matemática	$\forall (x,y)(\text{competencias específicas}(x) \wedge (\text{competencias componente}(y) \rightarrow \text{Tiene}(x,y)))$
Relaciones	evaluadaPorCC

Fuente: Elaboración propia.

3.3.6 Creación de Instancias.

Se crean cada una de las instancias de cada clase. En la tabla 10 se muestran algunas instancias de la ontología de verificación de competencias académicas.

Tabla 10. Creación de instancias.

Clase	Instancias
TroncalesTitulacion	Arquitectura de Aplicaciones Arquitectura de Computadoras Arquitectura y Computación Paralela Arquitectura y Seguridad de Redes Auditoría Informática Base de Datos Avanzada Electrónica Digital Estructura de Datos y Algoritmos Fundamentos de Base de Datos Fundamentos de Ingeniería de Software
CompetenciasEspecificas	Administrar centros de comunicación y datos (servidores y aplicaciones). Analizar, diseñar e instalar redes de telecomunicación. Analizar Problemas de Programación y Plantear Soluciones mediante Métodos Computacionales. Asimilar experiencias, conocimientos y herramientas para planificar o revisar sus estrategias de investigación en el área de ciencias de la computación.
CompetenciasComponenteAcademico	Comprender los conceptos que intervienen en el proceso del desarrollo de software. Conocer principios marcos de trabajo teorías y mejoras prácticas de la arquitectura empresarial. Evaluar y mejorar la accesibilidad, usabilidad y seguridad de sistemas y aplicaciones y servicios informáticos. Identificar el ámbito de la arquitectura empresarial.
Analizar	Averiguar comparar considerar descomponer detallar distinguir estudiar
Aplicar	Acercar acomodar

	adaptar adherir arrimar clavar
Comprender	Abarcar adivinar captar compendiar concebir conocer
Crear	Concebir designar elegir engendrar establecer fundar
Evaluar	Calcular determinar estimar justipreciar tasar valorar
Recordar	Acordar aludir evocar invocar mencionar

Fuente: Elaboración propia.

3.4 Implementación de la Ontología verificación de competencias académicas.

Existen un sinnúmero de editores que apoyan a la construcción de ontologías. Con las cuales se puede crear desde los conceptos, propiedades, relaciones, instancias, axiomas y restricciones. Una de las herramientas más completas usada para la construcción de ontologías es protégé; ya que se puede usar en diferentes formatos como: XML, OWL, RDF, RDFS; además está basado en java lo cual ayuda al desarrollo de aplicaciones y también es de licencia libre.

La versión de Protégé que se usará para la construcción de la ontología de verificación de competencias es **Protégé Versión 4.3 (build 304)**; aunque se usó además la versión 3.4.4, ya que en esta versión es más sencillo modificar nombres de clases, subclases, etc y

además se puede visualizar todos los atributos que forman parte de un individuo específicamente.

En la siguiente figura se puede ver una captura de la interfaz de protégé 4.3.

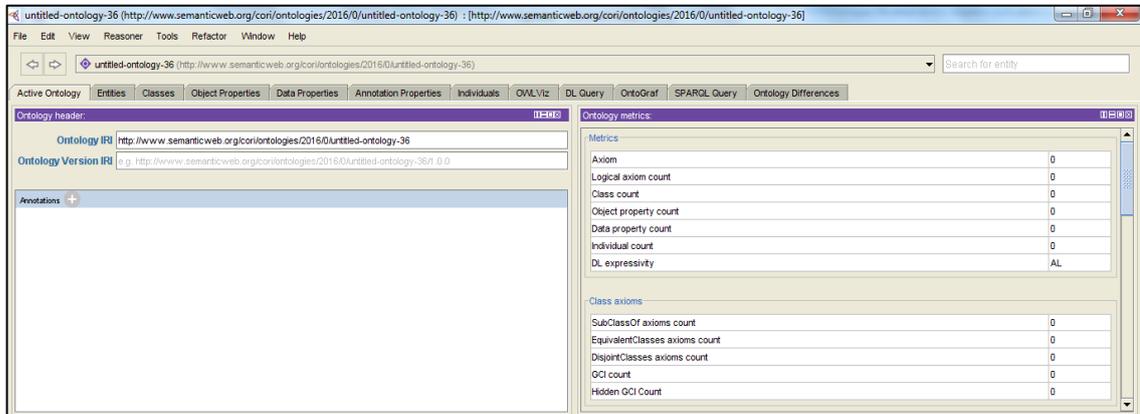


Figura 11. Interfaz Protégé.
Fuente: Elaboración propia.

3.4.1 Definición de los conceptos.

Para la definición de conceptos de la ontología se utiliza la pestaña *Classes* de Protégé. Aquí se da el nombre a cada una de las clases, se crean las superclases (clases padres), y las subclases (clases hijas).

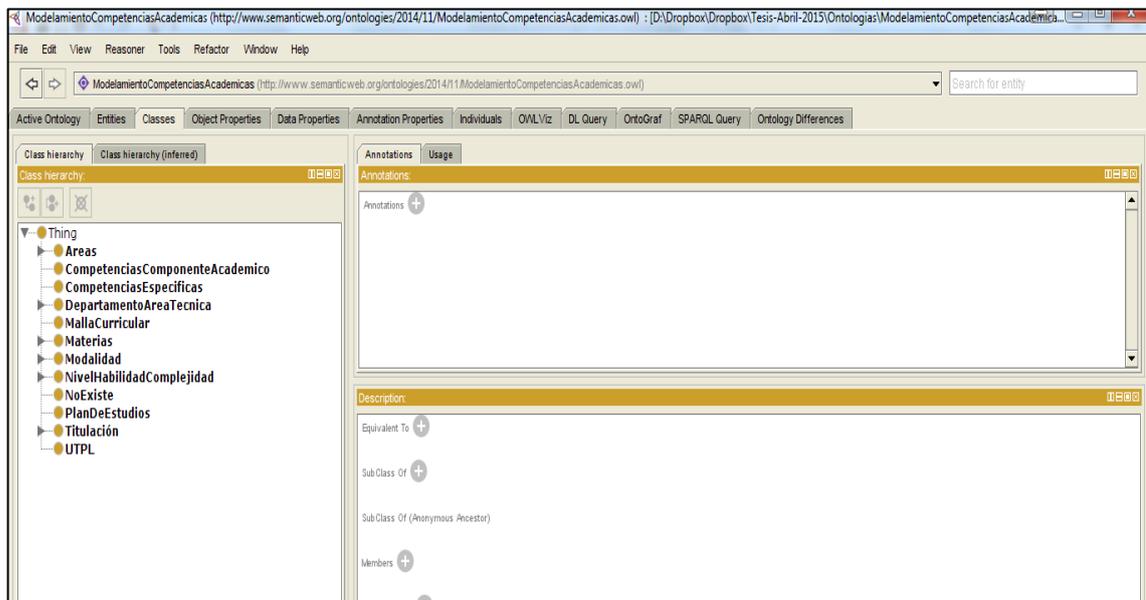


Figura 12. Definición de clases en Protégé.
Fuente: Elaboración Propia.

3.4.2 Definición de las relaciones entre clases.

Se definen las relaciones de las clases usando la pestaña Object Properties de Protégé, se le asigna el nombre a la relación y se debe asignar la clase dominio y la clase rango.

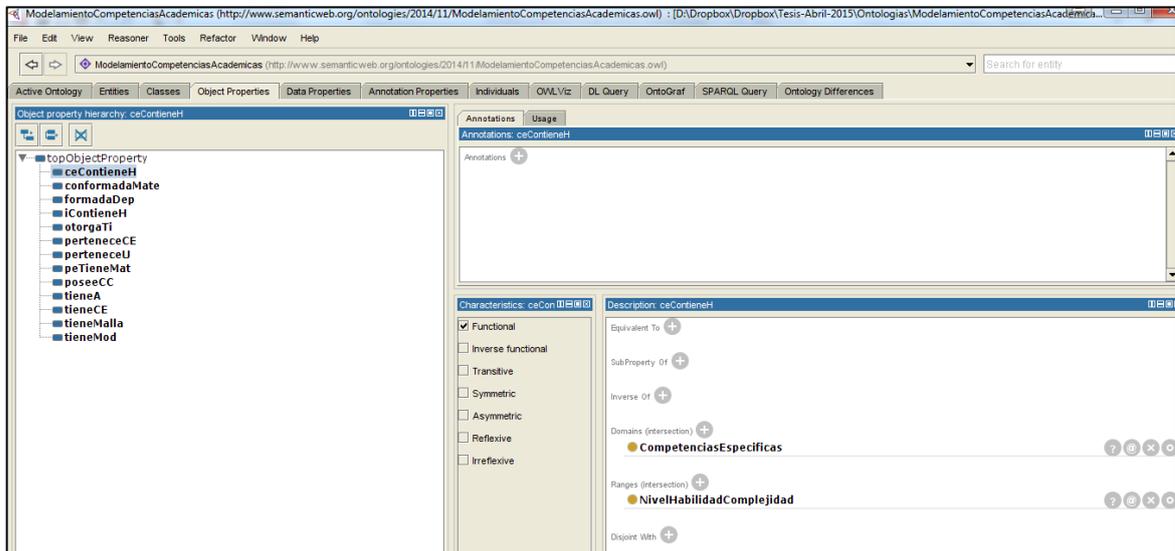


Figura 13. Definición de las relaciones entre clases.
Fuente: Elaboración propia.

3.4.3 Definición de atributos.

Para definir los atributos se usa la pestaña Data Properties, se debe asignar un nombre y el dominio que es a la clase que pertenece y el rango que detalla el tipo de propiedad.

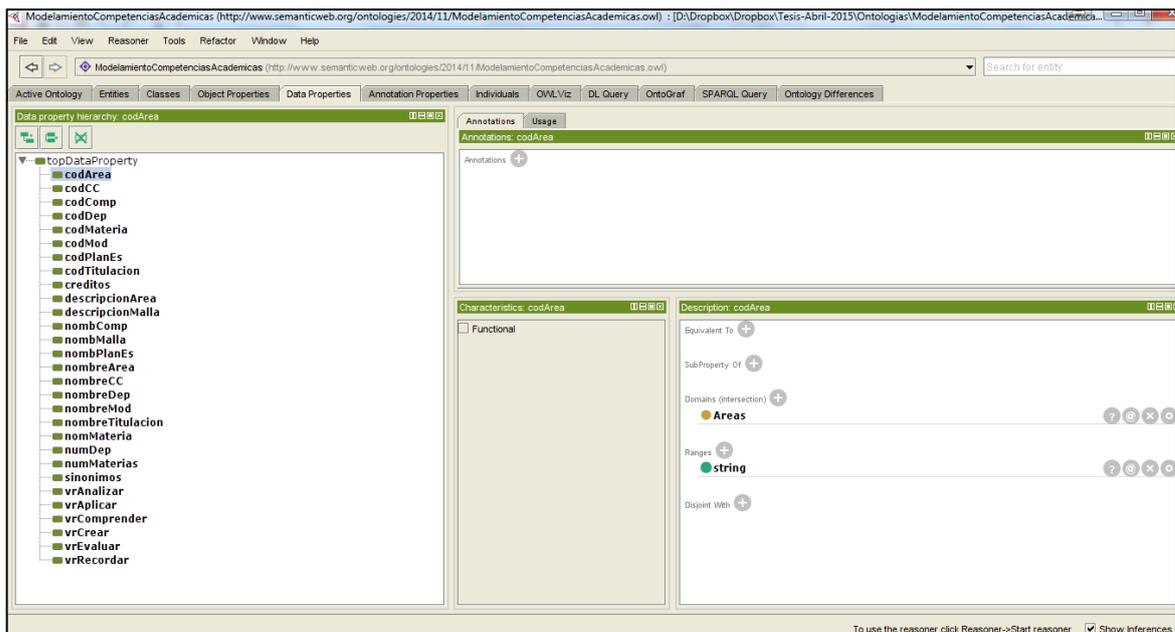


Figura 14. Definición de Atributos.
Fuente: Elaboración propia.

3.4.4 Definición de axiomas.

En la definición de axiomas se utilizan las relaciones y las clases existentes en la ontología para asignarles cardinalidad o cuantificadores ya sea universal o existencial.

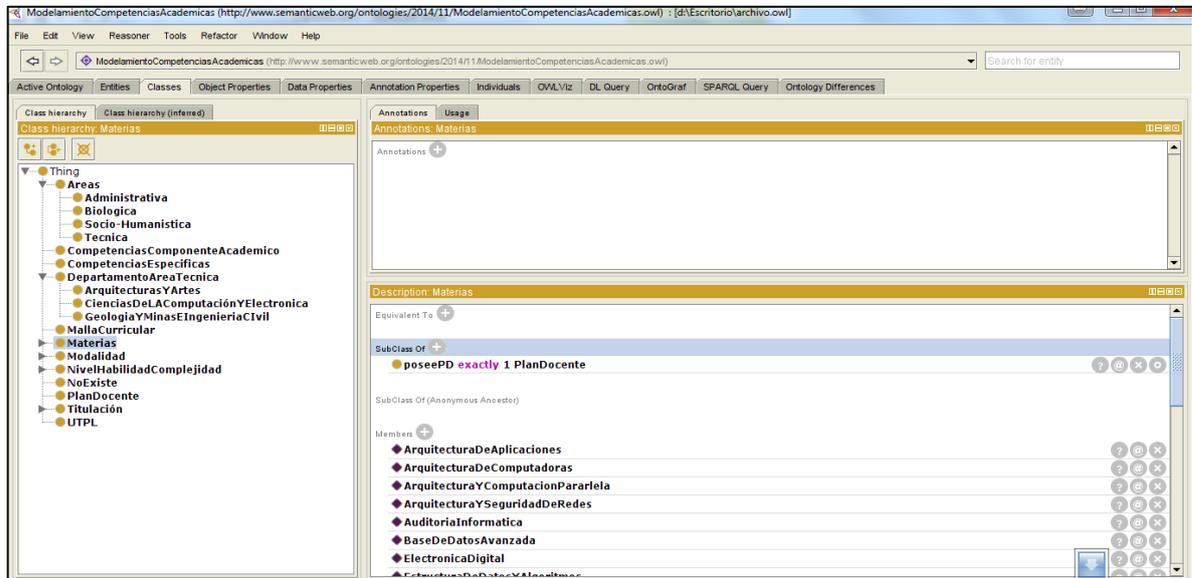


Figura 15. Definición de axiomas.

Fuente: Elaboración propia.

3.4.5 Declaración de Instancias.

Se definen instancias a cada clase dentro de la pestaña Individuals, se crea la instancia y se le asigna a la clase que pertenece. Aquí se asignan las Materias, competencias específicas, planes de componentes, niveles de la taxonomía de Bloom y el corpus de sinónimos.

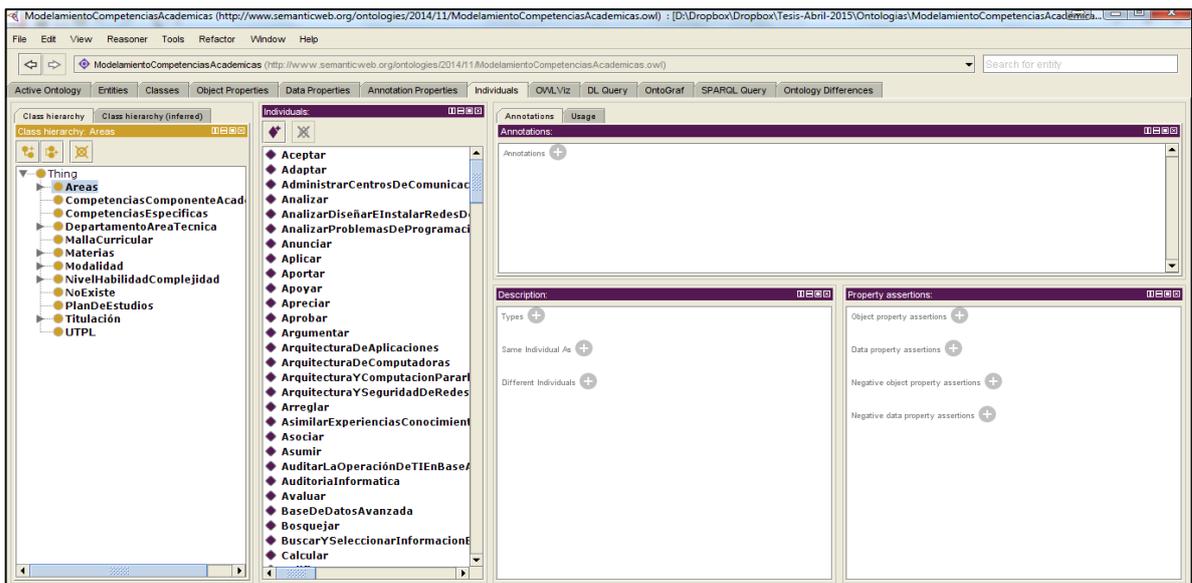


Figura 16. Definición de instancias de clases.

Fuente: Elaboración propia

Se crean además las instancias de los atributos creados como: nombre de competencias, nombres de materias, créditos de materias, nombre de plan de componente o estudios, ect.

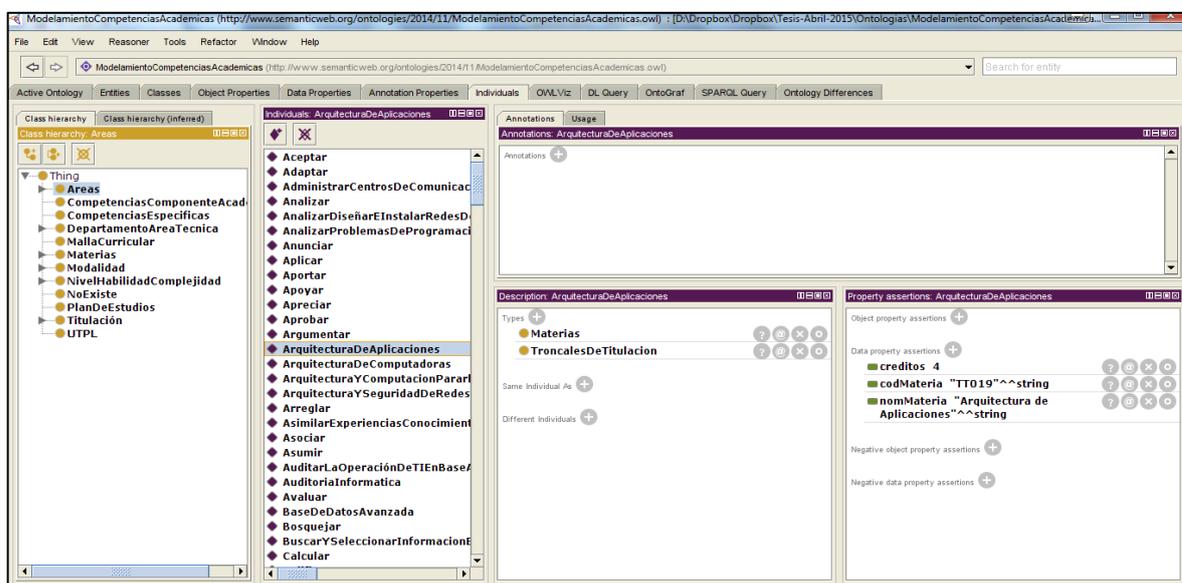


Figura 17. Definición de instancias de atributos.
Fuente: Elaboración propia.

3.4.6 Definición de consultas.

Se definirán algunas consultas realizadas sobre el dominio de verificación de competencias académicas, pero tres de ellas son las siguientes: La primera consulta a continuación mostrará las materias que se usaran para realizar la verificación de competencias académicas.

