

Universidad de las Ciencias Informáticas
Facultad 4



Universidad de las Ciencias
Informáticas

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

***Título: Ontología de apoyo al proceso de evaluación de productos
de software en la UCI.***

Autor: Leosmay Carrión Estradet

Tutor(es): Ing. Aliuska Castañeda Martínez

Msc. Yoan Antonio López Rodríguez

Dra. Yamilis Fernández Pérez

Declaración de autoría

Declaro ser el único autor del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas a ser uso del mismo en su beneficio.

Para que así conste, firmo la presente declaración jurada de autoría a los _____ días del mes de _____ del año _____.

Autor: Leosmay Carrión Estradet

Tutor: Dra. Yamilis Fernández Pérez

Tutor: Ing. Aliuska Castañeda Martínez

Tutor: MSc.Yoan Antonio López Rodríguez

Agradecimientos

A **Dios** por guiar cada uno de mis pasos y darme la fuerza para cumplir todas las metas que me propuse en la vida.

A mi **abuela** Lucía García Dominica Pérez por ser mi inspiración y mejor amiga.

A mi **madre** Magalis Estradet García por brindarme siempre su amor incondicional, por ser un ejemplo para mí, por su fuerza y dedicación.

A mi **tío** Fermín; de estar vivo se sentiría muy orgullosa de su sobrino, el profesional que siempre predijo.

A mis **hermanos** mayores Leonardo Carrión, Iván Estradet y Hiosvany Salgado por todo el apoyo que me han brindado para lograr cada una de mis metas.

A mis **padres** Leonardo Carrión Sagó y Domingo Duvergel Iznaga.

A mi **compañero de tesis** Carlos Parker Leiva, por tener paciencia con mi novia, por su compromiso, dedicación con el trabajo, darme la fuerza para ir todos los días al aula y participar en todas las actividades.

A mi **suegra** Olga Baró García por ser como una madre para mí.

A mis **tutores** Aliuska Castañeda, Yamilis Fernández y Yoan Antonio López porque sin su ayuda este trabajo no hubiese tenido un final, mucho menos feliz.

A todos mis **amigos**, en especial a Pinar, Oscar Luis, Adrián, Manuel, Hermes, Sergio, Caridad, Katy Más, Naylin, Roberto, Ever, Leote, Joel, Oridalmis, Dayani, Adriana, Chino, Alex y Eduardo Roque.

A mis **profesoras** Karenia y Yasirys por su amistad y dedicación.

Por último, pero no menos importante a mi amiga, abogada, madre, hermana, cotutora, profesora y el amor de mi vida y madre de mis futuros hijos: Nataly y Christian, **Lien Morell Baró**.

Dedicatoria

Dedico este trabajo a mis padres, a quienes jamás encontraré la forma de agradecer el amor, aliento, comprensión y apoyo brindado en los momentos buenos y malos de mi vida, este logro es dedicado a ellos y quiero que sepan que mis esfuerzos son inspirados en cada uno de ustedes.

Resumen

Con el paso de los años el uso de software ha aumentado en todos los sectores de la vida humana en la misma medida que se torna más difícil su implementación y las funciones que realiza. Es necesario, por tanto, conseguir sistemas cada vez más precisos y seguros. De ahí que tenga gran importancia y relevancia la evaluación de su calidad. El proceso de evaluación de productos de software se torna complejo y costoso, dada la cantidad de personas y actividades involucradas, así como la variedad de criterios a considerar para evaluar los productos, lo que conlleva a la búsqueda de herramientas y técnicas con el objetivo de reducir la pérdida o desaprovechamiento del conocimiento. El desarrollo de ontologías es una de estas técnicas empleadas para la gestión del conocimiento ya que las mismas permiten representar el conocimiento, almacenar la información, poder organizarla y recuperarla.

Con el fin de resolver los problemas existentes que presenta este proceso, se desarrolló una ontología de apoyo para el proceso de evaluación de productos de software en la Dirección de la Calidad de Software de la Universidad de las Ciencias Informáticas, la cual permite organizar y recuperar el conocimiento generado en dicho proceso.

Para validar la propuesta se utilizó un esquema para evaluar ontologías únicas para un dominio de conocimiento que consta de cuatro fases y las preguntas de competencias. Demostrando que la ontología satisface los requisitos para los cuales fue creada.

Palabras clave: calidad de software, evaluación del producto de software, gestión del conocimiento, ontología.

Índice general

Introducción	2
Capítulo 1 Fundamentación teórica	6
Introducción.....	6
1.1 Calidad y evaluación del producto de software	6
1.2 Gestión del conocimiento.....	8
1.2.1 Sistemas de Organización de Conocimientos.....	10
1.3 Ontologías	11
1.3.1 Tipos de ontologías.....	12
1.3.2 Componentes de una ontología	13
1.4 Metodologías para el desarrollo de ontologías.....	14
1.5 Ontologías existentes en la calidad de software.....	17
1.6 Herramientas y tecnologías para la construcción de ontologías.....	17
1.6.1 Herramientas para el desarrollo de ontologías	17
1.6.2 Razonadores de ontologías.....	18
1.6.3 Herramientas CASE	20
1.7 Conclusiones parciales	21
Capítulo 2: Diseño de la propuesta ontológica	22
Introducción.....	22
2.1 Descripción de la propuesta de solución.....	22
2.2 Aplicación de la metodología Methontology	24
2.2.1 Especificación	24
2.2.2 Conceptualización	25
2.2.3 Formalización	39
2.2.4 Implementación	45
2.2.5 Mantenimiento.....	46
2.3 Conclusiones del capítulo	47
Capítulo 3 Validación y prueba de la ontología	48
Introducción.....	48

3.1 Validación de la ontología	48
3.1.1 Fase 1. Uso correcto del lenguaje	49
3.1.2 Fase 2. Exactitud de la estructura taxonómica	49
3.1.3 Fase 3. Validez del vocabulario	50
3.1.4 Fase 4. Adecuación a requerimientos.....	51
3.2 Conclusiones del capítulo	53
Conclusiones generales.....	54
Recomendaciones	55
Referencias bibliográficas	56
Anexos.....	60
Anexo 1. Taxonomía de conceptos de la ontología Proceso de Evaluación de Software	60
Anexo 2. Casos de prueba de la ontología Proceso de Evaluación de Software	63
Anexo 3. Diseño de la entrevista	71
Anexo 4. Expertos entrevistados	72

Índice de tablas

Tabla 1: Documento de Especificación de Requerimientos.....24

Tabla 2: Glosario de términos.25

Tabla 3: Diccionario de conceptos de la ontología Proceso de Evaluación de Software.31

Tabla 4: Relación binaria de la ontología Proceso de Evaluación de Software.34

Tabla 5: Descripción de los atributos de instancias.....36

Tabla 6: Ejemplo de una regla en la ontología Proceso de Evaluación de Software.38

Tabla 7: Declaración de las clases.....45

Tabla 8: Declaración de una subclase de otra clase.45

Tabla 9: Declaración de las relaciones.....46

Tabla 10: Creación de los individuos de las clases.46

Tabla 11: Consulta SPARQL.....46

Tabla 12: Ejemplo de instancia de la clase Evaluación.51

Tabla 13: Ejemplo de instancia de la clase Especialista.....63

Tabla 14: Tabla 14: Ejemplo de instancia de la clase Especialista.....64

Tabla 15: Ejemplo de instancia de la clase Actividad.65

Tabla 16: Ejemplo de instancia de la clase Especialista.....65

Tabla 17: Ejemplo de instancia de la clase Completitud funcional.66

Tabla 18: Ejemplo de instancia de la clase Nivel de evaluación.....67

Tabla 19: Ejemplo de instancia de la clase Actividad.68

Tabla 20: Ejemplo de instancia de la clase Proyecto.69

Índice de figuras

Figura. 1: Proceso de evaluación de la calidad del producto se software. Fuente: (Pérez, y otros, 2016)....7	7
Figura. 2: Modelo conceptual del Proceso de Evaluación de Software. Fuente: Elaboración propia.23	23
Figura. 3: Taxonomía de conceptos de la ontología Proceso de Evaluación de Software. Fuente: Elaboración propia.30	30
Figura. 4: Diagrama de relaciones binarias de la ontología Proceso de Evaluación de Software. Fuente:Elaboración propia.30	30
Figura. 5: Definición de las clases creadas en Protégé. Fuente: Elaboración propia.....40	40
Figura. 6: Propiedades de las clases creadas en Protégé. Fuente: Elaboración propia.41	41
Figura. 7: Creación del tipo de datos de las propiedades en Protégé. Fuente: Elaboración propia.41	41
Figura. 8: Creación de las reglas. Fuente: Elaboración propia.42	42
Figura. 9: Creación de las instancias con Protégé. Fuente: Elaboración propia.43	43
Figura 10: Pestaña OWL Viz del Protégé. Fuente: Elaboración propia.44	44
Figura 11 : Aplicación del razonador. Fuente: Elaboración propia.....52	52
Figura 12: Resultado de la evaluación1. Fuente: Elaboración propia.....52	52
Figura 13: Taxonomía 2. Fuente: Elaboración propia.....60	60
Figura 14: Taxonomía 3. Fuente: Elaboración propia.....61	61
Figura 15: Taxonomía 4. Fuente: Elaboración propia.....61	61
Figura 16: Taxonomía 5. Fuente: Elaboración propia.....62	62
Figura 17: Caso de prueba 2. Fuente: Elaboración propia.63	63
Figura 18: Caso de prueba 3. Fuente: Elaboración propia.64	64
Figura 19: Caso de prueba 4. Fuente: Elaboración propia.65	65
Figura 20: Caso de prueba 5. Fuente: Elaboración propia.66	66
Figura 21: Caso de prueba 6. Fuente: Elaboración propia.67	67
Figura 22: Caso de prueba 7. Fuente: Elaboración propia.68	68
Figura 23: Caso de prueba 8. Fuente: Elaboración propia.69	69
Figura 24: Caso de prueba 9. Fuente: Elaboración propia.70	70

Introducción

En los últimos años, el software se ha convertido en un objetivo estratégico de cualquier organización interesada en optimizar sus procesos fundamentales y de apoyo. El uso de estos productos logra aumentar la productividad y las posibilidades de negocio de dichas organizaciones. Esta situación crea un ambiente de competencia y de especialización del software entre las empresas que lo producen y comercializan, lo que ha provocado especial atención la calidad del mismo, siendo factor clave del control la evaluación de esta.

La calidad se ha convertido en uno de los elementos diferenciadores en el ámbito mundial entre las compañías desarrolladoras de productos de software. La búsqueda de la calidad de los sistemas ha propiciado la creación de modelos, marcos de trabajo y metodologías para evaluar y asegurar su calidad (Abascal, 2017). La Universidad de las Ciencias Informáticas (UCI) como entidad desarrolladora de software no está exenta y se encuentra definiendo el proceso de evaluación de productos de software.

Este proceso se torna complejo y costoso, dada la cantidad de personas y actividades involucradas, así como la variedad de criterios a considerar para evaluar los productos. Lo cual conlleva a la consulta de varios modelos y estándares, como son los de Boehm, McCall, FURPS, y las normas NC ISO/IEC 25000, así como de varios subprocesos ya definidos en la UCI y que se incluyen en la evaluación (Pérez, y otros, 2017). Esta amalgama de procesos, estándares y modelos de calidad dificulta su institucionalización por la diversidad de términos y conceptos utilizados, donde coexisten inconsistencias y conflictos entre ellos. Además, hay principios y métodos que aún están siendo definidos y consolidados. Igualmente, si analizamos los estándares internacionales desarrollados por las principales organizaciones e instituciones de estandarización, como IEEE e ISO, la problemática se mantiene. Por ejemplo, la NC ISO/IEC 25000, contiene una visión general sobre la calidad de productos de software, pero hay aspectos, como la medición, que se complementan con otros estándares y ofrecen vistas parciales de este dominio. Hay otro ámbito, como el de las pruebas de software, que escasamente son tratados.

Todo lo anterior, conlleva a la gestión del conocimiento sobre el proceso de evaluación de productos de software en la Dirección de la Calidad de Software de la UCI, para ello se hace necesario organizar y recuperar la información existente sobre dicho proceso. Las ontologías constituyen una de las herramientas más utilizadas para organizar y recuperar el conocimiento. Se analizaron propuestas

ontológicas relacionadas con el proceso de evaluación de productos de software. La principal dificultad para reutilizar estas propuestas radica, en que las mismas tratan un dominio que no cubren todos los términos y conceptos de dicho proceso. Además, no se puede reutilizar las propuestas ontológicas existentes, debido a que los autores no proporcionan el código generado.

Por lo demás, en la industria cubana de software al entrevistar a especialistas de empresas destacadas como Calisoft, XETID, SOFTEL y especialistas del Subcomité 7 de la ISO en Cuba, se llegó a la conclusión que hay una insuficiente organización de los términos utilizados, no existe un vocabulario común. Por lo que se hace necesario proporcionar un vocabulario común para resolver el problema de integridad e inconsistencia, identificado en los diversos estándares y en la información involucrada en el proceso, con el objetivo de reducir la pérdida o desaprovechamiento del conocimiento.

Atendiendo la situación planteada y enfocándose en su solución, se propone como **problema de investigación**: ¿cómo contribuir a la organización y recuperación del conocimiento generado en el proceso de evaluación de productos de software definido en la UCI?

Definiendo como **objeto de estudio**: la gestión del conocimiento basado en ontologías enmarcando como **campo de acción** las ontologías para gestionar el conocimiento en el proceso de evaluación de productos de software definido en la UCI.

Para dar solución al problema de investigación se propone como **objetivo general**: desarrollar una ontología que permita organizar y recuperar el conocimiento generado en el proceso de evaluación de productos de software definido en la UCI. Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos**:

1. Desarrollar el marco teórico de la investigación para un mayor entendimiento de las ontologías y del proceso de evaluación de productos de software.
2. Caracterizar el proceso de evaluación de productos de software para determinar los principales conceptos y sus relaciones.
3. Definir una metodología para el desarrollo de ontologías.
4. Definir las herramientas y tecnologías a utilizar para el desarrollo de la ontología que se propone.
5. Diseñar la ontología que represente el conocimiento generado en el proceso de evaluación de

productos de software definido en la Dirección de la Calidad UCI.

6. Validar la propuesta ontológica en función de la organización y recuperación del conocimiento generado en el proceso de evaluación de productos de software.

Para asistir los objetivos específicos se emplearon los **métodos científicos** que se exponen a continuación:

Métodos teóricos:

Entre los métodos teóricos tenemos:

1. **Histórico - lógico:** permitió fundamentar los conceptos relacionados con la gestión del conocimiento, la calidad de software, el proceso de evaluación del producto de software y la ingeniería de ontológica, así como sus técnicas, metodologías y herramientas.
2. **Analítico - sintético:** posibilitó analizar por partes el objeto de estudio mediante la determinación de sus componentes elementales y la relación entre ellos, desde los Sistemas de Organización de Conocimientos, la recuperación de la información y las ontologías.

Métodos empíricos:

1. **Entrevista:** facilitó la obtención de información a través de un grupo de expertos de empresas como XETID, SOFTEL, Calisoft y el subcomité 7 de la ISO en Cuba para saber cómo se ejecuta el proceso de evaluación de productos de software.
2. **Análisis documental:** facilitó el amplio estudio del dominio de la bibliografía consultada, seleccionando información más certera sobre el tema.

El **resultado esperado** en esta investigación es: una ontología que apoye el proceso de evaluación de productos de software en la UCI, que permitirá la organización y recuperación del conocimiento generado en la Dirección de Calidad de Software.