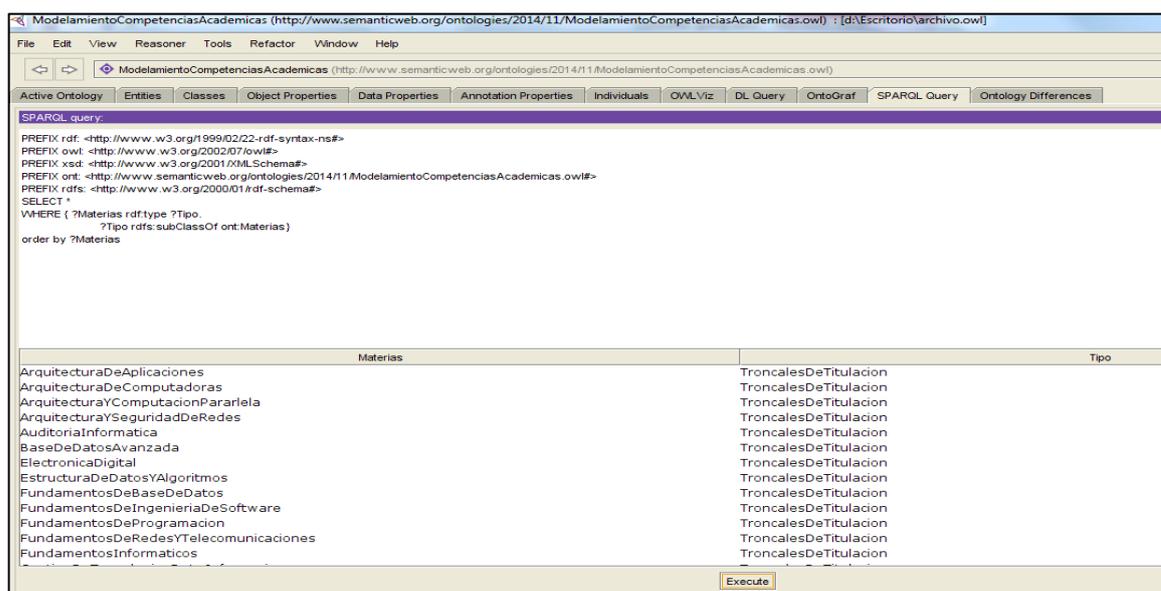


Figura 18. Consulta de materias troncales de titulación.
Fuente: Elaboración propia.

En la segunda consulta expondrá las competencias específicas de la titulación.

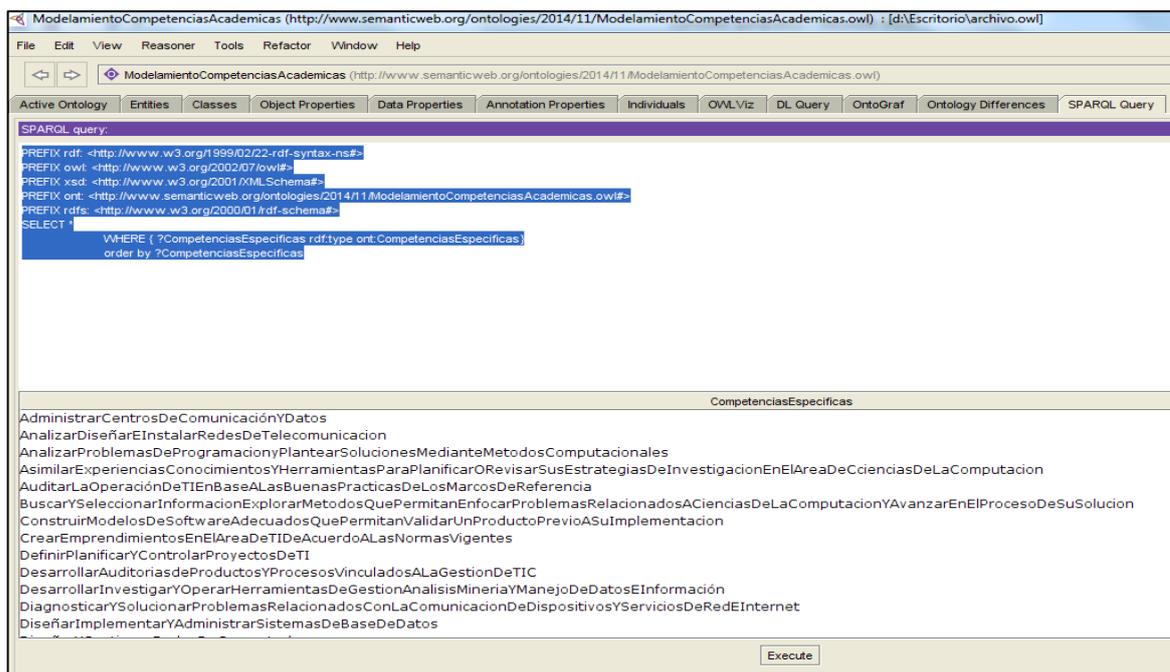


Figura 19. Consulta de competencias específicas.
Fuente: Elaboración propia.

La última consulta será de los verbos que pertenecen a la taxonomía de Bloom, con sus respectivos verbos relacionados y sinónimos.

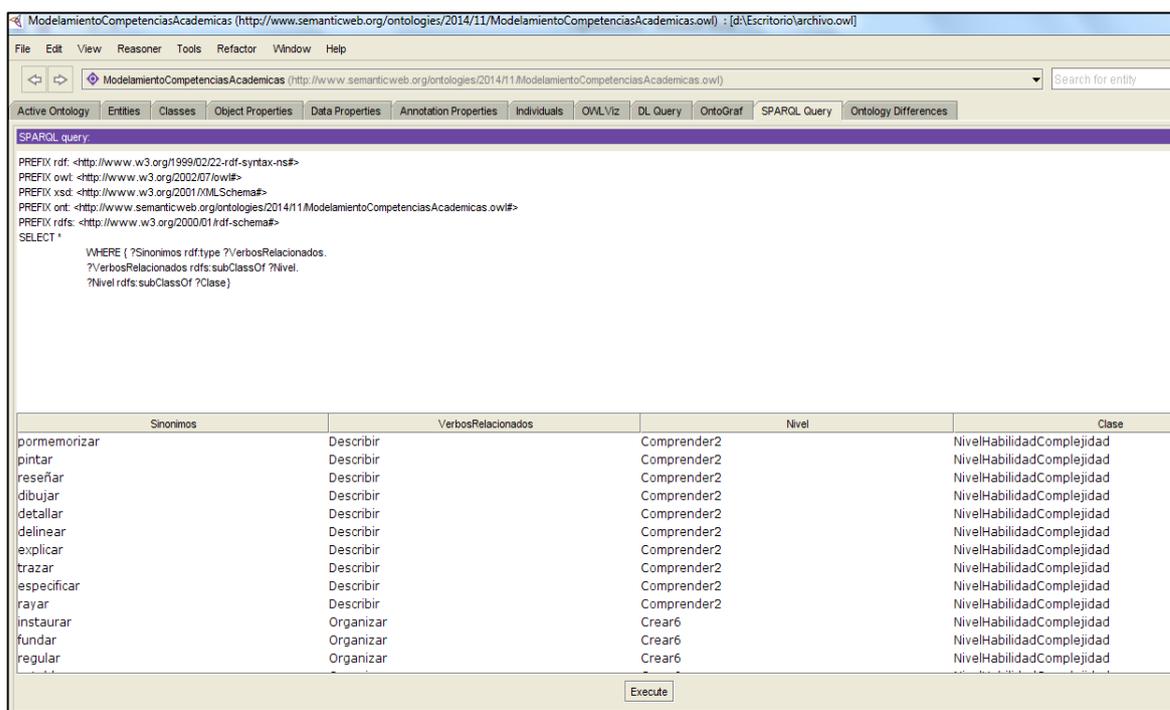


Figura 20. Consulta verbos que pertenecen a la taxonomía de Bloom.
Fuente: Elaboración propia.

3.4.7 Validación de la Ontología.

La validación de la ontología, se la realizo con el uso del razonador Pellet, tanto en Protégé 3.4.8; como en Protégé 4.3; con Protégé 3.4.8 para poder demostrar que tanto la consistencia, taxonomía e inferencia están correctamente, ya que esta versión lo muestra a través de mensajes de avisos y mucho más fácil identificar en qué fase está fallando la ontología, y en la versión 4.3 porque fue con la que se trabajó el modelado y nos muestras las inferencia que realiza el razonador.

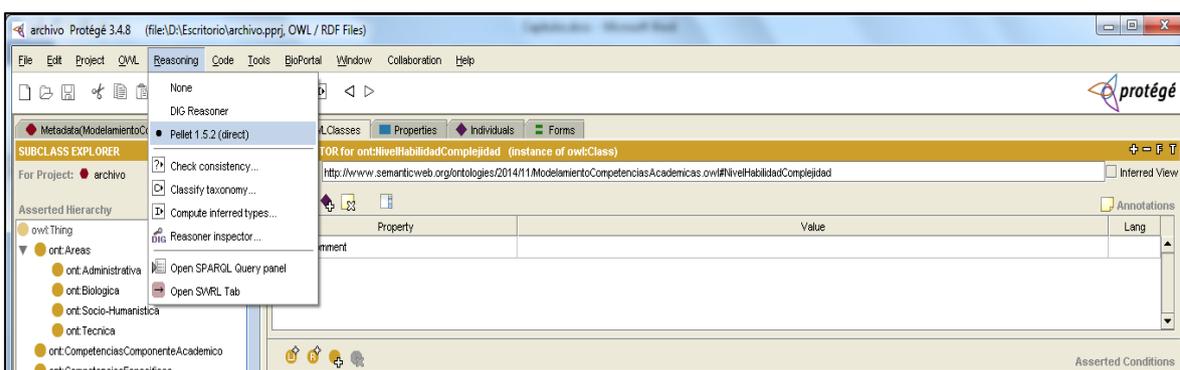


Figura 21. Escoger el razonador (Pellet) Protégé 3.4.8.
Fuente: Elaboración propia.

Verificación de inconsistencias permite constatar que no hay contradicción en la ontología construida.

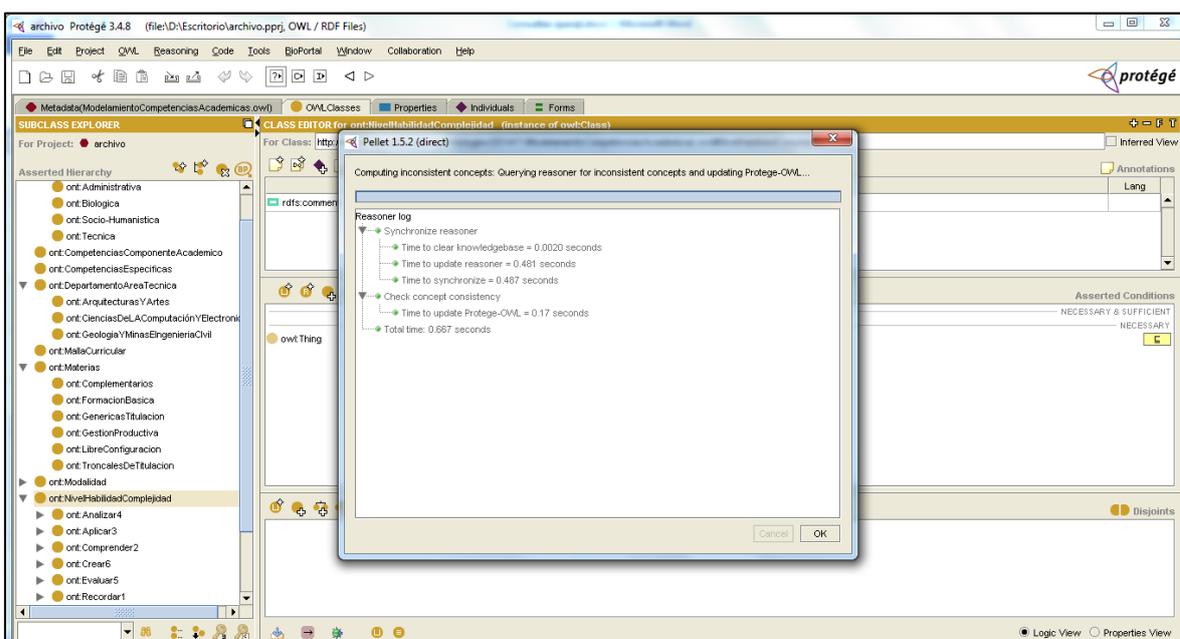


Figura 22. Verificación de inconsistencia de datos, Protégé 3.4.8.
Fuente: Elaboración propia.

Verificación de taxonomía de clases, prestar atención a la relación entre cada clases y verifica que la jerarquía de clases este completa.

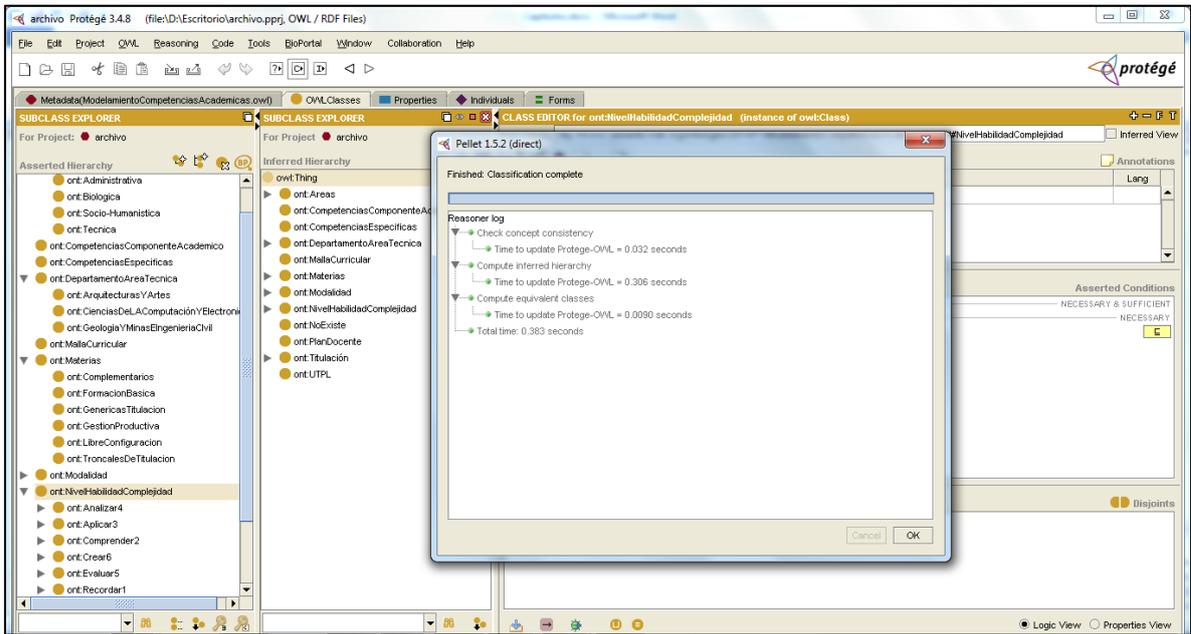


Figura 23. Verificación de Taxonomía de clases, Protégé 3.4.8.
Fuente: Elaboración propia.

Verificación de inferencia de clases, encuentra las clases a las que corresponde cada uno de los individuos

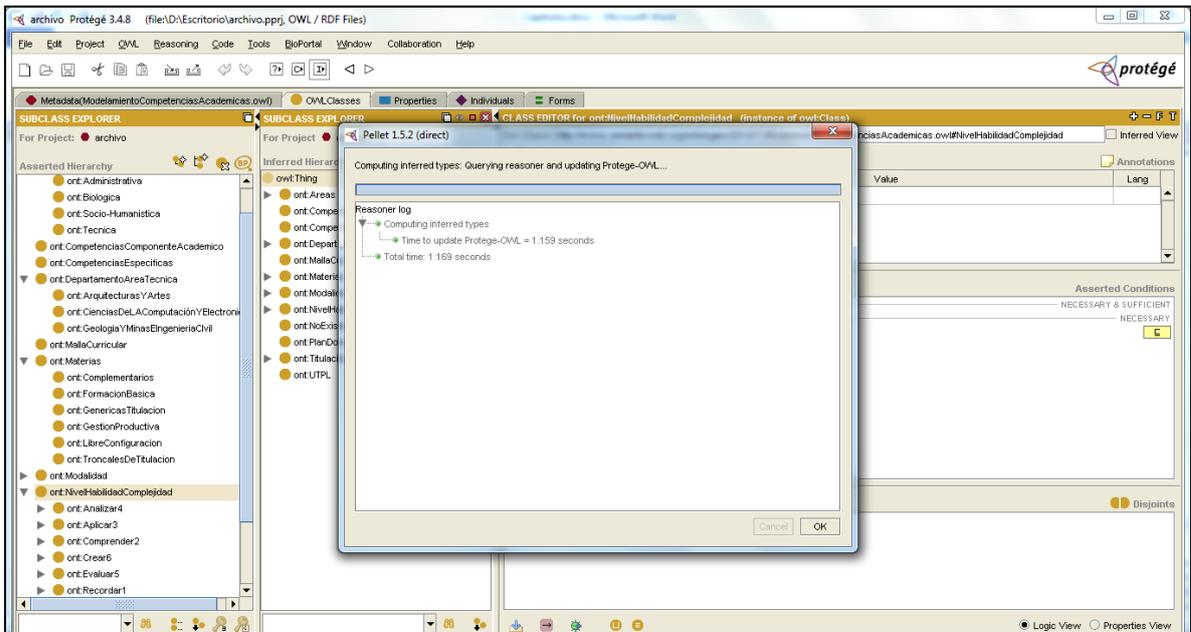


Figura 24. Verificación de inferencias de clases, Protégé 3.4.8.
Fuente: Elaboración propia.

Inferencia, el razonador muestra las clases o individuos que según las relaciones y propiedades son parte de forma predeterminada.

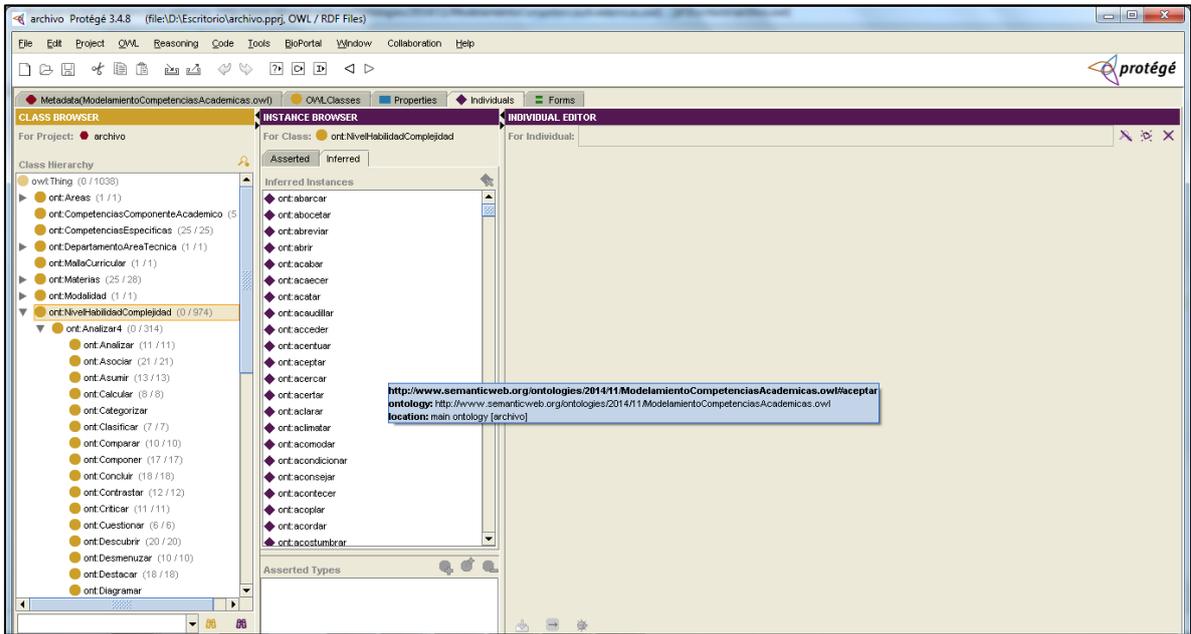


Figura 25. Inferencia realizada por el razonador Protégé 3.4.8.
Fuente: Elaboración propia.