El presente documento consta de tres capítulos, la descripción de los mismos se presenta a continuación:

- **Capítulo 1: Fundamentación teórica:** cuenta con el respaldo teórico de los temas tratados en el trabajo de investigación, necesarios para el correcto entendimiento de la solución propuesta. Se analizaron los principales conceptos relacionados con el tema de investigación, así como las características fundamentales de las herramientas y tecnologías para desarrollar las ontologías en función de la gestión del conocimiento. Además, se mostraron algunas formas de representar la ontología y se presentó el ambiente de desarrollo necesario para el análisis y diseño de la misma.
- **Capítulo 2: Diseño de la propuesta ontológica:** se describió el desarrollo de la propuesta con cada uno de sus aspectos y detalles a través de la metodología seleccionada para la creación de la ontología.
- **Capítulo 3: Validación y prueba de la ontología:** se describió la fase de validación del diseño de la propuesta ontológica utilizando un esquema para evaluar ontologías únicas para un dominio de conocimiento. Por último, se presentaron las conclusiones y recomendaciones derivadas de la investigación, una relación de las referencias bibliográficas y los anexos para facilitar la comprensión de la investigación.

Capítulo 1 Fundamentación teórica

Introducción

En el presente capítulo se realizó un estudio del arte relacionado con el proceso de evaluación del producto de software, enfocándose en sus principales conceptos y definiciones. Además, se abordaron los conceptos fundamentales relacionados con las ontologías, la calidad y la gestión del conocimiento. En este capítulo también se realizó un estudio de algunas tecnologías y herramientas para la creación de ontologías, con el objetivo de seleccionar las más indicadas para el desarrollo de una solución con las características deseadas.

1.1 Calidad y evaluación del producto de software

En el diccionario de la Real Academia Española, el término calidad se define como: propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor (Española, 2017). Analizando ésta definición, la calidad dependerá de la opinión de la persona que evalúe la misma, provocando éstos la subjetividad de la definición de calidad. De ahí, surge el término calidad de software.

La calidad de software se refiere a: grado en que el producto de software cumple las necesidades declaradas e implícitas cuando se utiliza en condiciones especificadas (Norma-Cubana-25040, 2016). Esta se divide en la calidad del producto obtenido y la calidad del proceso de desarrollo. Las mismas son dependientes ya que para lograr calidad en un producto debe haber calidad en su proceso de desarrollo. Las metas que se establezcan para la calidad del producto van a determinar los objetivos del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de la calidad del mencionado proceso. Los requerimientos de calidad más significativos del proceso de software son (Tello, 2016):

- Que produzca los resultados esperados.
- Que los resultados estén basados en una correcta definición.
- Que los resultados sean mejorados en función de los objetivos de negocio.

Dado esta definición surgen varios términos que describen los procesos y actividades, entre ellos se encuentra la evaluación del producto de software.

La evaluación del producto de software es la operación técnica que consiste en realizar una evaluación de una o más características del producto de acuerdo con un procedimiento específico. Este proceso se realiza buscando defectos que provoquen la mala operación del sistema (Norma-Cubana-25040, 2016). El mismo tiene como entrada el producto o productos de software, que se conduce por un flujo de actividades donde la calidad del software se valora a partir de un conjunto de criterios o características que se representan en un modelo de calidad. Por último se obtiene como salida el índice de calidad del producto (Pérez, y otros, 2016) (Ver figura 1).

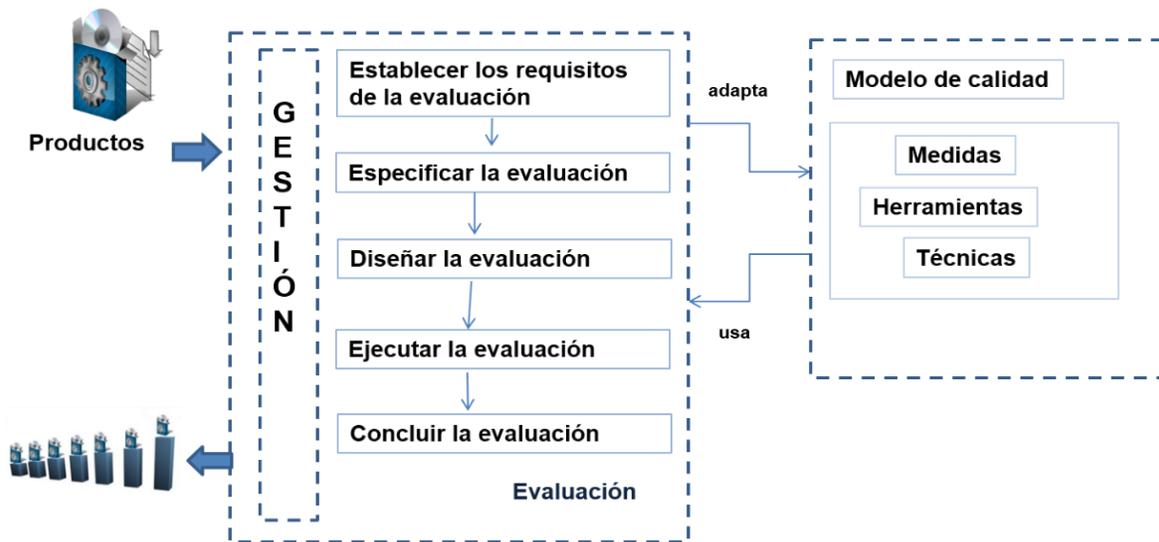


Figura. 1: Proceso de evaluación de la calidad del producto de software. Fuente: (Pérez, y otros, 2016)

El proceso de evaluación de productos de software utilizado en la Dirección de la Calidad de Software cumple con lo descrito en la norma NC ISO/IEC 25040 (Norma-Cubana-25040, 2016), es genérico y se utiliza para evaluar las características de calidad definidas en la norma NC ISO/IEC 25010:2016. Consta de cinco actividades fundamentales e intervienen cuatro especialistas: el especialista coordinador de la evaluación, el especialista de tipos de pruebas de sistema específicos, el especialista del equipo de desarrollo y el probador. A continuación, se describen cada una de las actividades del proceso de evaluación de productos de software:

1. Establecer los requisitos de evaluación.

Esta actividad se basa en establecer el propósito de la evaluación, luego obtener los requisitos de la calidad del producto de software, identificar las partes del producto a incluirse en la evaluación y por último definir el rigor de la evaluación.

2. Especificar la evaluación.

En esta segunda actividad se selecciona las medidas de la calidad y se definen criterios de decisión¹ para la medida de la calidad y para la evaluación.

3. Diseñar la evaluación.

Se planean las actividades que estarán involucradas en la evaluación.

4. Ejecutar la evaluación.

En esta actividad se realizan las mediciones y se aplican criterios de decisiones para las medidas de la calidad y para la evaluación.

5. Concluir la evaluación.

Esta es la última actividad de la evaluación de software, está comprendida por las revisiones de los resultados de la evaluación, luego se crea el informe de dicha evaluación. Se revisa la evaluación de la calidad y se proporciona retroalimentación de la organización. Por último, se ejecuta la disposición de los datos de la evaluación.

Para el logro de estas actividades se hizo necesario organizar la información y recuperar todo el conocimiento que se genera en el proceso de evaluación de software. Para lograr esta tarea de manera correcta se realizó una adecuada gestión de dicho conocimiento.

1.2 Gestión del conocimiento

La gestión del conocimiento (GC) es un conjunto de procesos sistemáticos que transita desde una etapa de identificación y captación del capital intelectual; a otra etapa de tratamiento, desarrollo, compartimiento y utilización del conocimiento (Pérez, 2007).

¹ Son umbrales u objetivos numéricos utilizados para determinar la necesidad de acción o mayor investigación, o para describir el nivel de confianza en un resultado dado.