En la versión 4.3 se escoge el razonador y luego se lo inicializa en *Start reasoner*.

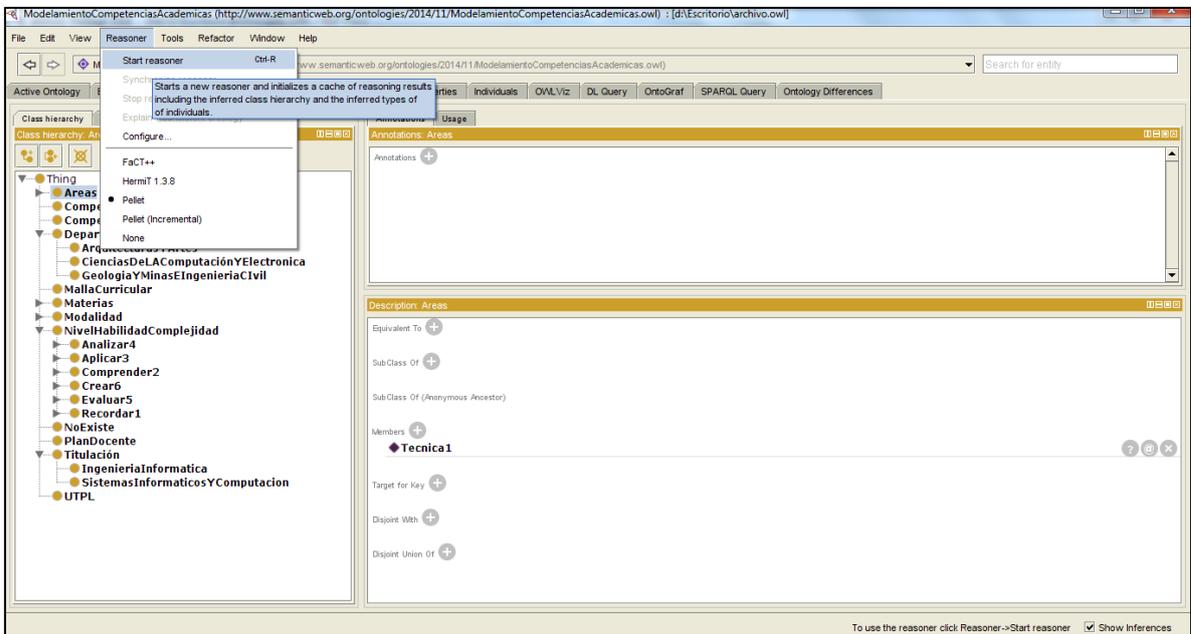


Figura 26. Escoger el razonador e iniciarlo. Protégé 4.3.
Fuente: Elaboración propia.

Al igual que la versión 3.4.8 se puede ver las inferencias realizadas por el razonador.

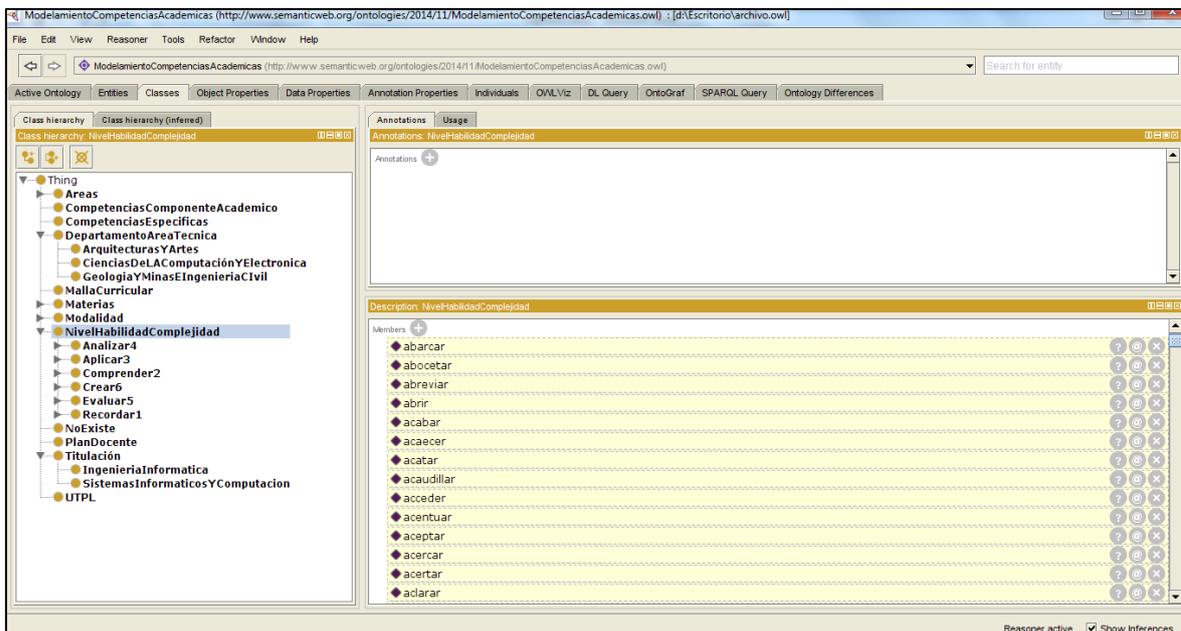


Figura 27. Instancias de nivel de habilidad inferidas por el razonador. Protégé 4.3.
Fuente: Elaboración propia.

3.5 Evaluación de versiones de la ontología.

En la construcción de una ontología se realiza un sinnúmero de pasos, los mismo que se van corrigiendo conforme se tiene una idea clara del modelado que se va a realizar, por eso puede darse el caso que existan algunas versiones de la ontología construida.

En el caso de la ontología de verificación de competencias se construyeron tres versiones. A continuación se muestran cada una de las versiones.

- En la primera versión, se crea la ontología en base al conocimiento inicial, es por eso que en la primera versión tiene más número de clases, atributos y relaciones.

Tabla 11. Primera versión de la ontología.

criterio	Valor
Número de Clases	223
Número de Atributos	35
Número de relaciones	17

Fuente: Elaboración propia.

Ontology metrics:	
Metrics	
Axiom	3966
Logical axiom count	2651
Class count	223
Object property count	17
Data property count	35
Individual count	1040
DL expressivity	ALCF(D)

Figura 28. Primera versión de la ontología Protégé 4.3.
Fuente: Elaboración propia

- En la segunda versión en cambio, se eliminó clases que no eran necesarias para modelar el dominio, se eliminaron también las respectivas relaciones y atributos. Además se hizo el cambio de una clase por individuo.

Tabla 12. Segunda versión de la ontología.

Criterio	Valor
Número de Clases	222
Número de Atributos	32
Número de relaciones	16

Fuente: Elaboración propia.

Ontology metrics:	
Metrics	
Axiom	3961
Logical axiom count	2651
Class count	222
Object property count	16
Data property count	32
Individual count	1040
DL expressivity	ALCF(D)

Figura 29. Segunda versión de la ontología Protégé 4.3.
Fuente: Elaboración propia.

- En la tercera y última versión se eliminó clases no instanciadas, los atributos y relaciones no usadas, Por último se obtuvo una ontología consistente dentro del dominio planteado.

Tabla 13. Segunda versión de la ontología.

Criterio	Valor
Número de Clases	214
Número de Atributos	26
Número de relaciones	14

Fuente: Elaboración propia.

Ontology metrics:	
Metrics	
Axiom	5512
Logical axiom count	4035
Class count	214
Object property count	14
Data property count	26
Individual count	1223
DL expressivity	ALCF(D)
Class axioms	
SubClassOf axioms count	260
EquivalentClasses axioms count	0
DisjointClasses axioms count	1
GCI count	0
Hidden GCI Count	0

Figura 30. Segunda versión de la ontología.
Fuente: Elaboración propia.

3.6 Métricas de Evaluación.

Dentro de las métricas utilizadas para la evaluación de la ontología fue la completión de clases y la completión de relaciones.

3.6.1 Completión de clases.

La completión de clases comprueba que la ontología tenga los conceptos necesarios para esta. La evaluación se la realiza mediante la riqueza de clases y riqueza de propiedades.

3.6.1.1 Riqueza de clases.

Si **RC** es muy alto (cercano a 1) se puede afirmar que la ontología representa, en gran parte, el conocimiento del dominio.

$$RC = \frac{|C'|}{|C|}$$

$|C'|$ es el número de clases que han sido instanciadas y $|C|$ es el número total de clases.

Tabla 14. Evaluación de riqueza de clases.

Criterio	Valor
Número de clases Instanciadas	202
Número total de clases	214
Riqueza de clases	0,94

Fuente: Elaboración propia.

$$RC = \frac{202}{214} = |0,94$$

Como está cercano a 1 se puede decir que la ontología representa el dominio deseado.

3.6.1.2 Riqueza de propiedades.

RP representa el promedio de propiedades por clase. Si **RP** es muy alto indica que cada clase tiene, en promedio, un número alto de atributos y por lo tanto contiene más conocimiento. Si **RP** es muy bajo podría indicar que se brinda menos información sobre cada clase.

$$RP = \frac{|A|}{|C|}$$

|A| es el número de atributos/ propiedades para todas las clases y |C| es el número de clases

Tabla 15. Evaluación de riqueza de propiedades.

Criterio	Valor
Número de propiedades	26
Número de clases	214
Riqueza de clases	0,12

Fuente: Elaboración propia.

$$RP = \frac{26}{214} = 0,12$$

El promedio de propiedades por clase se encuentra muy bajo, pero es debido a que existen un gran número de subclases, las cuales heredan las propiedades de la superclase.

3.6.2 Compleción de relaciones

La completión de relaciones, determina si el número total de relaciones permiten inferir el conocimiento que esta explícito en la ontología.

3.6.2.1 Riqueza de relaciones.

Si **RR** está cerca de cero indica que la mayoría de las relaciones es de tipo clase-subclase (es decir, relación *is-a*). Si **RR** está cerca de uno indica que la mayoría de las relaciones son de otro tipo que las relaciones de subclases.

$$RC = \frac{|P|}{|SC| + |P|}$$

Donde $|P|$ es el número de relaciones definidas en la ontología y $|SC|$ es el número de subclases.

Tabla 16. Evaluación de riqueza de relaciones.

Criterio	Valor
<i>Número de relaciones</i>	14
<i>Número clases</i>	203
<i>Riqueza de clases</i>	0,06

Fuente: Elaboración propia.

$$RC = \frac{|14|}{|203| + |14|} = 0.06$$

RR está más cerca de cero, por lo que indica que la mayoría de clases de tipo is-a ósea subclases.

CAPITULO IV: APLICACIÓN DE VERIFICACIÓN DE COMPETENCIAS ACADÉMICAS.

4.1 Metodología de la Aplicación

Para la construcción del DEMO de verificación de competencias académicas, se utilizó la metodología de desarrollo XP, la cual se enfoca principalmente en el usuario y en realizar cambios durante la marcha. La programación extrema es una metodología ligera que se fundamenta en la simplicidad, comunicación, retroalimentación, coraje y respeto. El ciclo de vida de la metodología XP cumple con los siguientes pasos:

- Planeación: Se toman las historias de usuarios, a partir de estas se definen los requerimientos posteriormente se crea un plan de entregas especificando las fechas y los artefactos a entregarse en cada etapa.
- Diseño: el cual tiene que ser un diseño muy sencillo y debe implementarse en base a la historia ni más ni menos, se permite el rediseño que controlar dichas modificaciones, sugiriendo pequeños cambios en el diseño que tienen la capacidad de mejorarlo en forma radical.
- Codificación: se realiza pruebas unitarias en base a las historias, interacción continua con el usuario.
- Pruebas: conforme se desarrolla, se deben implementar pruebas unitarias si es posible a diario para que se corrijan los errores a tiempo, también se realizan pruebas de aceptación estas se nacen a partir de las historias de los usuarios que se han implementado como parte de la liberación del software.

A continuación en la Figura 31, se muestra el ciclo de vida de la metodología XP.



Figura 31. Ciclo de Vida de la metodología XP
Fuente: (Gonzalez, 2012)

4.1.1 Arquitectura de la Aplicación.

Para el manejo de la ontología se hará el uso del API Jena en versión jena-2.10.1; que está basado en java y sirve para el desarrollo de aplicaciones web, para la implementación se usará el editor de desarrollo NetBeans IDE 8.0.2.

En la siguiente figura se muestra la arquitectura de la aplicación de verificación de competencias.

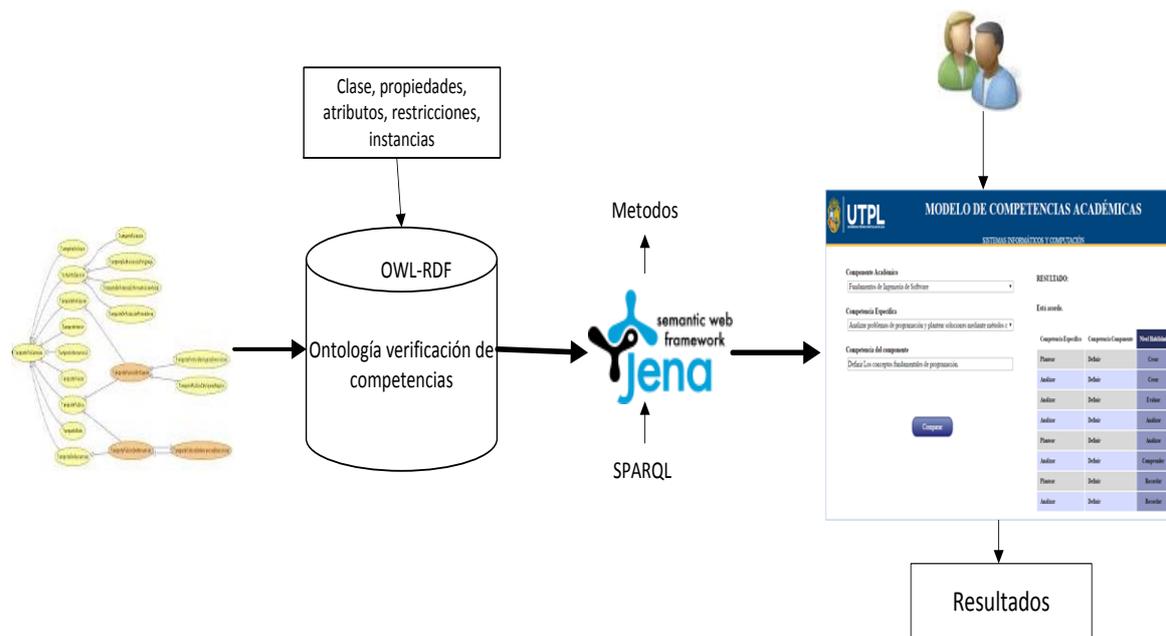


Figura 32. Arquitectura de la aplicación.

Fuente: Elaboración propia.

Para el desarrollo del aplicativo primeramente se tuvo que agregar la librería de Jena, luego se copió la ontología en el proyecto directamente, para no tener problemas de ubicación con el archivo.

Se implantan los métodos, y las consultas SPARQL pertinentes que me permitan devolver la información requerida y de esta forma presentar los resultados de la comparación de niveles de habilidad.

4.1.2 Caso de uso

A continuación en la figura 33 se muestra el caso de uso de la aplicación de verificación de competencias académicas.

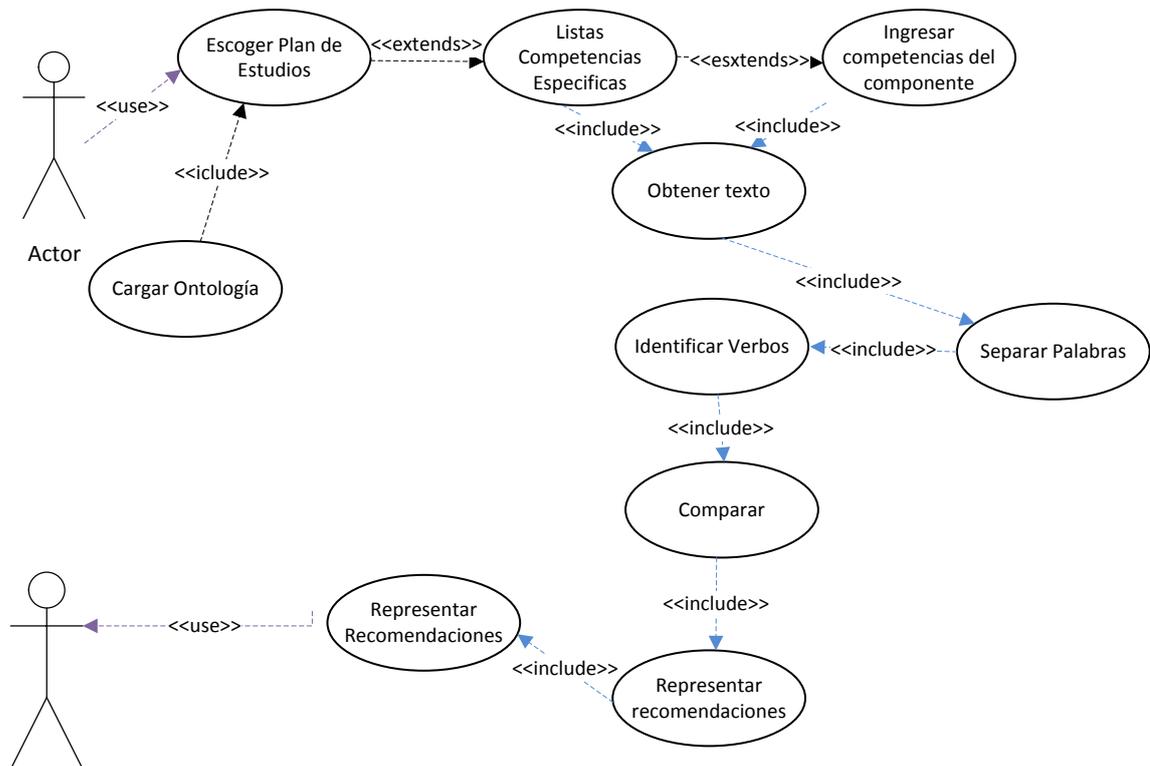


Figura 33. Casos de uso de verificación de competencias académicas.
Fuente: Elaboración propia.

En la aplicación se definirán las reglas y condiciones que debe cumplir la aplicación, entre ellas tenemos:

- **Cargar la ontología:** a través de Jena, se carga la ontología a un servicio web.
- **Seleccionar Planes de componente:** se realiza una consulta a la ontología de donde se obtendrán todas los planes de la titulación se sistemas informáticos y computación.
- **Seleccionar competencias específicas:** se realiza una consulta a la ontología la cual ya tiene asignadas las competencias específicas para cada materia, por lo cual se mostraran solo las competencias de la materia que se escoja.
- **Asignar competencias del componente:** se cree conveniente que se pueden ingresar las competencias del componente, ya que estas son ingresadas por el docente conforme a su experiencia y por lo tanto no necesitan estar cargadas.

- **Separar Palabras:** Separar cada una de las palabras de la oración, para poder extraer el verbo.
- **Identificar verbos:** se crea la condición de que obtenga todas las palabras de la frase que terminen en ar, er, ir; ya que la acción está representada por un verbo activo y estos verbos en infinitivo terminan en ar, er, ir.
- **Convertir todas las palabras en mayúsculas:** se hizo esta conversión porque si la palabra se la escribía con la primera letra mayúscula y en la ontología estaba en minúscula no la reconocía como la misma palabra; para esto se tuvo que agregar el prefijo `fn:concat`.
- **Comparar los verbos obtenidos:** se verifica que los verbos se encuentren en los niveles implementados en la ontología.
- **Presentar resultados:** Se menciona si la habilidad de la competencia específica está acorde o no a la competencia del componente.
- **Mostrar Recomendaciones:** se construye las tablas con sus respectivos resultados, ya sea en el caso de que estén acorde o no los verbos del componente académico con los de la competencia específica.

4.1.3 Modelo visto desde la aplicación web

A continuación se especifica el comportamiento del modelo, en la página de inicio hay una breve descripción de lo que hace la aplicación.



Figura 34. Página de inicio.
Fuente: Elaboración propia.

Luego se hace clic en ENTRAR, y aparece la página principal en donde se realizará la verificación de competencias académicas. Aquí primero se selecciona la materia o componente académico.

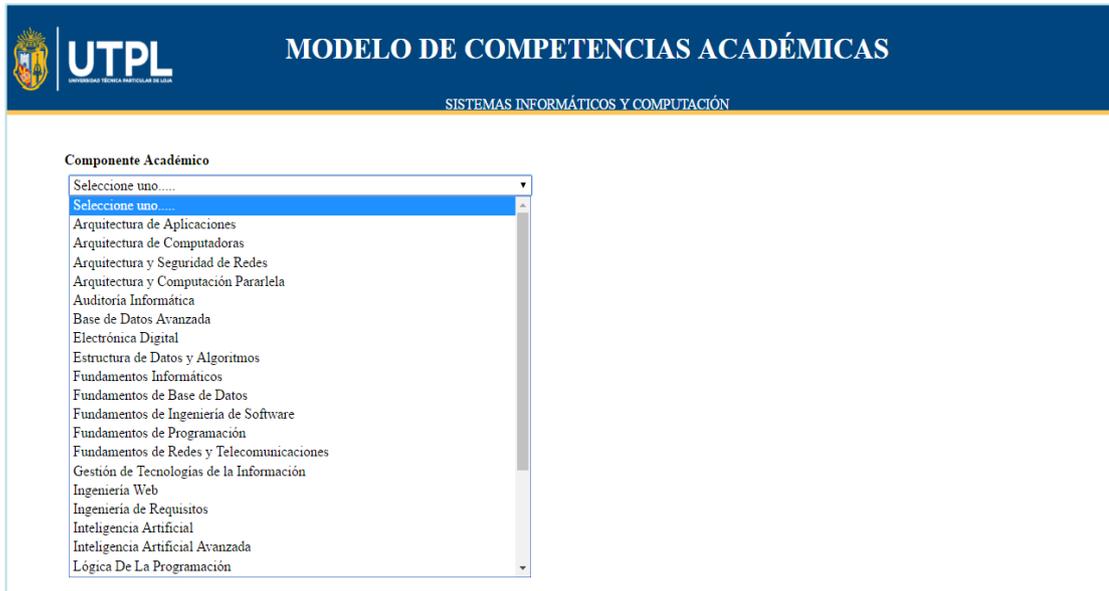


Figura 35. Seleccionar componente académico.
Fuente: Elaboración propia.

Luego se selecciona la competencia específica que desea verificar de la materia que se escogió como se muestra en la figura 36.



Figura 36. Seleccionar competencia específica.
Fuente: Elaboración propia.

Una vez seleccionada la competencia específica, se debe escribir la competencia del componente que se quiera asignar.

Figura 37. Ingresar la competencia del componente.
Fuente: Elaboración propia.

Finalmente damos clic en comparar, y nos mostrará los resultados obtenidos y las recomendaciones.

Competencia Especifica	Competencia Componente	Nivel Habilidad
Plantear	Definir	Crear
Analizar	Definir	Crear
Analizar	Definir	Evaluar
Analizar	Definir	Analizar
Plantear	Definir	Analizar
Analizar	Definir	Comprender
Plantear	Definir	Recordar
Analizar	Definir	Recordar

Figura 38. Presentación de resultados
Fuente: Elaboración propia.

4.2 Casos de pruebas Manuales y de la Aplicación

Para los casos de pruebas manuales y de la aplicación se tomó como referencia los planes de estudios, y este el componente académico o materia, las competencias específicas y las competencias del componente académico, luego se identificó los verbos de cada una de las competencias, para después verificar que los verbos de las competencias se encuentren en el corpus de sinónimos de la taxonomía de Bloom.

En el caso de que los verbos si estén dentro del corpus, se especifica a qué nivel de habilidad (*crear, evaluar, analizar, aplicar, comprender y recordar*) corresponden para de esta manera comparar si la competencia del componente se encuentra o no en el mismo nivel de habilidad de la competencia específica.

Los casos identificados fueron tres: caso exitoso, caso sobre dimensionado, caso sub dimensionado y caso de no concordancia.

4.2.1 Caso Exitoso.

Cuando los verbos de habilidad de las competencias específicas corresponde a los niveles de habilidad de las competencias del componente.

Tabla 17. Caso de prueba exitoso.

Componente Académico	Competencias Específicas	Competencias del Componente Académico	Prueba Manual	Verbos Acorde
Fundamentos de Programación	Analizar problemas de programación y plantear soluciones mediante métodos computacionales.	Definir Los conceptos fundamentales de programación.	CE-> Analizar, Evaluar, Crear, Recordar, Comprender CE-> Aplicar, Crear, Analizar Definir-> Analizar, Recordar, Evaluar, Crear, Comprender.	Crear Evaluar Analizar Comprender Recordar

Fuente: Elaboración propia.

En la **figura 39**. Se muestra el caso exitoso con la primera competencia del componente asignado en la materia de Fundamentos de la Programación.

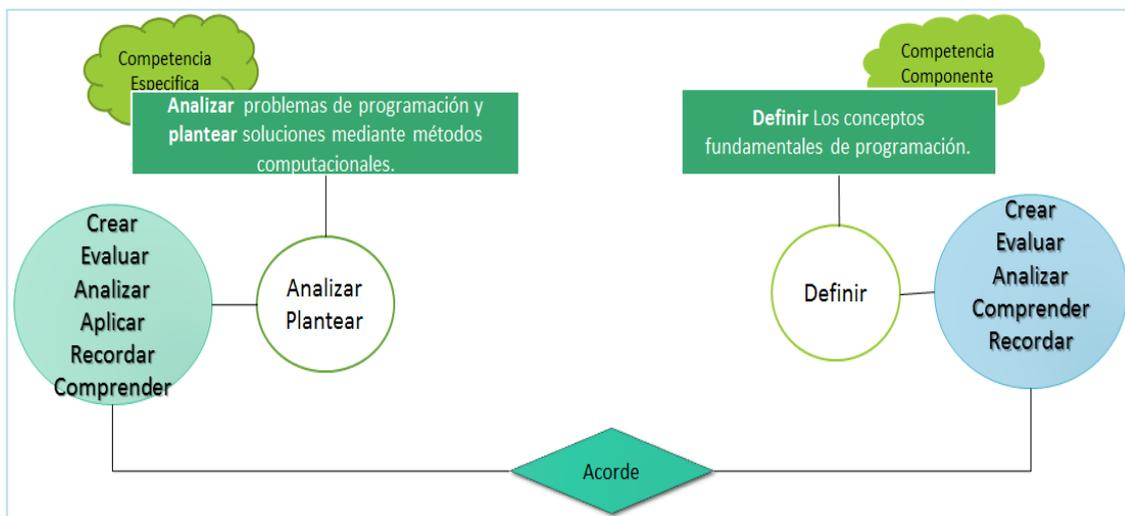


Figura 39. Caso exitoso
Fuente: Elaboración propia.

En el caso de prueba de la aplicación, nos dice que está acorde y además nos muestra una tabla especificando el nivel al que corresponde cada uno de los verbos, comprando respectivamente las competencias específicas con las competencias del componente.

UTPL
MODELO DE COMPETENCIAS ACADÉMICAS

SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Componente Académico
Fundamentos de Programación

Competencia Específica
Analizar problemas de programación y plantear soluciones mediante métodos c

Competencia del componente
Definir Los conceptos fundamentales de programación.

Comparar

RESULTADO:
Está acorde.

Competencia Específica	Competencia Componente	Nivel Habilidad
Plantear	Definir	Crear
Analizar	Definir	Crear
Analizar	Definir	Evaluar
Analizar	Definir	Analizar
Plantear	Definir	Analizar
Analizar	Definir	Comprender
Plantear	Definir	Recordar
Analizar	Definir	Recordar

Figura 40. Caso exitoso de la aplicación
Fuente: Elaboración propia.

4.2.2 Caso sobre dimensionado.

El caso sobre dimensionado es cuando la habilidad de la competencia del componente se encuentra en un nivel más alto que la habilidad de la competencia específica.

Tabla 18. Caso de prueba Sobre dimensionado.

Componente Académico	Competencias Específicas	Competencias del Componente Académico	Prueba Manual	Recomienda
<i>Redes y Sistemas Distribuidos</i>	Explorar y evaluar soluciones alternativas de tecnologías de la información y las comunicaciones, y emplearlas en la gestión y mejora de procesos empresariales.	Adecuar modelos de servicios de red.	CE-> Evaluar, Recordar, Comprender, Analizar.	No Acorde Evaluar.
			CE-> Evaluar, Aplicar, Analizar. Adecuar-> Crear	

Fuente: Elaboración propia.

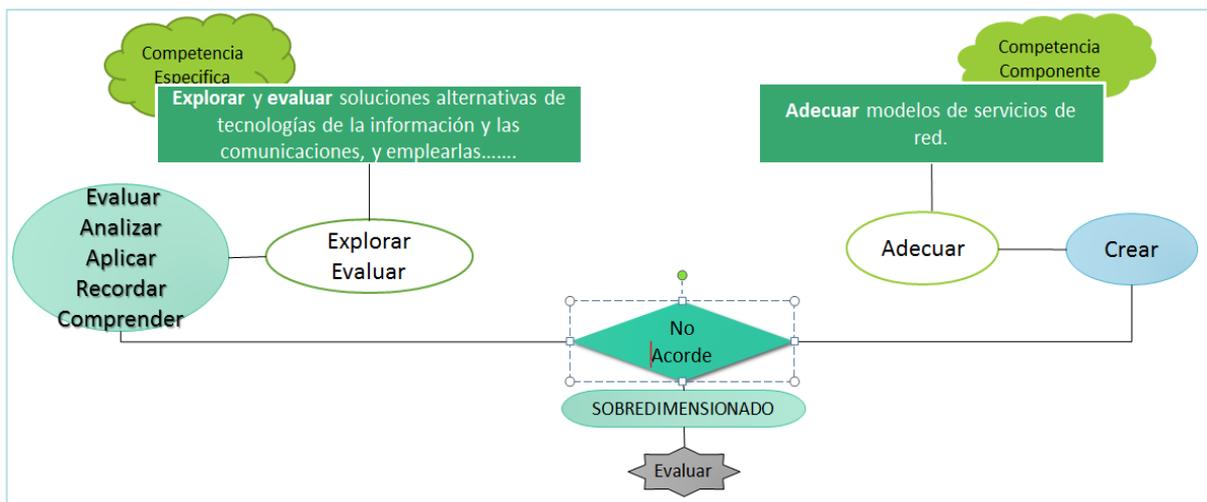


Figura 41. Caso Sobre dimensionado

Fuente: Elaboración propia.

En el caso sobre dimensionado en la aplicación menciona que no está acorde, recomienda el nivel más salto al que corresponde la competencia específica y recomienda una lista de verbos que pueden ser usados para dicho nivel de habilidad. Como se muestra a continuación en la **Figura 42**.

Figura 42. Caso sobre dimensionado de la aplicación
Fuente: Elaboración propia.

4.2.3 Caso sub dimensionado.

El caso sub dimensionado es cuando la habilidad de la competencia del componente se encuentra en un nivel más bajo que la habilidad de la competencia específica.

Tabla 19. Caso de prueba Sub dimensionado Manual.

Componente Académico	Competencias Específicas	Competencias del Componente Académico	Prueba Manual	Recomienda
Redes y Sistemas Distribuidos	Administrar centro de comunicación y datos.	Identificar los componentes de un modelo cliente-servidor de comunicaciones.	CE-> Crear Identificar-> Recordar, Comprender	No Acorde Crear.

Fuente: Elaboración propia.

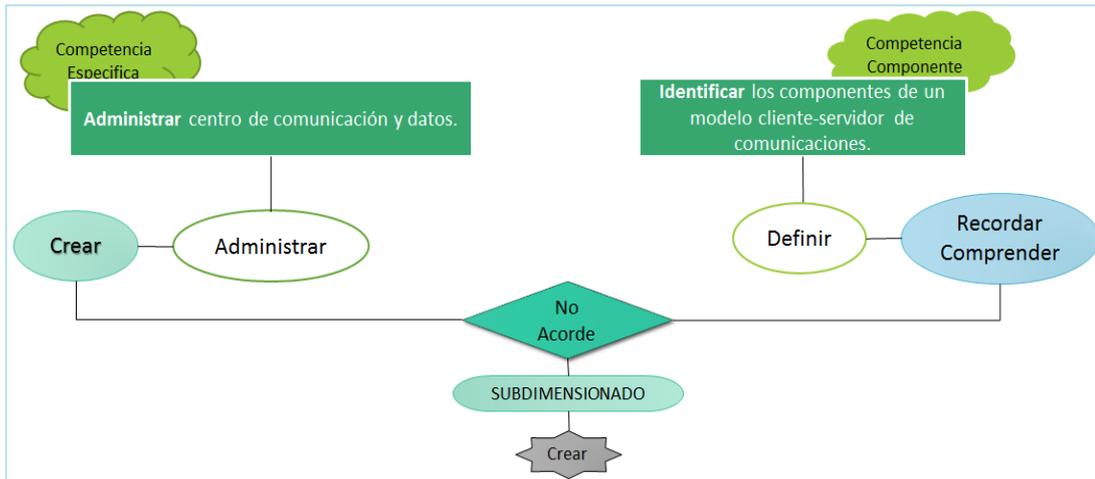


Figura 43. Caso Sub dimensionado Manual
Fuente: Elaboración propia.

Figura 44. Caso Sub dimensionado de la aplicación
Fuente: Elaboración propia.

4.2.4 Caso de no concordancia.

El último caso es cuando la habilidad de la competencia del componente no existe dentro de los niveles de habilidad aplicados.

Tabla 20. Caso de no Concordancia.

Componente Académico	Competencias Especificas	Competencias del Componente Académico	Prueba Manual	Recomienda
Redes y Sistemas Distribuidos	Explorar y evaluar soluciones alternativas de tecnologías de la información y las comunicaciones, y	Simular protocolos de capa de aplicación en varios escenarios de red.	CE-> Evaluar, Recordar, Comprender, Analizar. CE-> Evaluar, Aplicar, Analizar.	No Acorde Evaluar

	emplearlas en la gestión y mejora de procesos empresariales.		Simular -> No se encuentra en el corpus.	
--	--	--	---	--

Fuente: Elaboración propia.

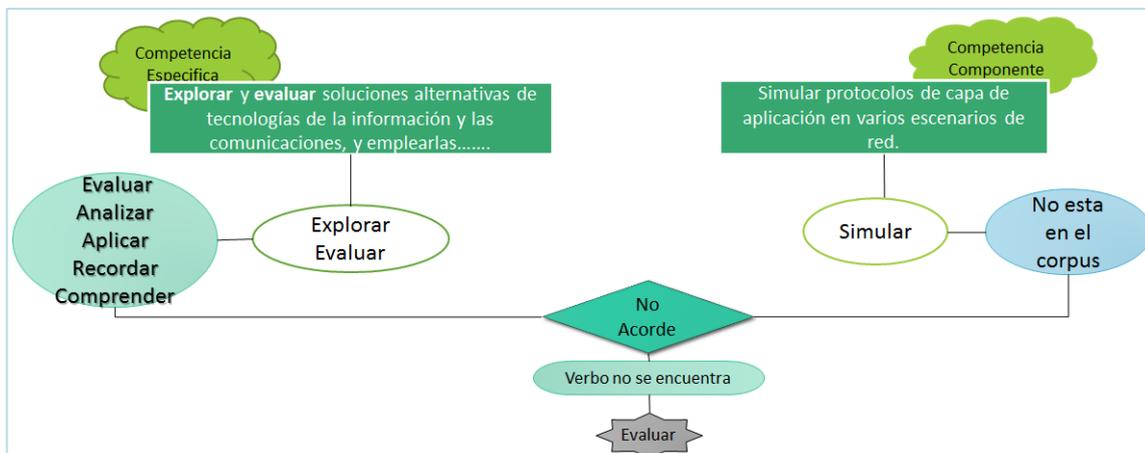


Figura 45. Caso de no concordancia manual.

Fuente: Elaboración propia.

MODELO DE COMPETENCIAS ACADÉMICAS

SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Componente Académico

Competencia Específica

Competencia del componente

RESULTADO:

No está acorde. Recomendamos usar:

Recomendados: (Evaluar)

-
-
-
-
-
-
-
-

Figura 46. Caso de no concordancia de la aplicación

Fuente: Elaboración propia.

ANÁLISIS DE RESULTADOS

- Caso de Éxito: la habilidad de la competencia específica y del componente se encuentran en el mismo nivel de habilidad, por lo tanto están acorde.
- Caso Sobre dimensionado: cuando la habilidad de la competencia del componente se encuentra en un nivel más alta que la habilidad de la competencia específica, por lo tanto no tiene relación con el nivel que está evaluando la competencia específica de la titulación, y se retroalimenta al docente recomendando el verbo del nivel más alto al que pertenece la habilidad de la competencia específica.
- Caso Sub dimensionado: cuando la habilidad de la competencia del componente se encuentra en un nivel más bajo que la habilidad de la competencia específica, en este caso se recomienda los verbos del nivel más alta al que pertenece la habilidad de la competencia específica.
- Caso no acorde: existen casos en los que la habilidad no pertenencia a ningún nivel. y si el caso que no se le asigne ninguna competencia del componente a la competencia específica el resultado será el mismo.
- Vale recalcar que actualmente existieron muy pocos de los casos de no concordancia.

CONCLUSIONES

En base al proyecto realizado se puede concluir lo siguiente que:

- El proceso de asignación de competencias de componente no es un tarea sencilla para el docente, la misma que debe a ser analizada meticulosamente al momento de definir la habilidad o el conocimiento, por lo que tener un modelo que sirva para la retroalimentación de habilidades, ayuda a evitar equivocaciones y agilizar el proceso.
- El uso de modelos ontológicos permiten realizar la verificación de competencias cuando se trata de establecer niveles de habilidad.
- El uso de metodologías y herramientas para la construcción de modelos ontológicos permiten que estos sean creados de forma satisfactoria, en base al modelo para resolver el problema citado.
- En base a los casos de prueba realizados, se demuestra que existen falencias al momento de definir el nivel de habilidad.
- Las verificaciones que se realizan en el modelo ontológico, demuestra que este fue implementado de forma correcta, ya que se logra la identificación de los niveles y la recomendación necesaria.

RECOMENDACIONES

Algunas consideraciones importantes que se resaltan en el desarrollo del presente trabajo de fin de titulación, las mismas que pueden servir como recomendaciones para futuros trabajos relacionados son:

- Implementar la verificación de competencias académica en todas las titulaciones de la UTPL.
- Usar más de una taxonomía de aprendizaje para la verificación de niveles de habilidad dentro de las competencias, y así incrementar el rango de las habilidades.
- Realizar la inclusión de la relación del dominio de conocimiento con la habilidad como parte del proceso de verificación de competencias académicas.
- Desarrollar un sistema automático que ayude a la vinculación directa entre las competencias y los indicadores de rendimiento.
- Implementar PLN en verificación de competencias, para que se definan correctamente cada una de las palabras usadas, ya sean habilidades o dominio de conocimiento.