Una definición más sistemática define la gestión de conocimiento en un proceso integrador en el que convergen la gestión de la información, la tecnología y los recursos humanos y su implementación se orienta a perfeccionar los procesos de mayor impacto, mejor explotación del conocimiento en función de los procesos y su distribución en toda la organización, sobre la base del uso intensivo de las redes y las tecnologías (María Aurora Soto, 2009).

El conocimiento no siempre está disponible cuando es necesario para la organización. Para tratar este problema han surgido nuevas formas de gestionarlo. La gestión del conocimiento se ocupa de la identificación, captura, recuperación, compartimiento y evaluación del conocimiento organizacional. El objetivo es que todo el conocimiento que reside en una empresa pueda ser utilizado por quien lo necesite para actuar de manera adecuada en cada momento (Vanguardia, 2015).

Para lograr este cometido intervienen otras disciplinas como los sistemas de información y la organización del conocimiento (OC).

La interacción de la información y el conocimiento dentro los diferentes procesos de una organización, y su articulación e identificación dentro de los sistemas de gestión de información institucionales para la toma de decisiones es tema de interés, desde hace algún tiempo, por parte de diversos especialistas. Existen diferentes iniciativas en el análisis y estructuración de los sistemas de información en función de la naturaleza de la información que se gestione, de la estructura del sector del conocimiento que abarque, de los tipos de procesos que involucra, así como de las características de la organización y sus recursos humanos (Rivero Amador, 2013).

La OC es un campo amplio e interdisciplinario, mucho más extenso que la Biblioteconomía y Documentación (Hjorland, 2004). No obstante, otros autores se refieren a la OC, como una especialidad dentro de la Bibliotecología y la Ciencia de la Información, como la ciencia de estructurar y organizar sistemáticamente las unidades de conocimiento (conceptos), de acuerdo con sus propios elementos de conocimiento (características) y la aplicación de conceptos y clases de conceptos ordenados por este campo, para la asignación de los contenidos válidos de conocimiento de referentes (objetos / sujetos) de todo tipo (Dahlberg, 2012).

Recientemente se ha demostrado que la OC tiene también un papel importante en la gestión del conocimiento de organizaciones y empresas. La organización y representación del conocimiento, en esencia, son vías de importante relevancia en el proceso de identificación de los elementos abstractos y cognitivos del pensamiento

humano, acerca del mundo que lo rodea, y de los saberes que constituyen el pilar de conocimiento de su entorno o ambiente, permitiendo mapearlos por medio de diversas técnicas. El papel que juega la Organización y Representación del Conocimiento, tanto para una organización como para la sociedad en general, es muy importante debido a las ventajas que de ello se deriva (Huertas, 2013).

Esto implica la existencia de un sistema utilizado para recuperar y transmitir el conocimiento. Los Sistemas de Organización de Conocimientos (SOCs) son propuestas para la recuperación de dicha organización y representación del conocimiento en un área especializada o propósito (López-Huertas, 2008).

1.2.1 Sistemas de Organización de Conocimientos

Los SOCs son una integración de una serie de tecnologías que ayudan a crear, gestionar y visualizar modelos que muestren una perspectiva simplificada de los conceptos que forman determinado dominio y de la estructura semántica subyacente. El objetivo de estos sistemas es principalmente ayudar a la comprensión, gestión y recuperación de los conceptos contenidos en cada dominio. Dado que cualquier disciplina tiene necesidad de mostrarse de forma comprensible y de facilitar mecanismos de gestión y de recuperación conceptual (Sanchez, 2009).

Todo SOCs se presenta a la vez como estructura y como representación. En cuanto estructura, constituye un entramado de conceptos, representados por sus respectivos términos, que establecen determinado sistema de relaciones internas, en función de la proximidad o lejanía (es decir, 'distancia') conceptual recíproca y la afiliación de cada término a una familia o red de términos que le son más afines. Por esto se dice que todo sistema de organización de conocimiento es una estructura conceptual, incluso en aquellos casos en que esa estructura no se visibiliza detrás del orden alfabético, como en las listas de encabezamientos o de descriptores (Broughton, 2008).

En cuanto a representación, constituye una réplica algo distorsionada del conocimiento acumulado, visto desde una determinada concepción filosófica o epistemológica. Al definir un léxico, el sistema de organización del conocimiento desintegra el discurso de especialidad, lo desmenuza en sus unidades lingüísticas más significativas, aquellos términos que son pertinentes según los criterios de inclusión/exclusión que se hayan establecido, y establece formas fijas y normalizadas de representación de conceptos (Barité, 2014).

Las formas más simples de un SOCs son, después de todo, las tablas de contenido y los índices de los libros de texto. El conocimiento se halla en el texto; estos sistemas son una herramienta complementaria que ayuda al lector a transitar a lo largo del texto. Con el desarrollo tecnológico y el surgimiento de nuevas herramientas de apoyo, este fenómeno se ha tornado más complejo, y han comenzado a ejercer funciones más amplias, han requerido denominaciones más notables, como lenguajes de recuperación, taxonomías, categorizaciones, léxicos, tesauros, u ontologías. Son vistos hoy como esquemas que organizan, gestionan y recuperan información. (Gómez, 2008)

Por tanto los sistemas de clasificación, tesauros, taxonomías, folksonomías, mapas de tópicos o 'Topic maps' y las ontologías, deben entenderse como sistemas que organizan conceptos y sus relaciones semánticas básicamente (Barité, 2014). Son una manera de representar el conocimiento en una determinada disciplina.

1.3 Ontologías

En el campo de la filosofía, la ontología se ha considerado una rama de la filosofía que se ocupa de la naturaleza y organización de la realidad. A partir de los años 90, dicho concepto se empezó a utilizar en el campo de la Inteligencia Artificial, Informática teórica (Ingeniería de Software, Ingeniería del Conocimiento, Ingeniería Web y Sistemas de Información), e Informática aplicada (Informática Médica e Informática Educativa). En el campo de la documentación, las ontologías son una herramienta para el intercambio y uso del conocimiento ya que proveen una comprensión compartida y consensuada de un dominio del conocimiento, que puede ser comunicada entre personas y sistemas heterogéneos (Lapuente, 2013).

Las ontologías proveen un vocabulario común y sin ambigüedades para referirse a los términos de un área específica, pudiéndose compartir o reutilizar éstos entre diferentes aplicaciones que hagan uso de la ontología. Además de un vocabulario común, especifican una taxonomía o herencia de conceptos que establecen una categorización o clasificación de las entidades del dominio. Una buena taxonomía es simple y fácil de recordar, separa sus entidades de forma mutuamente excluyente, y define grupos y subgrupos sin ambigüedad. El vocabulario y la taxonomía representan un marco de trabajo conceptual para el análisis, discusión o consulta de información de un dominio. Una ontología incluye una completa generalización o especificación de sus clases y subclases, las cuales están formalmente especificadas (incluyendo sus relaciones e instancias) asegurando la consistencia en los procesos deductivos. La misma está implementada en un lenguaje específico de

representación ontológica (*ontology representation languages*) de manera que la especificación de sus clases, relaciones entre éstas y sus restricciones dependerán de las características de dicho lenguaje (Vitelli, 2011).

Según Navarro, una ontología define un conjunto de primitivas con las cuales se modela un dominio de conocimiento o discurso, en el que, dichas primitivas son típicamente clases, atributos y relaciones entre las clases (Navarro, 2012).

Según Edgar, y otros (Ricardo, y otros, 2012), una ontología define los términos a utilizar para describir y representar un área de conocimiento. Son utilizadas por las personas, las bases de datos y las aplicaciones que necesitan compartir un dominio de información. Incluyen además definiciones de conceptos básicos del dominio, y las relaciones entre ellos que son útiles para los ordenadores. Se puede concluir que una ontología es una manera de describir y representar un área de conocimiento. No sólo se utiliza para almacenar la información, sino también permite buscarla y recuperarla.

Por las características dichas anteriormente, fue necesario desarrollar una ontología que defina los términos y las relaciones básicas para la comprensión del proceso de evaluación de productos de software, así como las reglas para poder definir las extensiones de este tipo de vocabulario controlado.