BIBLIOGRAFÍA

- Alatrish, E. S. (2012). *Comparison of Ontology Editors*. *E-RAF Journal on Computing*, 4, 23–38.
- Alvarado, R. D. (2010). *Metodología para el desarrollo de ontologías*, 1–30.
- Alzina, R. B. (2003). *Educación Emocional Y Competencias*. *Revista de Investigación Educativa*, 21, 7–43. Retrieved from <http://revistas.um.es/rie/article/viewFile/109531/104121>
- Andujar, O. (2015). *Taxonomía de Bloom – Esquema para Redactar Competencias*. Retrieved November 10, 2015, from <http://www.orientacionandujar.es/2015/07/31/taxonomia-de-bloom-esquema-para-redactar-competencias/>
- Aquilino, J., Andrés, J. de, Nieto, C., Suárez, M., Ramón, J., Cernuda, A., ... Redondo, J. M. (n.d.). *Definición de competencias específicas y genéricas del Ingeniero en Informática*. Retrieved from <http://di002.edv.uniovi.es/~albizu/articulos/ice06b.pdf>
- Cabrera, N., & Mas, X. (n.d.). *El Plan Docente como metodología integrada para el diseño y la creación de asignaturas en la UOC*.
- Camacho, E. (2008). *VISUALIZACIÓN DE ONTOLOGÍAS Trabajo Especial de Grado Presentado ante la Ilustre Universidad Central de*.
- Cartuche, M. (2008). *Metodologías para el desarrollo de ontologías | Mackartuche's Weblog en WordPress.com*. Retrieved November 24, 2015, from <https://mackartuche.wordpress.com/2008/04/17/metodologias-para-el-desarrollo-de-ontologias/>
- Charria Ortiz, V. H., Sarsosa Prowesk, K. V., Uribe Rodríguez, A. F., López Lesmes, C. N., & Arenas Ortiz, F. (2011). *Definition and theoretical classification of academic, professional and work related competencies: The competencies of the Psychologist in Colombia*. *Psicología Desde El Caribe*, (28), 133–165. Retrieved from http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-417X2011000200007&lng=en&nrm=iso&tlng=es
- Churches, A. (n.d.). *TAXONOMÍA DE BLOOM PARA LA ERA DIGITAL*. Retrieved from <http://www.eduteka.org/pdfdir/TaxonomiaBloomDigital.pdf>
- Cideal. (2012). *Gestión por competencias*. Retrieved November 18, 2015, from <http://fortalecimiento.cideal.org/web/page/116>
- Corcho, O., Fernandez, M., Gomez, A., & Lopez-Cima, A. (2005). *Construcción de ontologías legales con la metodología METHONTOLOGY y la herramienta WebODE. Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*, 142–157. Retrieved from <http://oa.upm.es/5289/>
- Ecc. (2011). *Modelo de Innovación Docente Proyecto Piloto para el Mejoramiento de la Formación Basada en Competencias en la Escuela de Ciencias de la Computación Ficha Documento*, 1–18.
- Farias, N., Verduzco, J., Barbosa, H., Chavez, R., Guerrero, I., Nava, D., ... Chavez, D. (2010). *Medición de las Competencias del Estudiante a través de la evaluación del Aprendizaje. VI CONGRESO INTERNACIONAL EN INNOVACIÓN Y DESARROLLO TECNOLÓGICO*, (OCTOBER 2010), 0–9. <http://doi.org/10.13140/2.1.4519.0404>
- Fermoso, A., Salvador, A., & Sicilia, M. (2008). *Una ontología en OWL para la*

- representación semántica de objetos de aprendizaje. V Simposio Pluridisciplinar Sobre Diseño Y Evaluación de Contenidos Educativos Reutilizables , SPDECE08, 1–9.
- Fierro, F. (n.d.). *Ontología para la recomendación de enlaces de la Biblioteca Virtual de la UTPL*.
- Gallardo, K. (2009). *La Nueva Taxonomía de Marzano y Kendall: una alternativa para enriquecer el trabajo educativo desde su planeación*. Retrieved from http://www.cca.org.mx/profesores/congreso_recursos/descargas/kathy_marzano.pdf
- Gascueña, J. M., Fernández-caballero, A., & González, P. (2005). *Ontologías del modelo del alumno y del modelo del dominio en sistemas de aprendizaje adaptativos y colaborativos*. VI Congreso Interacción Persona Ordenador, Universidad de Granada. Retrieved from <http://www.aipo.es/>
- Gonzalez, C. (2012). □La presentación “METODOLOGÍA ÁGIL XP, ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA AUTOMATIZACIÓN DE ADMINISTRACIÓN DE CLIENTES, VEHÍCULOS, FACTURACIÓN,.” Retrieved March 31, 2016, from <http://slideplayer.es/slide/2273638/>
- Gruber, T. (1993). *Definición de ontología como especificación del conocimiento*. Retrieved November 24, 2015, from <http://elies.rediris.es/elies18/531.html>
- Guzmán, J., Bonilla, M., & Durley, I. (2012). *Metodologías y métodos para la construcción de ontologías*. *Scientia et Technica Año XVII, No 50 Universidad Tecnológica de Pereira*, (50), 133–140. Retrieved from <file:///home/newkrux/Dialnet-MetodologiasYMetodosParaLaConstruccionDeOntologias-4316679.pdf>
- León, R., Largo, E., & Tuza, E. (2014). *Web semántica aplicado a la carrera de Ingeniería en Sistemas*. *Web Semántica Con JENA*. Retrieved from <http://es.slideshare.net/RicharLen/web-semantica-con-jena>
- Lévy-Leboyer, C. (2000). *Gestión de las competencias, 3*. Retrieved from <http://cidseci.dgsc.go.cr/datos/Gestion de las competencias-Claude Levy.pdf>
- Maldonado, B. (2006). *Educación Superior por competencias*. Retrieved from <http://es.slideshare.net/rafaelallaico/educacin-superior-por-competencias>
- Ñauta, F., & Escandón, M. A. (2011). *Estudio, diseño e implementación de un buscador semántico*. Retrieved from <http://dspace.ucuenca.edu.ec/handle/123456789/629>
- Paquette, G. (2007). *An Ontology and a Software Framework for Competency Modeling and Management Competency in an Instructional Engineering Method (MISA)*. *Educational Technology & Society*, 10, 1–21.
- Perez, C., & Sepulved, M. F. (2008). “*Taxonomia de Benjamin Bloom*” en *WordPress.com*. Retrieved November 10, 2015, from <https://mafrita.wordpress.com/>
- Rafael, A. (2011). *AZUAY Diploma Superior en Educación Universitaria por Competencias*.
- Reina, M. J. (2008). *Universidad de Granada Tesis doctoral*, 50–51.
- Sarango, P., Alvarado, R., & Villa, W. (n.d.). *Proceso de adaptación de una Wiki tradicional a una Wiki Semántica*.
- Silva-Sprock, A., Lopez, M., Montañó, N., Miguel, V., Ramos, L., & Virllarroel, O. (2007). *Hacia Un Proceso De Ingeniería Del Conocimiento En La Creación De La Ontología De Ambar*, 6.
- Soto-Fuentes, P., Reynaldos-Grandón, K., Martínez-Santana, D., & Jerez-Yáñez, O. (2014). *Competencias para la enfermera/o en el ámbito de gestión y administración: desafíos actuales de la profesión*. *Aquichan*, 14(1), 79–99. <http://doi.org/10.5294/2684>

Tobón, S., Pimienta, J. H., & García, J. A. (2010). SECUENCIAS DIDÁCTICAS: aprendizaje y evaluación de competencias.

UCC. (2014). Guía para la Y SUPRESIÓN de programas académicos. Colombia. Retrieved from <https://www.ucatolica.edu.co/portal/wp-content/uploads/adjuntos/resoluciones/vicepresidencia-019-14-anexo-1.pdf>

UTPL. (2011). Plan Estratégico de la UTPL.

UTPL. (2012). Oferta Académica UTPL. Retrieved October 28, 2015, from <http://www.utpl.edu.ec/sites/default/files/oferta-academica-sept12-feb13.pdf>

ANEXOS

ANEXO 1

En este anexo se muestra la tabla de las macro competencias con sus respectivas competencias específicas de la titulación de sistemas informáticos y computación.

Tabla 21. Competencias específicas de la titulación de Sistemas Informáticos y Computación.

Macro competencias	Competencias Específicas
Analizar, diseñar e implementar soluciones de ingeniería basadas en TICs	Analizar problemas de programación y plantear soluciones mediante métodos computacionales.
	Implementar aplicaciones a partir de especificaciones y modelos de software utilizando estándares de documentación y de programación.
	Usar Librerías, API'S y Frameworks para la implementación de soluciones integrales.
	Construir modelos de software adecuados que permitan validar un producto previo a su implementación.
	Analizar, diseñar e instalar redes de telecomunicación.
	Diagnosticar y solucionar problemas relacionados con la comunicación de dispositivos y servicios de red e internet.
	Diseñar y gestionar redes de computadoras.
	Instalar redes de computadoras y operar dispositivos e infraestructura de telecomunicaciones.
	Diseñar, implementar y administrar sistemas de base de datos.
	Administrar centros de comunicación y datos (servidores y aplicaciones).
Gestionar y fiscalizar proyectos de TI	Definir, planificar y controlar proyectos de TI.
	Elaborar presupuestos y estimaciones de alcance, costo y tiempo en proyectos de TI.
	Seleccionar, reclutar y hacer <i>mentoring</i> a equipos de trabajo en proyectos de TIC.
	Desarrollar auditorías de productos y procesos vinculados a la gestión de TIC.
	Auditar la operación de TI en base a las buenas prácticas de los marcos de referencia (COBIT, ITIL, RISKIT, ISO27000).
Colaborar en proyectos de investigación en el área de ciencias de la computación	Buscar y seleccionar información, explorar métodos que permitan enfocar problemas relacionados a ciencias de la computación y avanzar en el proceso de su solución (estado del arte).
	Asimilar experiencias, conocimientos y herramientas para planificar o revisar sus estrategias de investigación en el área de ciencias de la computación.
	Plantear propuestas innovadoras de solución adecuadas a preguntas de investigación en áreas específicas de ciencias de la computación.
	Mejorar la autonomía intelectual ante procesos de ciencia y tecnología en el campo de las TIC.
Transferir conocimientos	Desarrollar, investigar y operar herramientas de gestión, análisis,

en el área de TIC y generar emprendimientos de base tecnológica	minería y manejo de datos e información.
	Escribir planes de negocio mediante la aplicación de conocimientos de TI, aplicada a problemas específicos.
	Crear emprendimientos en el área de TI de acuerdo a las normas vigentes.
	Obtener certificación internacional en un tema específicos de TI (Oracle, Cisco, Microsoft, Linux, PMP; ITIL, COBIT).
	Explorar y evaluar soluciones alternativas de tecnologías de la información y las comunicaciones, y emplearlas en la gestión y mejora de procesos empresariales.

Materias Troncales de Titulación

Tabla 22. Materias Troncales de Titulación

Materias	Ciclo	Créditos
Lógica de la Programación	1	4
Fundamentos Informáticos	1	3
Fundamentos de Programación	2	4
Estructura de datos y algoritmos	3	4
Programación de algoritmos	3	4
Programación Avanzada	4	4
Electrónica Digital	4	4
Arquitectura de Computadores	5	4
Fundamentos de Base de Datos	5	4
Fundamentos de Ingeniería de Software	6	4
Sistemas Operativos	6	4
Base de Datos Avanzada	6	4
Ingeniería de Requisitos	7	4

Materias	Ciclo	Créditos
Teoría de Autómatas y Compiladores	7	4
Fundamentos de Redes y Telecomunicaciones	7	4
Ingeniería Web	8	4
Gestión de tecnologías de la información	8	3
Redes y Sistemas Distribuidos	8	4
Arquitectura de Aplicaciones	9	4
Inteligencia Artificial	9	4
Arquitectura y Seguridad de Redes	9	4
Arquitectura y Computación Paralela	9	4
Procesos de Ingeniería de Software	10	4
IA Avanzado	10	4
Sistemas Basados en el Conocimiento	10	4
Auditoría Informática	10	4

Fuente: Elaboración propia

ANEXO 2

En este anexo se especifica cada uno de los niveles de habilidad con sus respectivos verbos relacionados y este a su vez con sus sinónimos, como se muestran a continuación en las siguientes tablas.

Tabla 23. Corpus del nivel habilidad Crear.

NIVEL HABILIDAD	VERBOS RELACIONADOS	SINONIMOS
Crear		inventar, engendrar, procrear, imaginar, concebir, instaurar, fundar, establecer, instituir, legitimar, nombrar, designar, elegir.
	Adaptar	aclimatar, acomodar, acondicionar, adecuar, ajustar, habituar, acostumbrar, cambiar, arreglar, transformar
	Arreglar	ordenar, disponer, acordar, decidir, resolver, solventar, ajustar, acomodar, acondicionar, adecuar, conciliar, coordinar, regular, normalizar, metodizar, sistematizar, clasificar, regularizar, reparar, recomponer, reformar, restaurar, desenredar, desembrollar, corregir, enmendar, retocar, pulir, remendar, empastar, mejorar, adecentar, limpiar, sanear
	Cambiar*	canjear, intercambiar, trocar, permutar, conmutar, evolucionar, innovar, renovar, distorsionar, diversificar, modificar, reformar,
	Coleccionar	recopilar, compilar, reunir, seleccionar
	Combinar	componer, concertar, coordinar, acoplar, agregar, compaginar, conjuntar, juntar, unir, casar

	Compilar	recopilar, reunir, agrupar, inventariar, coleccionar, codificar, clasificar
	Componer*	hacer, reparar, armar, apañar, restaurar, remendar, rectificar, confeccionar, crear, arreglar, corregir, organizar, constituir, formar, armar, integrar, acomodar, adornar, acicalar, arreglar, ataviar, emperifollar, engalanar, maquillar, embellecer, aviar
	Concluir*	terminar, acabar, finalizar, finiquitar, rematar, ultimar, saldar, liquidar, agotar, apurar, consumir, clausurar, deducir, inferir, razonar, discurrir, colegir, derivar.
	Construir*	edificar, fabricar, erigir, levantar, cimentar, obrar
	Deducir	descontar, rebajar, disminuir, restar, colegir, inferir, suponer, derivar, desprender, resultar, seguir, concluir.
	Definir*	determinar, fijar, precisar, aclarar, puntualizar, concretar, constreñir, delimitar, detallar, especificar, explicar
	Desarrollar*	extender, desenvolver, desenrollar, desliar, desplegar, acrecentar, aumentar, incrementar, amplificar, fomentar, crecer, progresar, evolucionar, prosperar, perfeccionar,
	Desenvolver	desliar, desempaquetar, desplegar, desdoblar, extender, aclarar, averiguar, escudriñar, descubrir, aumentar, prosperar, ampliar, expandir, manejarse, valerse, apañar
	Dirigir	acaudillar, presidir, encabezar, regir, capitanear, liderar, conducir, gobernar, guiar, administrar, controlar, manejar, mandar, coordinar, aconsejar, adiestrar, apuntar, canalizar, educar, orientar, encarrilar, encauzar, enderezar, enfocar, mediatizar, enviar, destinar, llevar, trasladar, ir, marchar, encaminar
	Diseñar	proyectar, trazar, esbozar, esquematizar, abocetar, delinear, plantear
	Elaborar	confeccionar, fabricar, hacer, producir, realizar, transformar, proyectar
	Eliminar	descartar, excluir, suprimir, expeler, quitar.
	Ensamblar*	acoplar, encajar, empalmar, enlazar, articular, unir, juntar
	Escribir*	transcribir, manuscibir, copiar, anotar, firmar, rubricar, autografiar, trazar, caligrafiar, mecanografiar, taquigrafiar, componer, redactar, expresar, exponer, crear, novelar, publicar, editar, imprimir
	Escoger*	elegir, seleccionar, preferir, distinguir, separar, apartar, entresacar, aislar, nombrar, optar.
	Establecer	fundar, constituir, instaurar, erigir, crear, fijar, poner, decretar, implantar, disponer, afincar, asentar, instalar, radicar, avecindar, habitar, residir
	Especificar	determinar, establecer, precisar, delimitar, detallar, enumerar, diferenciar, pormenorizar
	Examinar*	investigar, comprobar, reconocer, inspeccionar, averiguar, indagar, observar, analizar, estudiar, verificar, mirar, ojear, investigar, comprobar, reconocer, inspeccionar, averiguar, indagar, observar, analizar, estudiar, verificar, mirar, ojear, concursar, opositar, concurrir, someterse, pasar, sufrir.
	Fabricar*	manufacturar, confeccionar, elaborar, producir, industrializar, transformar, crear, forjar, realizar
	Formular	enunciar, manifestar, decir, pronunciar, exponer, expresar, proponer, recetar, prescribir
	Gestionar	tramitar, diligenciar, ejecutar, dirigir, administrar, resolver, encargar, mandar, negociar
	Generalizar	universalizar, pluralizar, difundir, diversificar, divulgar, extender, publicar
	Hacer	producir, formar, crear, engendrar, concebir, originar, elaborar, descubrir, inventar, componer, realizar, trabajar, terminar, construir, emprender, fabricar, confeccionar, obrar, proceder, ejercer, ejecutar, representar, establecer, perpetrar, urdir, habituar, acostumbrar

	Hipotetizar	
	Idear	imaginar, concebir, pensar, discurrir, trazar, urdir, proyectar, planear, inventar, esbozar, reflexionar, suponer, sospechar, recordar
	Imaginar	fantasear, soñar, figurarse, idear, comprender, representar, sentir, percibir, pensar, reflexionar, suponer, sospechar, presentir, recelar, recordar, evocar, idear, concebir, planear, proyectar, crear, esbozar, calcular, descubrir, improvisar
	Implementar	imaginar, idear, concebir, planear, ingeniar, descubrir, discurrir, encontrar, hallar, pensar, proyectar, crear, fingir, engañar, mentir, urdir, fantasear, falsear
	Inventar	imaginar, idear, concebir, planear, ingeniar, descubrir, discurrir, encontrar, hallar, pensar, proyectar, crear, fingir, engañar, mentir, urdir, fantasear, falsear
	Integrar	componer, constituir, incorporar, afiliarse, añadir, completar, formar, participar, sumar, totalizar, entrar
	Interpretar*	comentar, glosar, explicar, analizar, aclarar, deducir, descifrar, entender, traducir, representar, actuar, declamar
	Maximizar	
	Minimizar	
	Modificar*	cambiar, mudar, alterar, variar, rectificar, reformar, innovar, transformar, enmendar
	Organizar*	estructurar, fundar, instituir, arreglar, constituir, disponer, establecer, instaurar, regular, regularizar, crear, ordenar.
	Originar	causar, producir, crear, hacer, formar, ocasionar, establecer, criar, engendrar, generar, provocar, provenir, dimanar, derivarse, resultar, trascender, emanar, descender, salir, brotar, germinar.
	Planear*	proyectar, planificar, programar, preparar, organizar, diseñar, estructurar, trazar, calcular, fraguar, pensar, idear, concebir, volar, descender, deslizar.
	Preparar*	disponer, organizar, prevenir, acondicionar, arreglar, proyectar, proporcionar, proveer, gestar, fraguar, acercar, desarrollar, elaborar, planificar, medir, intentar, comenzar, saber, conocer, capacitar, enseñar, entrenar, aprender, amañar, apañar, falsear, arreglar
	Prescribir	recetar, recomendar, disponer, determinar, ordenar, destinar, caducar, vencer, terminar, finalizar, extinguirse, concluir
	Proponer	sugerir, plantear, manifestar, expresar, exponer, enunciar, formular, opinar, insinuar, prometer, recomendar, presentar, empeñarse, pretender, intentar, aspirar, tratar, procurar, desear, proyectar, ambicionar, decidir, gestionar
	Producir*	hacer, fabricar, trabajar, crear, elaborar, realizar, obtener, rendir, rentar, reportar, ocasionar, provocar, resultar, originar, suceder
	Reconstruir	reedificar, restaurar, reparar, arreglar, recobrar, rehacer, repetir, reproducir
	Resolver*	solucionar, aclarar, averiguar, descifrar, solventar, determinar, sentenciar, decidir, despachar, diligenciar, decretar, ventilar, zanjar
	Reunir	unir, juntar, fusionar, concentrar, agrupar, aglutinar, amontonar, congregarse, aglomerar, acumular, entrevistar, concentrar, concurrir, encontrar
	Suceder	ocurrir, acontecer, acaecer, sobrevenir, venir, efectuar, producir, desarrollar, realizar, cumplir, seguir, reemplazar, sustituir, relevar, suplantar, continuar, proseguir, heredar, perpetuar, prolongar
	Suponer	conjeturar, sospechar, creer, figurar, imaginar, pensar, considerar, significar, constituir, evidenciar, implicar, entrafarse, conllevar
	Teorizar	suponer, especular, reflexionar, idealizar, sistematizar
	Trazar*	representar, diseñar, dibujar, delinear, bosquejar, esbozar, proyectar, formular, escribir, describir, discurrir, explicar, relacionar, elucubrar, maquinar, indicar, marcar, rayar.
	Recopilar	seleccionar, coleccionar, compilar, reunir, compendiar

	Redactar*	escribir, componer, expresar, describir, manifestar, reflejar, concebir, narrar, transcribir
	Sintetizar	reducir, resumir, abreviar, simplificar, compendiar, condensar, recopilar, extractar, acortar, disminuir

Fuente: Elaboración propia.

Tabla 24. Corpus del nivel habilidad Crear.

NIVEL HABILIDAD	VERBOS RELACIONADOS	SINONIMOS
Evaluar		tasar, valorar, calcular, determinar, estimar, justipreciar
	Aceptar	admitir, confesar, reconocer, aprobar, aplaudir, acceder, consentir, tolerar, soportar, transigir, someter, comprometer, obligar, recibir, recoger, tomar
	Aportar	contribuir, dar, colaborar, cooperar, ayudar, auxiliar, participar, concurrir, asistir, tributar, pagar, añadir, presentar
	Apreciar	querer, estimar, respetar, amar, reputar, considerar, distinguir, evaluar, justipreciar, juzgar, calificar, tasar, valorar, valuar
	Aprobar	admitir, aceptar, pasar, acceder, autorizar, consentir, asentir, permitir, certificar, avalar, afirmar, ratificar, reconocer, acreditar, garantizar, elogiar, alabar, celebrar, aplaudir
	Argumentar	razonar, argüir, analizar, colegir, inferir, deducir, demostrar, distinguir, discutir, impugnar, discrepar, objetar, contradecir, refutar, replicar, ergotizar
	Avaluar	Avaluar
	Categorizar*	
	Clasificar*	ordenar, organizar, separar, catalogar, encasillar, coordinar
	Calificar	apreciar, autorizar, caracterizar, conceptuar, considerar, graduar, aplicar, adjetivar, declarar, llamar, nombrar, nominar, bautizar, tildar, clasificar, definir, puntuar, acreditar, ennoblecer, ilustrar
	Comparar*	equiparar, parangonar, confrontar, cotejar, relacionar, compulsar, examinar, comprobar, verificar, contrastar
	Concluir*	terminar, acabar, finalizar, finiquitar, rematar, ultimar, saldar, liquidar, agotar, apurar, consumir, clausurar, deducir, inferir, razonar, discurrir, colegir, derivar
	Considerar	examinar, reflexionar, meditar, reparar, pensar, sopesar, estudiar, estimar, juzgar, observar, valorar, respetar, estimar, reputar, apreciar, honrar
	Criticar	analizar, juzgar, examinar, enjuiciar, opinar, censurar, murmurar, reprochar, reprobar, vituperar, reprender
	Debatir	disputar, polemizar, contender, discutir, litigar, tratar, deliberar, controvertir, rivalizar, porfiar, combatir, batallar, luchar, pelear, forcejear
	Decidir	resolver, determinar, disponer, sentenciar, arbitrar, decretar, dictaminar, dilucidar, fallar, atrever, osar, animar, lanzar, arriesgar
	Defender	amparar, proteger, resguardar, preservar, disculpar, exculpar, excusar, justificar, apoyar, auxiliar, sostener
	Determinar	acordar, decidir, decretar, definir, diagnosticar, disponer, resolver, fallar, mandar, ordenar, concluir, delimitar, describir, deslindar, encuadrar, especificar, fijar, limitar, precisar, señalar

	Descubrir*	revelar, manifestar, mostrar, exhibir, exteriorizar, revelar, inventar, investigar, crear, hallar, destapar, desnudar, desvestir, sorprender, atrapar, coger, pillar, pescar, desenmascarar, explorar, conquistar, colonizar
	Disputar	debatir, polemizar, argüir, cuestionar, discrepar, discutir, disentir, porfiar, contender, batallar, combatir, chocar, litigar, luchar, competir
	Dar importancia	
	Deducir*	descontar, rebajar, disminuir, restar, colegir, inferir, suponer, derivar, desprenderse, resultar, seguir, concluir
	Emitir un juicio	
	Estimar	valorar, apreciar, evaluar, tasar, enjuiciar, conceptuar, juzgar, creer, considerar, pensar, querer, amar, respetar, honrar, admirar
	Escoger*	elegir, seleccionar, preferir, distinguir, separar, apartar, entresacar, aislar, nombrar, optar
	Explicar*	
	Influenciar	Influir
	Influir	influir, actuar, predominar, intervenir, apoyar, ayudar, contribuir, respaldar, mediar, terciar
	Interpretar*	comentar, glosar, explicar, analizar, aclarar, deducir, descifrar, entender, traducir, representar, actuar, declamar
	Juzgar	conceptuar, considerar, opinar, estimar, pronunciarse, creer, valorar, fallar, sentenciar, dictaminar, enjuiciar, arbitrar, resolver, decidir, decretar, establecer
	Justificar	demostrar, alegar, acreditar, razonar, documentar, evidenciar, explicar, testimoniar, aducir, argumentar, disculpar, defender, librar, apoyar, respaldar, salvar, proteger, excusar
	Lograr	conseguir, obtener, alcanzar, conquistar, ganar, embolsar
	Marcar	señalar, señalar, sellar, signar, timbrar, contrastar, firmar, rubricar, rotular, etiquetar, estampillar, estigmatizar, caracterizar, distinguir, destacar, personalizar, singularizar, recalcar, perfilar, impresionar, conmocionar, afectar, traumatizar, puntuar, apuntar, anotar
	Medir	calcular, calibrar, evaluar, valorar, contar, mesurar, cronometrar, arquear, establecer, determinar, comprobar, contener, moderar, refrenar, dominar, competir, enfrentar
	Opinar	juzgar, valorar, pensar, exponer, decir, creer, considerar, estimar
	Percibir	advertir, notar, observar, ver, distinguir, divisar, percatarse, reparar, oír, escuchar, oler, gustar, cobrar, ingresar, recaudar, recibir, comprender, entender, discernir, intuir
	Premiar	galardonar, recompensar, honrar, laurear, coronar, estimular, homenajear, condecorar, retribuir
	Priorizar	
	Probar	acreditar, justificar, atestiguar, demostrar, evidenciar, convencer, certificar, gustar, catar, saborear, degustar, paladear, ensayar, comprobar, intentar, tratar, procurar
	Recomendar	advertir, aconsejar, avisar, indicar, insinuar, sugerir, invitar, interceder, influir, amparar, proteger, apoyar, apadrinar, enchufar, favorecer, mediar, ensalzar, alabar, elogiar
	Reglamentar	regular, normalizar, preceptuar, ordenar, establecer, legalizar, sistematizar

	Reprobar	censurar, desaprobar, criticar, reprochar, condenar, vituperar, suspender, catear, descalificar, revolver
	Seleccionar*	escoger, elegir, separar, apartar, destacar, distinguir, optar, preferir
	Significar	representar, denotar, figurar, ser, encarnar, simbolizar, sobrentender, personificar, evidenciar, entrafñar, implicar, suponer, equivaler, valer, importar, expresar, decir, manifestar, sobresalir, destacar, distinguir, señalar, descollar
	Valorar	evaluar, tasar, valuar, calcular, apreciar, estimar, justipreciar
	Valuar	valorar
	Elegir*	seleccionar, preferir, optar, escoger, entresacar, acotar, adoptar, apartar, nombrar, proclamar, votar, sacar
	Predecir	adivinar, vaticinar, pronosticar, augurar, presentir, revelar, conjeturar, imaginar
	Apoyar	secundar, defender, ayudar, proteger, auxiliar, amparar, animar, avalar, asistir, patrocinar, estimular, favorecer, respaldar, reafirmar, corroborar, confirmar, descansar, descargar, arrimar, gravitar, dosar, asentar, fundar, basar, armar, hincar, apuntalar, cargar, soportar, sustentar, sostener, reclinar, acodar, sujetar, afirmar, reafirmar, confirmar, corroborar

Fuente: Elaboración propia.

Tabla 25. Corpus del nivel habilidad Aplicar.

NIVEL HABILIDAD	VERBOS RELACIONADOS	SINONIMOS
Aplicar		adaptar, superponer, adherir, Sobreponer, acomodar, poner, arrimar, fijar, pegar, sujetar, clavar, acercar
	Calcular	computar, contar, meditar, prever, tasar, valorar, evaluar,
	Cambiar*	canjear, intercambiar, trocar, permutar, conmutar, evolucionar, innovar, renovar, distorsionar, diversificar, modificar, reformar,
	Comprobar	confirmar, confrontar, verificar, constatar, cerciorarse, compulsar, cotejar, revisar, examinar, escrutar, repasar
	Computar	calcular, contar, medir, evaluar, tasar, valorar, tantear
	Contrastar*	comparar, constatar, confrontar, verificar, comprobar, diferir, disentir, discordar, desentonar, chirrear, chocar, resaltar
	Construir*	edificar, fabricar, erigir, levantar, cimentar, obrar
	Convertir*	mudar, cambiar, transformar, trocar, transfigurar
	Demostrar	declarar, manifestar, revelar, señalar, indicar, exponer, exhibir, probar, razonar, argumentar, evidenciar, justificar, enseñar, mostrar
	Desarrollar*	extender, desenvolver, desenrollar, desliar, desplegar, acrecentar, aumentar, incrementar, amplificar, fomentar, crecer, progresar, evolucionar, prosperar, perfeccionar.
	Dibujar*	pintar, delinear, sombrear, describir, esbozar, esquematizar, perfilar, retratar, tatuar, trazar, rayar, bosquejar.
	Dramatizar	exagerar, impresionar, emocionar, hinchar, desorbitar
	Ejemplificar	ejemplarizar
	Emplear	usar, utilizar, gastar, disponer, destinar, aplicar, dedicar, manejar, valerse, servirse, colocar, situar, ocupar, enchufar, acomodar
	Ensamblar*	acoplar, encajar, empalmar, enlazar, articular, unir, juntar
	Entrevistar	entrevistar, interrogar, conversar, dialogar, departir
	Escoger*	elegir, seleccionar, preferir, distinguir, separar, apartar, entresacar, aislar, nombrar, optar

	Estimar*	valorar, apreciar, evaluar, tasar, enjuiciar, conceptuar, juzgar, creer, considerar, pensar, querer, amar, respetar, honrar, admirar.
	Extrapolar*	
	Fabricar*	manufacturar, confeccionar, elaborar, producir, industrializar, transformar, crear, forjar, realizar.
	Ilustrar*	instruir, educar, cultivar, formar, enseñar, preparar, aclarar, explicar, esclarecer, divulgar, informar, grabar, dibujar, estampar, iluminar.
	Interpolar	interponer
	Interpretar*	
	Manipular	manejar, utilizar, emplear, operar, actuar, tocar, sobar, manosear, maniobrar, mangonear, dirigir, mandar, ordenar, adulterar, falsificar, viciar
	Modelar	moldear, conformar, configurar, cincelar, tallar, esculpir, fomar, crear, componer, ajustar, organizar
	Modificar	cambiar, mudar, alterar, variar, rectificar, reformar, innovar, transformar, enmendar
	Mostrar*	exhibir, enseñar, exponer, presentar, revelar, destapar, abrir, manifestar, evidenciar, ofrecer, indicar, señalar, advertir, orientar, aconsejar, sugerir, encaminar, guiar
	Operar	actuar, ejecutar, obrar, realizar, elaborar, ejercitar, manipular, efectuar, intervenir
	Organizar*	estructurar, fundar, instituir, arreglar, constituir, disponer, establecer, instaurar, regular, regularizar, crear, ordenar
	Planear*	proyectar, planificar, programar, preparar, organizar, diseñar, estructurar, trazar, calcular, fraguar, pensar, idear, concebir,
	Practicar	ejercer, ensayar, ejecutar, habituar, acostumbrar, avezar, foguear, usar, experimentar, adiestrar, trabajar
	Preparar*	disponer, organizar, prevenir, acondicionar, arreglar, proyectar, proporcionar, proveer, gestar, fraguar, avecinar, desarrollar, elaborar, planificar, medir, intentar, comenzar, saber, conocer, capacitar, enseñar, entrenar, aprender, amañar, apañar, falsear, arreglar
	Producir*	hacer, fabricar, trabajar, crear, elaborar, realizar, obtener, rendir, rentar, reportar, ocasionar, provocar, resultar, originar, suceder
	Programar	exponer, planificar, plantear, proyectar, sistematizar, esquematizar, bosquejar
	Realizar	hacer, ejecutar, desarrollar, efectuar, elaborar, producir, confeccionar, componer, proceder, concluir, cumplir, crear, liquidar, vender, saldar, materializar
	Redactar*	escribir, componer, expresar, describir, manifestar, reflejar, concebir, narrar, transcribir
	Relatar*	contar, narrar, referir, exponer, describir, puntualizar, reseñar
	Resolver*	solucionar, aclarar, averiguar, descifrar, solventar, determinar, sentenciar, decidir, despachar, diligenciar, decretar, ventilar, zanjar
	Traducir*	interpretar, verter, derivar, trasladar, descifrar, transcribir
	Transformar	alterar, cambiar, variar, modificar, mudar, trocar, desfigurar, desvirtuar, distorsionar, reformar, restaurar, elaborar, industrializar
	Trazar*	representar, diseñar, dibujar, delinear, bosquejar, esbozar, proyectar, formular, escribir, describir, discurrir, explicar, relacionar, elucubrar, maquinar, indicar, marcar, rayar
	Usar	utilizar, emplear, consumir, dedicar, disponer, manejar, explotar, servir, valer, acostumbrar, soler, frecuentar
	Utilizar	emplear, usar, manejar, servir, beneficiar, disfrutar, gastar, consumir, aprovechar, dedicar, destinar, aplicar

	Esbozar	bosquejar, trazar, diseñar, apuntar, perfilar, plantear, proponer, sugerir, insinuar
	Solucionar	solventar, resolver, reparar, arreglar, componer, remediar, concluir, satisfacer, arbitrar, enmendar, enderezar, perfeccionar, decidir, zanjar, corregir, salvar

Fuente: Elaboración propia.

Tabla 26. Corpus del nivel habilidad Comprender.

NIVEL HABILIDAD	VERBOS RELACIONADOS	SINONIMOS
Comprender		abarcar, adivinar, captar, compendiar, concebir, conocer, contener, discernir, encerrar, englobar, entender, enterar, incluir, integrar, interpretar, intuir, penetrar, vislumbrar
	Clasificar*	ordenar, organizar, separar, atalogar, encasillar, coordinar
	Comparar*	equiparar, parangonar, confrontar, cotejar, relacionar, compulsar, examinar, comprobar, verificar, contrastar
	Contrastar*	comparar, constatar, confrontar, verificar, comprobar, diferir, disentir, discordar, desentonar, chirrear, chocar, resaltar
	Convertir*	mudar, cambiar, transformar, trocar, transfigurar
	Dar ejemplo	
	Describir	delinear, dibujar, pintar, rayar, trazar, explicar, reseñar, detallar, especificar, pormenorizar
	Discutir	analizar, argüir, argumentar, estudiar, examinar, razonar, tratar, deliberar, controvertir, contradecir, refutar, debatir, rebatir, impugnar
	Distinguir*	diferenciar, discernir, deslindar
	Explicar*	aclarar, demostrar, desembrollar, solucionar, esclarecer, interpretar, dilucidar, argüir, comentar, declarar, definir, desarrollar, describir, elucidar, enseñar, ilustrar, justificar
	Expresar	manifestar, declarar, exteriorizar, reflejar, revelar, decir, hablar, opinar, significar
	Identificar*	reconocer, fichar, establecer, referir, determinar, detallar, reseñar, registrar, describir
	Ilustrar*	instruir, educar, cultivar, formar, enseñar, preparar, aclarar, explicar, esclarecer, divulgar, informar, grabar, dibujar, estampar, iluminar
	Informar	advertir, anunciar, indicar, avisar, comunicar, documentar, indagar, instruir, revelar, investigar, noticiar, notificar, participar, prevenir, reseñar, enterar, publicar
	Interpretar*	comentar, glosar, explicar, analizar, aclarar, deducir, descifrar, entender, traducir, representar, actuar, declamar
	Ordenar*	organizar, arreglar, clasificar, acomodar, acondicionar, adecentar, sistematizar, estructurar, coordinar, regular, reglamentar, normalizar
	Parafrasear	comentar, explicar, glosar, interpretar, ilustrar, revelar, imitar, remedar, parodiar, reproducir
	Reafirmar	ratificar, asegurar, aseverar, confirmar, consolidar
	Reconocer*	examinar, inspeccionar, investigar, observar, explorar, contemplar, tantear, considerar, mirar, ver, vigilar, admitir, acatar, aceptar, asentir
	Resumir	abreviar, compendiar, extractar, recapitular, reducir, sintetizar, condensar, simplificar
	Traducir*	interpretar, verter, derivar, trasladar, descifrar, transcribir, cambiar
	Revisar	reexaminar, repasar, inspeccionar, comprobar, investigar, reconocer, considerar, controlar, explorar, fiscalizar, verificar, observar
	Seleccionar*	escoger, elegir, separar, apartar, destacar, distinguir, optar, preferir

Fuente: Elaboración propia.

Tabla 27. Corpus del nivel habilidad Recordar.

NIVEL HABILIDAD	VERBOS RELACIONADOS	SINONIMOS
Recordar	Anunciar	comunicar, declarar, advertir, avisar, divulgar, informar, notificar, publicar, proclamar, revelar
	Bosquejar	esbozar, proyectar, esquematizar, planear, planificar, diseñar, abocetar, delinear
	Citar	llamar, avisar, notificar, convocar, emplazar, aludir, mencionar, referir, invocar, probar, alegar, confirmar
	Contar	relatar, narrar, referir, detallar, expresar
	Copiar	transcribir, imitar
	Definir	determinar, fijar, precisar, aclarar, puntualizar, concretar, constreñir, delimitar, detallar, especificar, explicar
	Deletrear	silabear, pronunciar
	Decir	hablar, manifestar, declarar, expresar, exponer, mencionar, formular, anunciar, enunciar, enumerar, observar, recitar, contar, hablar, parlotear, articular
	Encontrar	descubrir, localizar, hallar, tropezar, topar, chocar, acertar, atinar, aparecer, converger, inventar
	Enlistar	
	Escoger*	elegir, seleccionar, preferir, distinguir, separar, apartar, entresacar, aislar, nombrar, optar
	Escribir	componer, redactar, expresar, exponer, crear, novelar, publicar, editar, imprimir
	Etiquetar	marcar, clasificar
	Identificar*	reconocer, fichar, establecer, referir, determinar, detallar, reseñar, registrar, describir
	Indicar	señalar, advertir, guiar, aconsejar, apuntar, marcar, mostrar, orientar, observar, sugerir, encaminar, exhortar
	Leer	estudiar, descifrar, interpretar, repasar, analizar, ojear
	Listar	enumerar, inventariar, catalogar
	Localizar	situar, ubicar, colocar, emplazar, fijar, hallar, encontrar, averiguar, descubrir
	Nombrar	mentar, denominar
	Nominar	nombrar, calificar, llamar, votar
	Mostrar*	exhibir, enseñar, exponer, presentar, revelar, destapar, abrir, manifestar, evidenciar, ofrecer
	Recitar	declamar, entonar, enunciar, cantar, narrar, pronunciar, relatar, contar, referir, exponer
	Recordar	evocar, recordar, conmemorar, acordarse, mencionar, aludir, invocar, recapitular
	Registrar	inspeccionar, cachear, rastrear, rebuscar, hurgar, reconocer, chequear, asentar, anotar, apuntar, inscribir, patentar, matricular, inventariar.
	Relatar*	contar, narrar, referir, exponer, describir, puntualizar, reseñar
	Repetir	reiterar, insistir, reincidir, volver, rehacer, duplicar
	Reportar	contener, refrenar, frenar, aplacar, apaciguar, calmar, sosegar
	Reproducir	copiar, imitar, calcar, repetir, duplicar, representar, falsificar, reimprimir, propagarse, multiplicarse, procrear, proliferar, fecundar, retoñar, desarrollarse
	Rotular	titular, inscribir, marcar, dibujar, diseñar, publicar
	Parear	

	Seleccionar*	escoger, elegir, separar, apartar, destacar, distinguir, optar, preferir
	Subrayar	rayar, marcar, trazar, señalar, acentuar, recalcar, resaltar, enfatizar, insistir
	Organizar*	estructurar, fundar, instituir, arreglar, constituir, disponer, establecer, instaurar, regular, regularizar, crear, ordenar
	Duplicar	doblar, desdoblar, redoblar, repetir, plagiar, copiar, reproducir, transcribir
	Enumerar	contar, listar, numerar, mencionar, exponer, especificar, detallar, enunciar, catalogar, inventariar
	Memorizar	recordar, retener, empollar
	Ordenar*	organizar, arreglar, clasificar, acomodar, acondicionar, adecuar, sistematizar, estructurar, coordinar, regular, reglamentar, normalizar
	Reconocer	examinar, inspeccionar, investigar, observar, explorar, contemplar, tantear, considerar, mirar, ver, vigilar, admitir, acatar, aceptar, asentir
	Relacionar	enlazar, vincular, conectar, unir, encadenar, coordinar

Fuente: Elaboración propia.