1.3.1 Tipos de ontologías

Las ontologías según Nicola Guarino (Guarino, 1998), se clasifican de acuerdo con su dependencia y relación con una tarea específica desde un punto de vista, en cuatro tipos:

Ontologías de Alto nivel o Genéricas: describen conceptos muy generales como espacio, tiempo o acción que son independientes de un problema o dominio particular. Parece razonable tener ontologías de alto nivel unificadas para una gran cantidad de comunidades de usuarios. En relación con los sistemas de información, estas ontologías describirían conceptos básicos.

Ontologías de Dominio: se representa el conocimiento especializado pertinente de un dominio o subdominio.

Ontologías de Tareas o Técnicas básicas: describen una tarea, actividad o artefacto. Por ejemplo, la evaluación de la contaminación sonora en ambientes urbanos o la descripción de características generales de componentes, procesos o funciones.

Ontologías de Aplicación: describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

De todos los tipos de ontologías que se estudiaron, teniendo en cuenta su dependencia y relación con una tarea específica desde un punto de vista, se optó por la ontología de aplicación, ya que la ontología propuesta describe conceptos que dependen del proceso de evaluación de software definido para la Dirección de la Calidad UCI, es decir que contienen todos los conceptos asociados a dicho proceso.

1.3.2 Componentes de una ontología

Como las ontologías constituyen una forma de representar el conocimiento que se tiene de un dominio, se han generalizado los elementos que la componen (Lapuente, 2013). Estos son:

Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc. La forma más común de definir los conceptos es mediante clases, que no son más que una manera de definir las propiedades y el estado interno que tendrán los objetos.

Propiedades (slots): son las características o atributos que describen a los conceptos.

Relaciones: representan la interacción y enlace entre los conceptos del dominio y especifican las dependencias entre unos y otros. Una relación está definida por un dominio, formado por una o más clases a las que se les aplica la relación directamente, mientras que el rango está definido por una o más clases que pueden ser aplicadas a la relación.

Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.

Instancias: representan objetos determinados de una clase, son las implementaciones específicas de un concepto, con valores concretos a sus propiedades, que permiten establecer unicidad entre todos ellos.

Axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son instancias de la clase C, entonces A no puede ser subclase de B”.

1.4 Metodologías para el desarrollo de ontologías

Las metodologías de desarrollo ontológico comprenden un conjunto establecido de principios, procesos, prácticas, métodos y actividades usados para el diseño, construcción, evaluación y puesta en producción de las mismas. Dichas metodologías, incluyen métodos para la unificación, reingeniería, mantenimiento y evolución de las ontologías (Vitelli, 2011). A continuación, se detallan las características esenciales de las metodologías más destacadas en la actualidad para la realización de ontologías.

CYC

Metodología CYC, publicada por Lenat y Guha desde 1990 (Lenat, 1990) , en la que divulgaron algunos pasos generales para la construcción de ontologías; el primero consiste en extraer manualmente el conocimiento común que está implícito en diferentes fuentes para después, cuando se tenga suficiente conocimiento en la ontología, adquirir nuevo conocimiento común usando herramientas de procesamiento de lenguaje natural o aprendizaje computacional (Guzmán Luna, 2012).

Gruninger y Fox

Metodología de Gruninger y Fox (Gruninger, y otros, 1995), cuyo primer paso consiste en identificar intuitivamente las aplicaciones posibles en las que se usará la ontología. Luego, se utilizan un conjunto de preguntas en lenguaje natural, llamadas cuestiones de competencia, para determinar el ámbito de la ontología. Se usan estas preguntas para extraer los conceptos principales, sus propiedades, relaciones y axiomas, los cuales se definen formalmente en prolog. Esta metodología fue usada para construir las ontologías del proyecto Toronto Virtual Enterprise (TOVE por sus siglas en inglés) en el Enterprise Integration Laboratory de la Universidad de Toronto. Dichas ontologías constituyen un modelo integrado y formalizado usando lógica de primer orden que incluye Enterprise Design Ontology, Project Ontology, Scheduling Ontology y Service Ontology (Guzmán Luna, 2012).

Metodología de Uschold y King

Metodología de Uschold y King (Uschold, et al., 1995), recrean una serie de pasos que permiten plasmar y especificar los conocimientos que se tienen sobre un dominio específico, centrando sus esfuerzos en la forma en la cual representar los conocimientos. Entre sus pasos para desarrollar ontologías propone: (i) identificar el

propósito; (ii) capturar los conceptos y relaciones entre estos conceptos y los términos utilizados para referirse a estos conceptos y relaciones; (iii) codificar la ontología. La ontología debe ser documentada y evaluada, y se pueden usar otras ontologías para crear la nueva. El proyecto más importante que se desarrolló usando esta metodología es *The Enterprise Ontology*, que es una colección de términos y definiciones relevantes a empresas de negocios. La ontología fue desarrollada bajo el *Enterprise Project* del *Artificial Intelligence Applications Institute* de la universidad de Edimburgo, con la colaboración de IBM (Guzmán Luna, 2012).

Methontology

La metodología Methontology, es una de las propuestas más completas para el desarrollo ontológico, ya que toma la creación de ontologías como un proyecto informático. Así, además de las actividades propias de la generación de la ontología esta metodología abarca actividades para la planificación del proyecto, la calidad del resultado y la documentación. Además, permite construir ontologías totalmente nuevas o reutilizar otras ontologías. El entorno incluye la identificación del proceso de desarrollo de la ontología donde se incluyen las principales actividades (evaluación, conceptualización, configuración, integración e implementación), un ciclo de vida basado en prototipos evolucionados y la metodología propiamente dicha, que especifica los pasos a ejecutar en cada actividad, las técnicas usadas, los productos a obtener y su forma de evaluación. Esta metodología está parcialmente soportada por el entorno de desarrollo ontológico WebODE y propone las siguientes etapas: (i) especificación, (ii) conceptualización, (iii) formalización, (iv) implementación y (v) mantenimiento (Guzmán Luna, 2012).

Actividades de la metodología Methontology

Las actividades de desarrollo identificadas para la metodología Methontology y por las que se realizó la ontología son las siguientes (Vitelli, 2011):

1. **Especificación:** permitió determinar por qué se construye la ontología, cuál será su uso, y quiénes serán sus usuarios finales.
2. **Conceptualización:** consistió en crear un glosario de términos que pertenecen al dominio, definirlos y crear una taxonomía (estableciendo una clasificación o jerarquía entre los conceptos, sus niveles, las

relaciones entre ellos, sus instancias, sus propiedades o atributos, e igualmente los axiomas o reglas). También existen componentes de modelado de ontologías que deben describirse detalladamente:

- **Tarea 1:** Construir el glosario de términos.
 - **Tarea 2:** Construir la taxonomía de conceptos.
 - **Tarea 3:** Construir un diagrama de relaciones binarias.
 - **Tarea 4:** Construir el diccionario de conceptos.
 - **Tarea 5:** Definir las relaciones binarias en detalle.
 - **Tarea 6:** Describir atributos de instancias.
 - **Tarea 7:** Describir atributos de clase
 - **Tarea 8:** Describir constantes.
 - **Tarea 9:** Describir reglas o axiomas.
 - **Tarea 10:** Describir instancias.
3. **Formalización:** se encargó de la transformación del modelo conceptual generado en la actividad anterior en un modelo formal o semi-computable.
 4. **Implementación:** construyó modelos computables en un lenguaje de ontologías (Ontolingua, RDF Schema, OWL).
 5. **Mantenimiento:** permitió la actualización y corrección de la ontología.

Después del análisis de cada metodología propuesta se decidió utilizar la metodología Methontology pues constituye una guía para llevar a cabo todo el desarrollo de ontologías en cualquier dominio a través de la especificación, la conceptualización, la formalización, la implementación y mantenimiento. Además, esta cuenta con una gran comunidad de usuarios, es una de las metodologías más completas entre las existentes ya que

divide sus actividades desde la especificación hasta el mantenimiento de manera secuencial, algo que facilita el trabajo futuro en la ontología que se quiere desarrollar.

1.5 Ontologías existentes en la calidad de software

Durante el desarrollo de la ontología es importante analizar la posibilidad de la reutilización de alguna ontología o cualquier otra fuente de información ya existente en el dominio seleccionado; con el propósito de rediseñar y ampliar los recursos que estas tienen con relación al dominio. Esta reutilización permite intercambiar con otras herramientas que utilizan ontologías ya definidas y validadas, así como también ahorrar esfuerzos. Existen propuestas de investigación relacionadas con el tema en la bibliografía estudiada como por ejemplo en (Lugones, 2017) , (Marín, y otros, 2011), (Hernández, 2015), (Cardoso, et al., 2013), (Morón, 2015), (Rodríguez, 2017) y (Santiago, 2015). Como principal dificultad para reutilizar estas propuestas está, que los autores no proporcionaron código generado, ni es posible acceder a ello. Además, las propuestas estudiadas tratan un dominio que, aunque tienen relación con el de la presente investigación no lo cubre totalmente puesto que en este trabajo se trata de representar todo el conocimiento generado en el proceso de evaluación del producto de software, completamente adaptado a las características de la Dirección de la Calidad de Software.

Por todas las dificultades para reutilizar las propuestas de investigación dichas anteriormente, se decidió desarrollar la ontología desde cero y no reutilizar algunas de las propuestas ya reflejadas.

1.6 Herramientas y tecnologías para la construcción de ontologías

En este epígrafe se realizó un estudio acerca de las principales herramientas utilizadas en la construcción de ontologías, haciendo énfasis en las herramientas de desarrollo, los razonadores que pueden ser utilizados en dichas herramientas y en la herramienta CASE a utilizar para diseñar el modelo conceptual.

1.6.1 Herramientas para el desarrollo de ontologías

Existen algunas herramientas para el desarrollo de ontologías tales como, Karlsruhe Ontology (KAON por sus siglas en inglés), WebODE y Protégé. Por requerimiento del cliente la herramienta a utilizar fue el Protégé. A continuación, se presentan las características más relevantes de dicha herramienta.

Protégé

Protégé, es una herramienta para el desarrollo de ontologías y sistemas basado en el conocimiento desarrollada en la Universidad de Stanford. La misma está desarrollada en Java por lo que puede ejecutarse en cualquier plataforma que soporte la máquina virtual de Java. Las ontologías desarrolladas con dicha herramienta son empleadas en resolución de problemas y toma de decisiones en dominios particulares. Además, de proporcionar una herramienta visual para el desarrollo de ontologías, Protégé incluye una herramienta de desarrollo (*PDK - Protégé Development - Kit*) para la gestión de sus repositorios, que proporciona, desde el código la capacidad de gestión de múltiples características tales como la persistencia del modelo, la ejecución de consultas o la modificación del propio esquema del modelo. En el caso particular de modelos OWL, esta herramienta ofrece la biblioteca Protégé - OWL para la gestión de modelos OWL. Protégé también reconoce, Frames, XML Schema, RDF Schema y OWL, que son lenguajes semánticos utilizados en la Web, en contraposición a la rigidez del HTML (Carrión, 2011).

1.6.2 Razonadores de ontologías

Una de las herramientas más utilizadas para trabajar con las ontologías son los razonadores, que sirven para realizar inferencia, a través de los conceptos y en algunos casos las instancias, obteniendo nuevo conocimiento. Hoy en día, son varios los razonadores o sistemas deductivos basados en lógica descriptiva que permiten el razonamiento y la inferencia en las ontologías. Los principales son:

JFact

JFact es un razonador desarrollado en la Universidad de Manchester, publicado bajo licencia LGPL. JFact es un puerto de Java puro de FaCT ++, con versiones para OWL-API 3.x y 4.x Se mantiene paralelamente con FaCT ++ y se actualiza regularmente. Se ha utilizado en los dispositivos android con OWL-API 3.5. Está disponible y viene incorporado como un plugins de Protégé, versiones 4.3 y 5. Posee Interfaces compatibles con: API OWL. Además, los servicios de razonamiento son compatibles en: realización, clasificación, vinculación y consistencia (Manchester, 2016).

FaCT++

FaCT++ es un razonador basado en lógica descriptiva (DL) que ha sido desarrollado bajo el proyecto europeo WonderWeb en la Universidad de Manchester. Está implementado en C++ y corresponde a una nueva versión del razonador DL FaCT. Para realizar inferencia implementa nuevas características y optimizaciones, permitiendo adicionar nuevas tácticas de razonamiento y capacidad de razonar con lógicas descriptivas más potentes y cercanas a la expresividad del lenguaje OWL. Entre las ventajas que ofrece FaCT++ están: posee licencia GPL. Además, Protégé 4.0.1 y versiones posteriores lo tienen incluido en sus plugins (Manchester, 2016).

HermiT

HermiT es razonador para ontologías escritas utilizando el Lenguaje de Ontologías Web (OWL). Dado un archivo OWL, HermiT puede determinar si es o no la ontología consistente, además puede identificar las relaciones de subsunción entre clases, y mucho más. HermiT es el primer razonador OWL disponible al público basado en cálculo "hypertableau", que ofrece un razonamiento mucho más eficiente que cualquier algoritmo previamente conocido. Permite clasificar las ontologías de manera rápida, mientras que otros razonadores tardan minutos y horas, por tanto, debido a esta característica HermiT es el primer razonador capaz de clasificar un número significativo de ontologías, cosa que ha sido demostrado previamente que es demasiado complejo para cualquier otro razonador existente (Manchester, 2016).

Pellet

Pellet fue el primer razonador que soportaba todo de OWL-DL. Mediante su uso es posible validar, comprobar la consistencia de ontologías, clasificar la taxonomía y contestar a un subconjunto de consultas Resource Description Query Language (RDQL por sus siglas en inglés) conocidas como consultas a ABox en terminología del DL. Se trata de un razonador DL basado en los algoritmos "tableaux" desarrollados para DL expresiva (Manchester, 2016).

Para la realización de inferencias se propuso como razonador a utilizar Pellet, aunque cualquier otro razonador que soporte OWL pudo haberse usado en lugar de éste, puesto que la ontología que se realizó provee acceso uniforme a razonadores basados en lógica descriptiva. Además, Pellet posee múltiples ventajas ya que está implementado en Java, lo cual permitió hacer inferencias y consultas basadas en ontologías descritas en OWL.

También, esta librería está desarrollada bajo un esquema de licenciamiento de código abierto y es de fácil integración con Protégé.

1.6.3 Herramientas CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE) son aplicaciones informáticas dedicadas al aumento de la productividad en el desarrollo de software. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas, como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (CASE, 2010).

Fue necesario utilizar como herramienta CASE para la realización del modelo conceptual el Visual Paradigm, el cual esta detallado a continuación.

Visual Paradigm 8.0

Visual Paradigm 8.0 es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software. El Lenguaje de Modelado Unificado (UML) permite una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite modelar todos los tipos de diagramas de clases, así como generar código inverso, código desde diagramas y documentación. Es fácil de instalar y actualizar además de ser compatible entre ediciones. Dentro de las principales características de la herramienta se encuentran: disponibilidad de múltiples versiones, de acuerdo a su necesidad, así como la capacidad de integrarse a los principales IDE (Entorno de Desarrollo Integrado). Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y se integra con varias herramientas Java (Eclipse/IBM WebSphere, NetBeans IDE). Es capaz de generar la documentación del proyecto en diferentes formatos como HTML o PDF y permite el control de versiones (Paradigm, 2012).

Dada la caracterización planteada anteriormente, se utilizó Visual Paradigm como herramienta CASE para la realización del modelo conceptual debido a su alta capacidad de integración. Es una herramienta multiplataforma y está disponible en varios idiomas. Además, es fácil de instalar y actualizar y facilita la organización de los diagramas generando la documentación del proyecto en varios formatos. También, es capaz de generar código, modelo de datos, entre otras funcionalidades útiles para el desarrollo de un producto.