ANEXO 3

Mapa conceptual de la verificación de competencias académicas

Aquí se especifica la creación del mapa conceptual de la verificación de competencias académicas en base al nivel de habilidad.

Para la construcción del mapa conceptual se lo realizará en la herramienta **CmapTools⁵ Coe v5.0** y como ayuda se agregó unas plantillas, a continuación se muestra como agregarlas a CmapTools.

Agregar Plantillas al CmapTools COE

Para hacer uso de las plantillas se deben seguir los siguientes pasos:

1. Descargar las plantillas, las mismas que pueden ser descargadas de los siguientes enlaces:
<http://www.mediafire.com/?qy77ig4f44r43pl>
2. Una vez descargadas éstas deben ser copiadas en la carpeta de *My Cmaps*. "D:\Mis documentos\My Cmaps".
3. Aquí se descomprime el archivo.
4. Vale recalcar que la carpeta *My Cmaps*, se crea al momento de instalar CmapTools y la ubicación depende de dónde se lo instale.

⁵ http://cmapdownload.ihmc.us/coe/Web_InstallersV5.0/install.htm

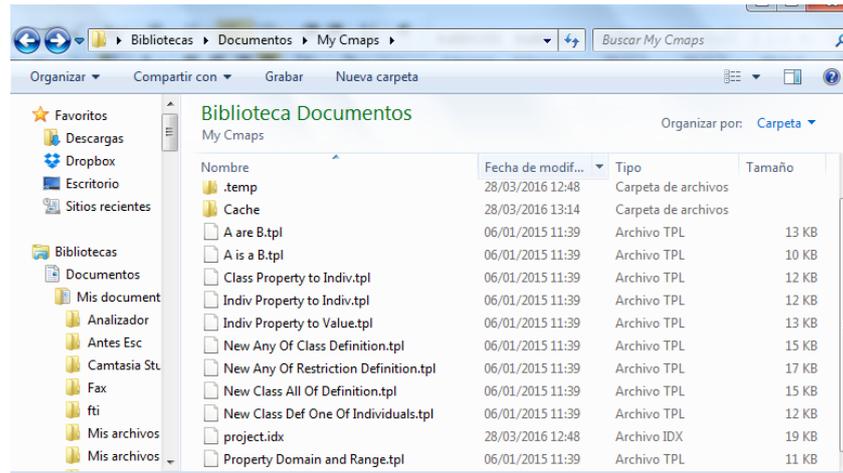


Figura 47. Carpeta donde se debe copiar la plantillas.
Fuente: Elaboración propia.

También se pueden agregar las plantillas, abriendo CmapTools, luego ir a la ventana “Vistas CmapTools”

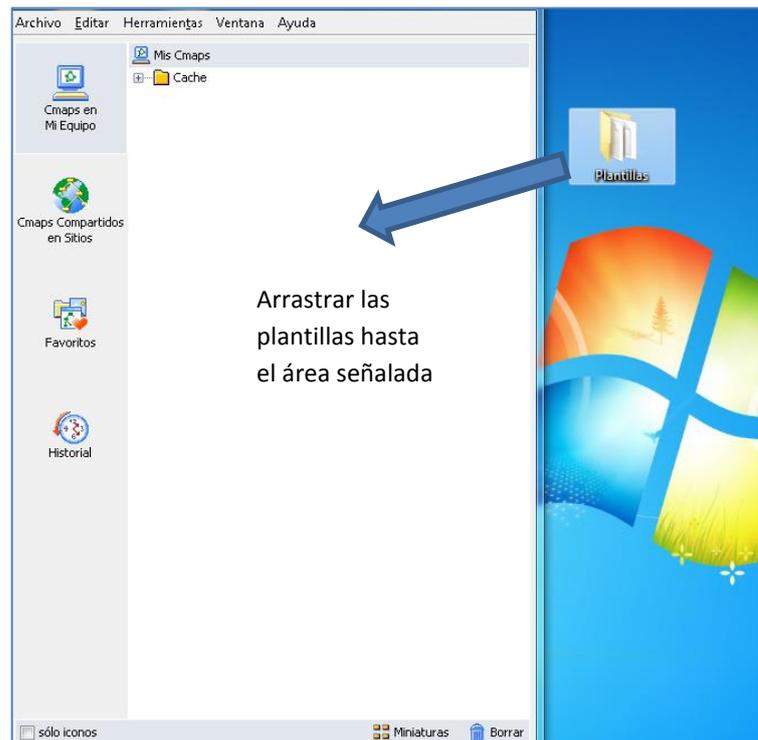


Figura 48. Ingresar las plantillas COE
Fuente: Elaboración Propia.

4. Ya agregadas las plantillas se las puede comenzar a usar, se encuentran de la siguiente forma en CmapTolls.

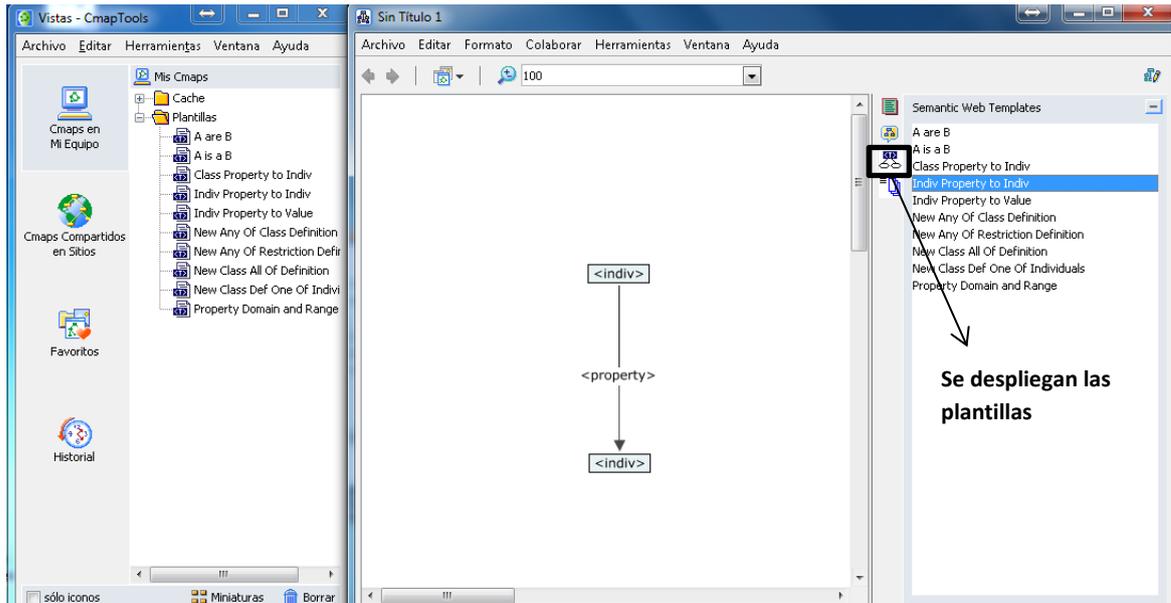
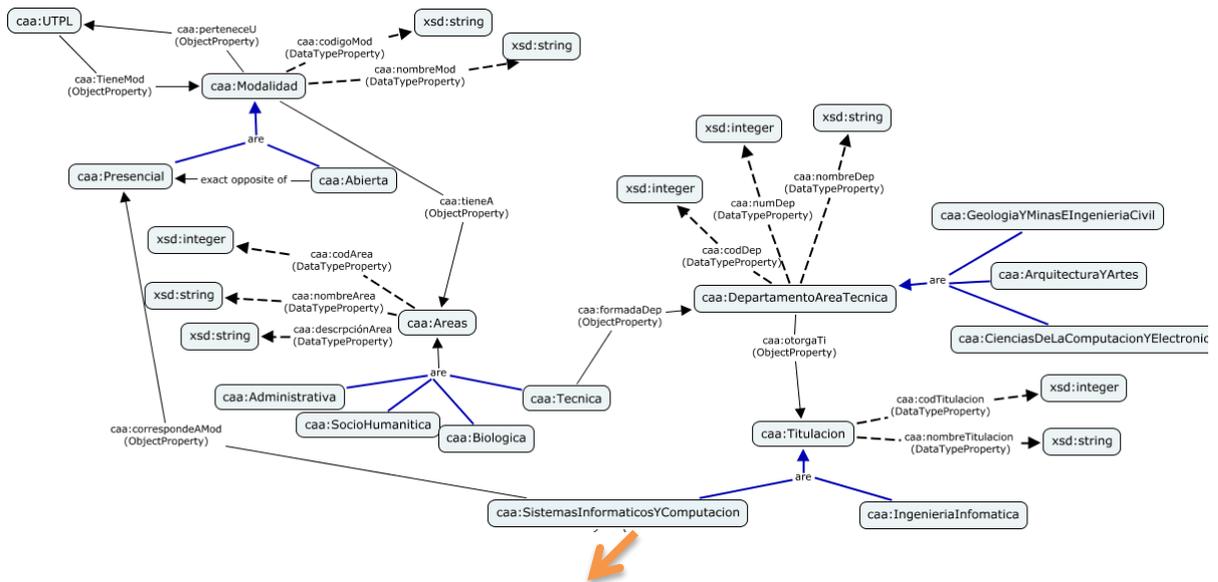
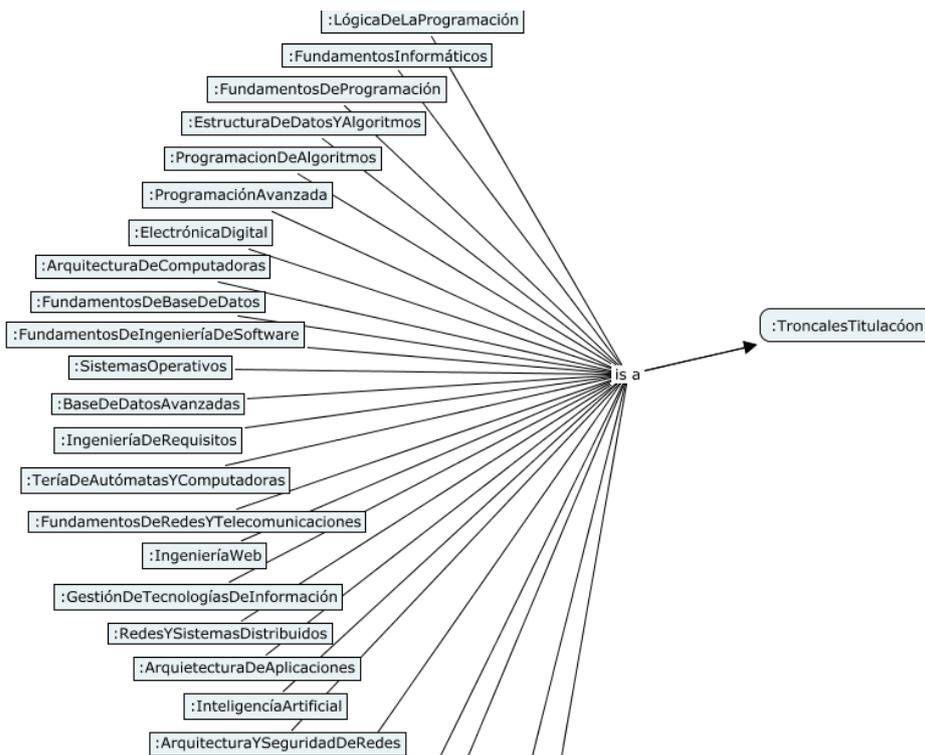
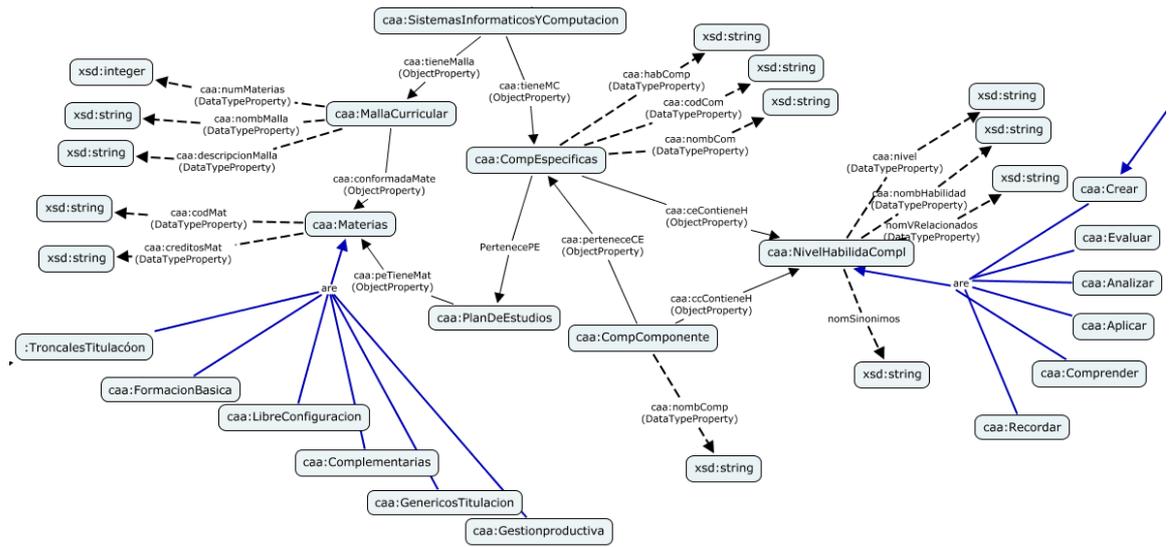


Figura 49. Plantillas COE agregadas.
Fuente: Elaboración propia.

ANEXO 4

Taxonomía finalizada de verificación de competencias académicas






```

</owl:Ontology>
<owl:Class rdf:ID="CompetenciasComponenteAcademico"/>
<owl:Class rdf:ID="TitulaciÃ³n"/>
<owl:Class rdf:ID="Tecnica">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Areas"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Analizar">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="NivelHabilidadComplejidad"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GenericasTitulacion">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Materias"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TroncalesDeTitulacion">
  <rdfs:subClassOf rdf:resource="#Materias"/>
</owl:Class>
<owl:Class rdf:ID="Presencial">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Modalidad"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="MallaCurricular"/>
<owl:Class rdf:ID="Administrativa">
  <rdfs:subClassOf rdf:resource="#Areas"/>
</owl:Class>
<owl:Class rdf:ID="Socio-Humanistica">
  <rdfs:subClassOf rdf:resource="#Areas"/>
</owl:Class>
<owl:Class rdf:ID="FormacionBasica">
  <rdfs:subClassOf rdf:resource="#Materias"/>
</owl:Class>
<owl:Class rdf:ID="Crear">
  <rdfs:subClassOf rdf:resource="#NivelHabilidadComplejidad"/>
</owl:Class>
<owl:Class rdf:ID="Comprender">
  <rdfs:subClassOf rdf:resource="#NivelHabilidadComplejidad"/>
</owl:Class>
<owl:Class rdf:ID="LibreConfiguracion">
  <rdfs:subClassOf rdf:resource="#Materias"/>
</owl:Class>
<owl:Class rdf:ID="UTPL"/>
<owl:Class rdf:ID="SistemasInformaticosYComputacion">
  <rdfs:subClassOf rdf:resource="#TitulaciÃ³n"/>
</owl:Class>
<owl:Class rdf:ID="Recordar">
  <rdfs:subClassOf rdf:resource="#NivelHabilidadComplejidad"/>
</owl:Class>
<owl:Class rdf:ID="Abierta">
  <owl:disjointWith rdf:resource="#Presencial"/>
  <rdfs:subClassOf rdf:resource="#Modalidad"/>

```

```

</owl:Class>
<owl:Class rdf:ID="PlanDeEstudios"/>
<owl:Class rdf:ID="GeologiaYMinasEIngenieriaCIvil">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="DepartamentoAreaTecnica"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="IngenieriaInformatica">
  <rdfs:subClassOf rdf:resource="#TitulaciÃ³n"/>
</owl:Class>
<owl:Class rdf:ID="Evaluar">
  <rdfs:subClassOf rdf:resource="#NivelHabilidadComplejidad"/>
</owl:Class>
<owl:Class rdf:ID="GestionProductiva">
  <rdfs:subClassOf rdf:resource="#Materias"/>
</owl:Class>
<owl:Class rdf:ID="Complementarios">
  <rdfs:subClassOf rdf:resource="#Materias"/>
</owl:Class>
<owl:Class rdf:ID="Aplicar">
  <rdfs:subClassOf rdf:resource="#NivelHabilidadComplejidad"/>
</owl:Class>
<owl:Class rdf:ID="ArquitecturasYArtes">
  <rdfs:subClassOf rdf:resource="#DepartamentoAreaTecnica"/>
</owl:Class>
<owl:Class rdf:ID="CompetenciasEspecificas"/>
<owl:Class rdf:ID="Biologica">
  <rdfs:subClassOf rdf:resource="#Areas"/>
</owl:Class>
<owl:Class rdf:ID="CienciasDeLAComputaciÃ³nYElectronica">
  <rdfs:subClassOf rdf:resource="#DepartamentoAreaTecnica"/>
</owl:Class>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.4.8, Build 629)
http://protege.stanford.edu -->

```

ANEXO 5

Aplicación de verificación de competencias académicas

Consultas SPARQL obtenidas:

A continuación se listan algunas de las consultas SPARQL que se realizaron para la visualización de los datos del modelo ontológico.