1.7 Conclusiones parciales

En el presente capítulo se especificaron los conceptos fundamentales de la investigación para lograr una mejor comprensión de las ontologías y del proceso de evaluación de productos de software, donde se definieron: tipos, componentes, metodologías, herramientas y tecnologías para el desarrollo propuesta ontológica, lo cual nos permitió llegar a las siguientes conclusiones:

- ✚ Se constató que en las ontologías existentes presentadas, no se encontró una propuesta ontológica que cumpliera con las necesidades propias para el dominio especificado. En diferentes casos estas enmarcan un dominio menos abarcador que el que se quiere representar en este trabajo o en ocasiones no proporcionan el código generado de la ontología, ni puede accederse a los mismos.
- ✚ El editor Protégé permite realizar ontologías con extensión OWL y RDF, debido a su entorno gráfico basado en plugins. Es una de las herramientas más utilizadas para la construcción de ontologías facilitando el trabajo con las clases y las instancias. La base de conocimiento que resulta se puede utilizar como un método para responder a preguntas y para solucionar problemas con respecto al dominio. Además, puede ser utilizado conjuntamente con el razonador Pellet.

Capítulo 2: Diseño de la propuesta ontológica

Introducción

En el presente capítulo se efectúa el diseño y desarrollo de la propuesta ontológica, haciendo uso de las herramientas descritas en el capítulo anterior guiado por la metodología Methontology. El principal objetivo de dicha propuesta ontológica es organizar y recuperar el conocimiento generado en el proceso de evaluación del producto de software.

2.1 Descripción de la propuesta de solución

La propuesta ontológica pretende, organizar y recuperar la información existente en el proceso de evaluación de productos de software a los expertos de la Dirección de la Calidad de Software y lograr con esto un mayor nivel de comprensión de dicho proceso en las diferentes organizaciones y empresas desarrolladoras de productos de software. Se describen en ella, los principales conceptos, términos y las relaciones entre los mismos dentro de dicho proceso, facilitando al experto una manera más entendible y computable de los datos del dominio en cuestión. Además, permitirá representar el conocimiento a manejar dentro del mismo. A continuación, se muestra una breve descripción con las principales clases y relaciones del proceso de evaluación de productos de software:

En una evaluación de software interviene el producto a evaluar y se realizó sobre la base de un modelo de calidad. El producto pertenece a un proyecto de desarrollo en el que intervienen especialistas. Estos desarrollan artefactos y en este accionar manipulan herramientas y consultan documentación que los guían en su proceso evaluativo. El modelo de calidad descompone la calidad en características de calidad, que a su vez se descomponen en sub-características de calidad las cuales se cuantifican por las medidas de calidad. Estas medidas de calidad se obtienen combinando los elementos de medida de calidad. Cada característica para su valoración utiliza técnicas de evaluación, que son seleccionados de acuerdo al nivel de evaluación. Los niveles de evaluación están relacionados con el grado de rigor asociado a la evaluación de esa característica, que a su vez constituye una guía para seleccionar las técnicas de evaluación. Se utilizan 4 niveles de evaluación (A, B, C y D). A representa el más alto nivel y D el más bajo. En el nivel A son aplicadas las técnicas más estrictas. El proceso de evaluación se conduce a través de un flujo de actividades entre las que se encuentran la

actualización del Expediente de la evaluación. En dicha actividad se detectan algunas No conformidades. Es necesario registrar la evaluación de los productos de software.

En la siguiente figura se muestra el modelo conceptual, confeccionado con la herramienta Visual Paradigm for UML 8.0 Enterprise Edition, en la misma se representan las clases y relaciones que se utilizaron para el diseño de la ontología (Ver figura 2).

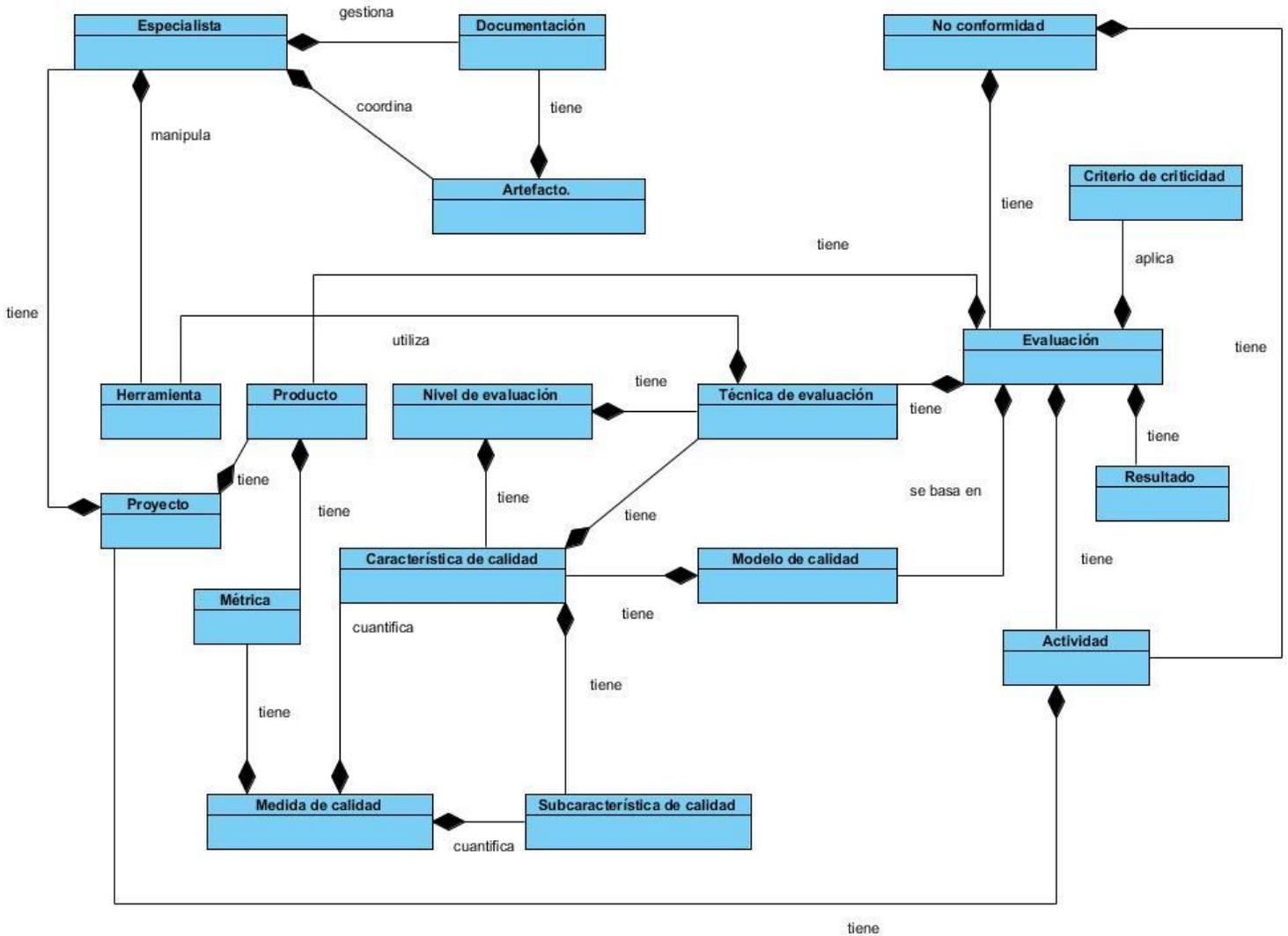


Figura. 2: Modelo conceptual del Proceso de Evaluación de Software. Fuente: Elaboración propia.