- Consulta para obtener los planes:

```
SELECT ?s
```

```
WHERE { ?z ont:nombPlanEs ?s.
```

```
?z rdf:type ont:PlanDeEstudios}
```

```
order by ?s
```

- Consulta para obtener las competencias específicas de un respectivo plan de estudios.

```
SELECT ?s ?z
```

```
WHERE { ?x ont:pertenece ?o.  
?o rdf:type ont:CompetenciasEspecificas .  
?x rdf:type ont:PlanDeEstudios.  
?o ont:nombComp ?z.  
?x ont:nombPlanEs ?s}
```

```
order by ?s
```

- Consulta para obtener los verbos.

```
SELECT distinct (fn:upper-case(?b) as ?verbo)
```

```
WHERE { ?a ont:nombreSinonimos ?b.  
?a rdf:type ?d.  
?f rdf:type ?e.  
?f ont:nivelHabilidad ?c.  
?d rdfs:subClassOf ?e.  
?e rdfs:subClassOf ont:NivelHabilidadComplejidad} union  
{?a ont:nombreVerbos ?b.  
?a rdf:type ?d.  
?f rdf:type ?e.  
?f ont:nivelHabilidad ?c.  
?d rdfs:subClassOf ?e. ?e rdfs:subClassOf ont:NivelHabilidadComplejidad} union  
{?a ont:nombreSinonimos ?b.  
?a rdf:type ?e.  
?f rdf:type ?e.  
?f ont:nivelHabilidad ?c.  
?e rdfs:subClassOf ont:NivelHabilidadComplejidad} union  
{?a ont:nombreNivel ?b.  
?a ont:nivelHabilidad ?c.  
?a rdf:type ?e.  
?e rdfs:subClassOf ont:NivelHabilidadComplejidad}  
filter(?e=ont:"Analizar")  
}order by ?b
```

Código java de la aplicación de verificación de Compendias

Servlet

```
package servlet;
```

```

import consultas.ConsultasVerbos;
import consultas.ConsultasPlanes;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Corina Ortiz
 *
 */
@WebServlet(name = "servlet", urlPatterns = {"/servlet"})
public class servlet extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        request.setCharacterEncoding("UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */

            ServletContext context = request.getServletContext();
            ConsultasPlanes c = new ConsultasPlanes(context);

            String planes = request.getParameter("materia");
            String competencia = request.getParameter("competencia");
            String componente = request.getParameter("componente");
            String comparar = request.getParameter("comparar");

            planes = planes != null && !planes.equals("Seleccione uno.....") ? planes : "";
            competencia = competencia != null && !competencia.equals("Seleccione uno.....") ?
competencia : "";
            componente = componente != null ? componente : "";

            String tablaCoincidencias = "";
            String tablaVerbosRecomendados = "";

```

```

String subSobreDimension = "";
if (comparar != null && comparar.equals("si")) {
    ConsultasVerbos d = new ConsultasVerbos(c.getCabecera(), c.getModelo());
    tablaCoincidencias = d.buscarCoincidencias(competencia, componente);
    subSobreDimension = tablaCoincidencias.length() > 0 ? "" :
d.buscarSubSobreDimension(competencia, componente);
    tablaVerbosRecomendados = tablaCoincidencias.length() > 0 ? "" :
d.buscarRecomendaciones(competencia);
}
out.println("<!DOCTYPE html>\n"
+ "<html>\n"
+ " <head>\n"
+ " <title>Modelador</title>\n"
+ " <meta charset=\"UTF-8\">\n"
+ " <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\">\n"
+ " <link rel=\"stylesheet\" type=\"text/css\" href=\"css/inicio.css\"/>\n"
+ " <link rel=\"stylesheet\" type=\"text/css\" href=\"css/tabla.css\"/>\n"
+ " </head>\n"
+ " <body>\n"
+ " <header>\n"
+ " <img src=\"img/logo.png\">\n"
+ " <div class=\"titulo\">\n"
+ " <h1>MODELO DE COMPETENCIAS ACADÉMICAS</h1>\n"
+ " <h2>SISTEMAS INFORMÁTICOS Y COMPUTACIÓN</h2>\n"
+ " </div> \n"
+ " </header>\n"
+ " <article>\n"
+ " \n"
+ " </article>\n"
+ " <script type=\"text/javascript\">\n"
+ " function materia(valor) {\n"
+ " document.getElementById('materia').value = valor;\n"
+ " document.getElementById('competencia').value = \"\";\n"
+ " document.getElementById('aceptar').submit();\n"
+ " }\n"
+ " function competencia(valor) {\n"
+ " document.getElementById('materia').value = \"\" + planes + \"\";\n"
+ " document.getElementById('competencia').value = valor;\n"
+ " document.getElementById('aceptar').submit();\n"
+ " }\n"
+ " function comparar() {\n"
+ " document.getElementById('materia').value = \"\" + planes + \"\";\n"
+ " document.getElementById('competencia').value = \"\" + competencia + \"\";\n"
+ " document.getElementById('comparar').value = 'si';\n"
+ " document.getElementById('aceptar').submit();\n"
+ " }\n"
+ " </script>\n"
+ " <div class=\"seleccion\">\n"
+ " <h3>Componente Académico</h3>\n"
+ " <select name=\"materia\" onchange=\"materia(this.value)\">\n"
+ " <option>Seleccione uno.....</option>\n"

```

```

+ (!planes.equals(""))
? c.getList(c.listaPlanes()).replace("<option>" + planes, "<option selected>" + planes)
+ "      </select>\n"
+ "      <h3>Competencia Específica</h3>\n"
+ "      <select name=\"competencia\" onchange=\"competencia(this.value)\">\n"
+ "        <option>Seleccione uno.....</option>\n"
+ (!competencia.equals(""))
? c.getList(c.listaCompetenciasMateria(planes))
.replace("<option>" + competencia.trim(), "<option selected>" + competencia.trim())
+ "      </select>\n"
+ "      <h3>Competencia del componente</h3>\n"
+ "      <form id=\"aceptar\" action=\"servlet\" method=\"POST\">\n"
+ "        <input id=\"materia\" name=\"materia\" style=\"display: none;\"/>\n"
+ "        <input id=\"competencia\" name=\"competencia\" style=\"display:
none;\"/>\n"
+ "        <input id=\"componente\" name=\"componente\" type=\"text\" value=\"\" +
componente + \"\"/>\n"
+ "        <input id=\"comparar\" name=\"comparar\" style=\"display: none;\"/>\n"
+ "      </form>\n"
+ "      <button onclick=\"comparar();\">Comparar</button>\n"
+ "    </div>\n"
+ (comparar != null && comparar.length() > 0 ? "    <div class=\"resultado\">\n"
+ "      <h6>RESULTADO:</h6>\n"
+ "      <h6>" + (tablaCoincidencias.length() > 0
? "Está acorde."
+ tablaCoincidencias
: (tablaVerbosRecomendados.length() > 0
? "No está acorde. " + (subSobreDimension.length() > 0 ? "Está " +
subSobreDimension + ". " : "")) + "Recomendamos usar:"
+ tablaVerbosRecomendados
: "No se encuentra en ningún nivel")) + "</h6>\n"
+ "    </div>\n" : "")
: c.getList(c.listaCompetenciasMateria(planes))
+ "      </select>\n"
+ "      <form id=\"aceptar\" action=\"servlet\" method=\"POST\">\n"
+ "        <input id=\"materia\" name=\"materia\" style=\"display: none;\"/>\n"
+ "        <input id=\"competencia\" name=\"competencia\" style=\"display:
none;\"/>\n"
+ "        <input id=\"comparar\" name=\"comparar\" style=\"display: none;\"/>\n"
+ "      </form>\n"
+ "    </div>\n")
: c.getList(c.listaPlanes())
+ "      </select>\n"
+ "      <form id=\"aceptar\" action=\"servlet\" method=\"POST\">\n"
+ "        <input id=\"materia\" name=\"materia\" style=\"display: none;\"/>\n"
+ "        <input id=\"competencia\" name=\"competencia\" style=\"display:
none;\"/>\n"
+ "        <input id=\"comparar\" name=\"comparar\" style=\"display: none;\"/>\n"
+ "      </form>\n"
+ "    </div>\n")
+ "  </body>\n"

```

```

        + "</html>");
    }
}

```

Consultas Planes

```

package consultas;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.ArrayList;
import javax.servlet.ServletContext;
import org.apache.jena.ontology.OntModel;
import org.apache.jena.ontology.OntModelSpec;
import org.apache.jena.query.Query;
import org.apache.jena.query.QueryExecution;
import org.apache.jena.query.QueryExecutionFactory;
import org.apache.jena.query.QueryFactory;
import org.apache.jena.query.QuerySolution;
import org.apache.jena.query.ResultSet;
import org.apache.jena.rdf.model.ModelFactory;

/**
 *
 * @author Corina Ortiz
 */
public class ConsultasPlanes {

    private final String cabecera = "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n"
        + "PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"
        + "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"
        + "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"
        + "PREFIX ont:
<http://www.semanticweb.org/ontologies/2014/11/ModelamientoCompetenciasAcademicas.owl#>
\n"
        + "PREFIX fn: <http://www.w3.org/2005/xpath-functions#>";
    OntModel modelo;

    public ConsultasPlanes(ServletContext context) {
        String ruta = context.getRealPath("onto") + File.separator + "archivo.owl";
        File archivo = new File(ruta);
        InputStream archivoIn = null;
        try {
            InputStream is = new FileInputStream(archivo);
            archivoIn = is;
        } catch (FileNotFoundException ex) {
        }
        modelo = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
        modelo.read(archivoIn, "RDF/XML");
    }
}

```

```

}

public String listaPlanes() {
    return cabecera + "SELECT ?s WHERE { ?z ont:nombPlanEs ?s.\n"
        + "?z rdf:type ont:PlanDeEstudios} order by ?s";
}

public String listaCompetenciasMateria(String planes) {
    return cabecera + "SELECT ?s ?z WHERE { \n"
        + "?x ont:poseeCE ?o.\n"
        + "?o rdf:type ont:CompetenciasEspecificas.\n"
        + "?x rdf:type ont:PlanDeEstudios.\n"
        + "?o ont:nombComp ?s.\n"
        + "filter(?z = \"\" + planes + "\"^^xsd:string).\n"
        + "?x ont:nombPlanEs ?z} order by ?s";
}

public ArrayList<String> hacerConsultaUnaColumna(String consultaString, OntModel modelo,
boolean sinonimo) {
    Query consulta = QueryFactory.create(consultaString);
    QueryExecution qexec = QueryExecutionFactory.create(consulta, modelo);
    ArrayList<String> lista = new ArrayList<>();
    try {
        ResultSet resultado = qexec.execSelect();
        while (resultado.hasNext()) {
            QuerySolution salida = resultado.next();
            lista.add(salida.get("?s").toString().trim().replaceAll(" ", " ").replaceAll(", ", " "));
        }
    } catch (Exception ex) {
    }
    qexec.close();
    return lista;
}

public String getLista(String consulta) {
    ArrayList<String> r = hacerConsultaUnaColumna(consulta, modelo, false);
    String devolver = "";
    for (int i = 0; i < r.size(); i++) {
        String resultado = r.get(i).split("^^")[0].replace("\\", "").trim().replaceAll(" ", " ").replaceAll(", ", " ");
        devolver += !resultado.equals("") ? "      <option>" + resultado + "</option>\n" : "";
    }
    return devolver;
}

public OntModel getModelo() {
    return modelo;
}

public String getCabecera() {
    return cabecera;
}

```

```
}
```

```
}
```

Consultas Verbos

```
public class ConsultasVerbos {

    private final String cabecera;
    private final OntModel modelo;
    private final String consultaCompleta = "{?a ont:nombreSinonimos ?b. ?a rdf:type ?d. ?f rdf:type
?e. ?f ont:nivelHabilidad ?c. ?d rdfs:subClassOf ?e. ?e rdfs:subClassOf
ont:NivelHabilidadComplejidad} union\n"
        + "{?a ont:nombreVerbos ?b. ?a rdf:type ?d. ?f rdf:type ?e. ?f ont:nivelHabilidad ?c. ?d
rdfs:subClassOf ?e. ?e rdfs:subClassOf ont:NivelHabilidadComplejidad} union\n"
        + "{?a ont:nombreSinonimos ?b. ?a rdf:type ?e. ?f rdf:type ?e. ?f ont:nivelHabilidad ?c. ?e
rdfs:subClassOf ont:NivelHabilidadComplejidad} union\n"
        + "{?a ont:nombreNivel ?b. ?a ont:nivelHabilidad ?c. ?a rdf:type ?e. ?e rdfs:subClassOf
ont:NivelHabilidadComplejidad}";

    public ConsultasVerbos(String cabecera, OntModel modelo) {
        this.modelo = modelo;
        this.cabecera = cabecera;
    }

    public String buscarCoincidencias(String competencia, String componente) {
        competencia = competencia.toUpperCase();
        componente = componente.toUpperCase();
        List<String> palabrasCompetencia = Arrays.asList(competencia.replace(","," ").replace(" ", "
").split(" "));
        List<String> palabrasComponente = Arrays.asList(componente.replace(","," ").replace(" ", "
").split(" "));
        List<String> verbosOntologia = buscarVerbos(palabrasCompetencia);
        List<String> verbosBuscados = buscarVerbos(palabrasComponente);
        String consulta = cabecera + "SELECT distinct ?verbo ?verbo2 ?nhc WHERE{\n{\n"
            + consultaInterna(verbosOntologia) + "}}\n.\n{\n"
            + consultaInterna(verbosBuscados)
            .replace("?verbo", "?verbo2")
            .replace("?nivel", "?nivel2")
            .replace("?nhc", "?nhc2") + "}}\n.\n"
            + "FILTER(?nhc=?nhc2)\n"
            + "}}";
        return hacerTablaVerbos(consulta);
    }

    public String buscarRecomendaciones(String competencia) {
        competencia = competencia.toUpperCase();
        List<String> palabrasCompetencia = Arrays.asList(competencia.replace(","," ").replace(" ", "
").split(" "));
        List<String> verbosOntologia = buscarVerbos(palabrasCompetencia);
        String consulta = cabecera + consultaInterna(verbosOntologia);
```

```

String nivel = hacerLista(consulta, "?nhc").get(0);
consulta = cabecera + "SELECT distinct (fn:upper-case(?b) as ?verbo)"
    + " WHERE {" + consultaCompleta + ".\n"
    + "filter(?e=ont:" + nivel + ")\n"
    + "}order by ?b";
return hacerTablaRecomendados(hacerListaSimple(consulta, "?verbo"), nivel);
}

public String buscarSubSobreDimension(String competencia, String componente) {
    competencia = competencia.toUpperCase();
    componente = componente.toUpperCase();
    List<String> palabrasCompetencia = Arrays.asList(competencia.replace(", ", " ").replace(" ", " ").split(" "));
    List<String> palabrasComponente = Arrays.asList(componente.replace(", ", " ").replace(" ", " ").split(" "));
    List<String> verbosOntologia = buscarVerbos(palabrasCompetencia);
    List<String> verbosBuscados = buscarVerbos(palabrasComponente);
    int nivelOntologia = Integer.valueOf(hacerListaSimple(cabecera +
        consultaInterna(verbosOntologia, "?nivel").get(0).substring(0, 1));
    int nivelBuscado = 0;
    try {
        nivelBuscado = verbosBuscados.size() > 0 ? Integer.valueOf(hacerListaSimple(cabecera +
            consultaInterna(verbosBuscados, "?nivel").get(0).substring(0, 1)) : 0;
    } catch (Exception ex) {
    }
    if (nivelBuscado > 0) {
        if (nivelBuscado < nivelOntologia) {
            return "Sobre Dimensionado";
        } else if (nivelBuscado > nivelOntologia) {
            return "Sub Dimensionado";
        } else {
            return "Igual Dimensión";
        }
    }
    return "No se encuentra en ningún nivel";
}

private String consultaInterna(List<String> verbos) {
    String subConsulta = "SELECT distinct (fn:upper-case(?b) as ?verbo)(?c as ?nivel) (?e as ?nhc)
WHERE {\n"
        + consultaCompleta + "\n.\n"
        + "FILTER(";
    for (String verbo : verbos) {
        subConsulta += "fn:upper-case(?b)=\"\" + verbo + "\"^^xsd:string ||";
    }
    subConsulta = subConsulta.substring(0, subConsulta.length() - 2);
    subConsulta += ")\n"
        + "}order by ?c ?e ?d ?b ?a";
    return subConsulta;
}

```

```

private List<String> hacerLista(String consultaSparql, String columna) {
    List<String> lista = new ArrayList<>();
    try {
        Query consulta = QueryFactory.create(consultaSparql);
        QueryExecution qexec = QueryExecutionFactory.create(consulta, modelo);
        ResultSet resultado = qexec.execSelect();
        while (resultado.hasNext()) {
            QuerySolution salida = resultado.next();
            lista.add(obtenerTexto(salida, columna));
        }
        qexec.close();
    } catch (Exception ex) {
    }
    return lista;
}

```

```

private List<String> hacerListaSimple(String consultaSparql, String columna) {
    List<String> lista = new ArrayList<>();
    try {
        Query consulta = QueryFactory.create(consultaSparql);
        QueryExecution qexec = QueryExecutionFactory.create(consulta, modelo);
        ResultSet resultado = qexec.execSelect();
        while (resultado.hasNext()) {
            QuerySolution salida = resultado.next();
            lista.add(obtenerTextoSimple(salida, columna));
        }
        qexec.close();
    } catch (Exception ex) {
    }
    return lista;
}

```

```

private List<String> buscarVerbos(List<String> palabras) {
    List<String> verbos = new ArrayList<>();
    for (String palabra : palabras) {
        String parteFinal = palabra.length() > 2
            ? palabra.substring(palabra.length() - 2,
                palabra.length())
            : null;
        if (parteFinal != null && (parteFinal.equals("ar")
            || parteFinal.equals("er") || parteFinal.equals("ir")
            || parteFinal.equals("AR")
            || parteFinal.equals("ER") || parteFinal.equals("IR"))) {
            verbos.add(palabra);
        }
    }
    return verbos;
}

```

```

private String obtenerTexto(QuerySolution salida, String columna) {
    try {

```

```

String obtenido = salida.get(columna.trim()).toString();
obtenido = obtenido.split("#")[1];
obtenido = obtenido.substring(0, 1).toUpperCase()
    + obtenido.substring(1).toLowerCase();
return obtenido;
} catch (Exception ex) {
return ex.getMessage();
}
}

private String obtenerTextoSimple(QuerySolution salida, String columna) {
try {
String obtenido = salida.get(columna.trim()).toString();
obtenido = obtenido.substring(0, 1).toUpperCase()
    + obtenido.substring(1).toLowerCase();
return obtenido;
} catch (Exception ex) {
return ex.getMessage();
}
}

private String hacerTablaRecomendados(List<String> listado, String nivel) {
String devolver = "<table>\n"
    + "    <thead>\n"
    + "        <tr>\n"
    + "            <th>Recomendados: (" + nivel + ")</th>\n"
    + "        </tr>\n"
    + "    </thead>\n"
    + "    <tbody>";
for (String palabra : listado) {
devolver += "        <tr>\n"
    + "            <td>" + palabra + "</td>\n"
    + "        </tr>\n";
}
devolver += "    </tbody>\n"
    + "</table>\n";
return devolver;
}

private String hacerTablaVerbos(String consultaSparql) {
try {
String devolver = "";
Query consulta = QueryFactory.create(consultaSparql);
QueryExecution qexec = QueryExecutionFactory.create(consulta, modelo);
ResultSet resultado = qexec.execSelect();
while (resultado.hasNext()) {
QuerySolution salida = resultado.next();
devolver += "\n"
    + "        <tr>\n"
    + "            <td>" + obtenerTextoSimple(salida, "?verbo") + "</td>\n"
    + "            <td>" + obtenerTextoSimple(salida, "?verbo2") + "</td>\n"

```

```

        + "          <th>" + obtenerTexto(salida, "?nhc") + "</th>\n"
        + "          </tr>\n";
    }
    devolver = devolver.length() > 0 ? "<table>\n"
    + "      <thead>\n"
    + "      <tr>\n"
    + "          <td>Competencia Específica</td>\n"
    + "          <td>Competencia Componente</td>\n"
    + "          <th>Nivel Habilidad</th>\n"
    + "      </tr>\n"
    + "      </thead>\n"
    + "      <tbody>"
    + devolver
    + "      </tbody>\n"
    + "      </table>\n" : "";
    return devolver;
} catch (Exception ex) {
    return "";
}
}
}

```