2.2 Aplicación de la metodología Methontology

Como se mencionó en el capítulo anterior la metodología Methontology se basa en las actividades de Especificación, Conceptualización, Formalización, Implementación y Mantenimiento. Las cuáles serán desarrolladas a continuación:

2.2.1 Especificación

Para la especificación de la ontología se creó una plantilla con el nombre Documento de Especificación de Requerimientos. En esta plantilla se coloca el dominio al que se refiere la ontología, fecha en que comienza el desarrollo, quiénes son los desarrolladores, cuál es el propósito, qué nivel de formalidad alcanzará la ontología, su alcance especificando las preguntas de competencia y cuáles serán las fuentes de conocimiento (Vitelli, 2011). En el Tabla 1 muestra dicha plantilla para esta ontología.

Tabla 1: Documento de Especificación de Requerimientos.

Dominio	Proceso de Evaluación de Software en la Dirección de la Calidad UCI.
Fecha	18 de enero del 2018
Desarrollador(es)	Leosmay Carrión Estradet
Propósito	Construir una ontología con el objetivo de proporcionar una mejor organización, representación y recuperación para la base de conocimientos existente en la Dirección de la Calidad UCI.
Nivel de Formalidad	Formal
Alcance	¿Qué tipo de evaluación tiene el producto1? ¿Qué especialista manipula la herramienta1? ¿Qué especialista coordina el artefacto Instalador de aplicación? ¿Qué actividad se realiza en la evaluación 1? ¿Qué especialista gestiona la documentación2 y 4? ¿Qué tipo de métrica utiliza el producto1? ¿En cuál nivel de evaluación se aplica la técnica de evaluación Verificación del uso de facilidades de lenguajes de programación específicos? ¿Qué tipo de actividad utiliza la nc1? ¿Cuál es el proyecto al que pertenece el producto1?

Fuentes de Conocimiento	Expertos de la Dirección de la Calidad de Software de la UCI, Calisoft, XETID, SOFTEL y especialistas del Subcomité 7 de la ISO en Cuba.
--------------------------------	--

2.2.2 Conceptualización

La conceptualización es la segunda actividad de la metodología Methontology, se basa en organizar y convertir una percepción informal del dominio en una especificación semi-formal que pueden ser fácilmente comprendidas. Esta actividad define once tareas para el modelado de ontologías. A continuación, se describen detalladamente cada tarea que compone la actividad.

Tarea 1: Construir el glosario de términos.

En esta tarea cuyo propósito es construir el glosario de términos, define cada término relevante del dominio incluyendo los conceptos, instancias y relaciones (Ver tabla 2).

Tabla 2: Glosario de términos.

No.	Término	Descripción	Tipo
1	Evaluación	Determinación sistemática del grado en que una entidad cumple con los criterios especificados.	Concepto
2	Artefacto	Entregable que generan en cada uno de las etapas del desarrollo del software.	Concepto
3	Producto	Un conjunto de programas informáticos, procedimientos y posiblemente documentación y datos asociados.	Concepto
4	Modelo de calidad	Conjunto de características definidas y relaciones entre ellas que proporciona una estructura para especificar los requisitos y evaluación de la calidad.	Concepto
5	Criterio de criticidad	Elementos o bases para determinar el estado a un proyecto al presentar un número considerable de No Conformidades.	Concepto
6	Sub-característica de calidad	El concepto Característica de calidad tiene Sub-característica de calidad.	Concepto
7	Medida de calidad	Medida interna de la calidad del software, medida externa de la calidad del software o medida de la calidad en el uso del software.	Concepto

8	Proceso de evaluación de software	Operación técnica que consiste en realizar una evaluación de una o más características a un producto de software de acuerdo con un procedimiento especificado.	Concepto
9	Proyecto	Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos, incluyendo las limitaciones de tiempo, costo y recursos.	Concepto
10	Técnica de evaluación	Métodos y habilidades necesarios para llevar a cabo una actividad específica.	Concepto
11	No Conformidad (NC)	Problemas detectados en un artefacto según la insatisfacción con el resultado final de un Elemento de Configuración, lo pactado con anterioridad con el cliente, o no cumplimiento de un requisito.	Concepto
12	Característica de calidad	Categoría de los atributos de la calidad del software que influye en la calidad del software.	Concepto
13	Especialista	Persona encarga de coordinar la evaluación en la Dirección de la Calidad de Software.	Concepto
14	Herramienta	Instrumento que puede utilizarse durante la evaluación para recolectar datos, llevar a cabo la interpretación de los mismos o automatizar parte de la evaluación.	Concepto
15	Resultado	Informe de evaluación de la calidad del producto de software.	Concepto
16	Nivel de evaluación	Rigor a aplicarse durante la evaluación que define la profundidad o la exhaustividad de esta en términos de técnicas a aplicarse y los resultados de la evaluación a alcanzarse.	Concepto
17	Actividad	Etapas por la que transita el proceso de evaluación.	Concepto
18	Documentación	Documentos generados en cada una de las actividades del proceso de evaluación.	Concepto
19	Métrica	Medida, que es una medida base o una derivada, utilizada para la construcción de las medidas de la calidad del software.	Concepto
20	act1	Instancia perteneciente a la clase Actividad	Instancia
21	artefacto1	Instancia perteneciente a la clase Artefacto	Instancia

No.	Término	Descripción	Tipo
22	cc1	Instancia perteneciente a la clase Características de calidad	Instancia
23	crierioc1	Instancia perteneciente a la clase Criterio de criticidad	Instancia
24	documentación1	Instancia perteneciente a la clase Documentación	Instancia
25	especialista1	Instancia perteneciente a la clase Especialista	Instancia
26	evaluación1	Instancia perteneciente a la clase Evaluación	Instancia
27	herramienta1	Instancia perteneciente a la clase Herramienta	Instancia
28	medida de calidad1	Instancia perteneciente a la clase Medida de calidad	Instancia
29	mc1	Instancia perteneciente a la clase Modelo de calidad	Instancia
30	métrica1	Instancia perteneciente a la clase Métrica	Instancia
31	nivel1	Instancia perteneciente a la clase Nivel de evaluación	Instancia
32	nc1	Instancia perteneciente a la clase No Conformidad	Instancia
33	producto1	Instancia perteneciente a la clase Producto	Instancia
34	proyecto1	Instancia perteneciente a la clase Proyecto	Instancia
35	resultado1	Instancia perteneciente a la clase Resultado	Instancia
36	subcaract1	Instancia perteneciente a la clase Sub-característica de calidad	Instancia
37	técnica1	Instancia perteneciente a la clase Técnica de calidad	Instancia
38	tiene Evaluación	Es la relación que existe los conceptos Proceso evaluación de software y el concepto Evaluación.	Relación
39	tiene Actividad	Es la relación que existe entre los conceptos No Conformidad y Actividad.	Relación
40	tiene actividad	Es la relación que existe entre los conceptos Evaluación y Actividad.	Relación
41	tiene característica de calidad	Es la relación que existe entre los conceptos Nivel de evaluación y Característica de calidad.	Relación
42	tiene característica de calidad	Es la relación que existe entre los conceptos Modelo de calidad y Característica de calidad.	Relación
43	tiene documentación	Es la relación que existe entre los conceptos Artefacto y Documentación.	Relación
44	tiene especialista	Es la relación que existe entre los conceptos Proyecto y Especialista.	Relación

No.	Término	Descripción	Tipo
45	tiene evaluación	Es la relación que existe entre los conceptos Proceso de evaluación de software y Evaluación.	Relación
46	tiene métrica	Es la relación que existe entre los conceptos Medida de calidad y Métrica.	Relación
47	tiene métricaP	Es la relación que existe entre los conceptos Producto y Métrica.	Relación
48	tiene modelo de calidad	Es la relación que existe entre los conceptos Evaluación y Modelo de calidad.	Relación
49	tiene producto	Es la relación que existe entre los conceptos Proyecto y Producto.	Relación
50	tiene productoE	Es la relación que existe entre los conceptos Evaluación y Producto.	Relación
51	utiliza herramienta	Es la relación que existe entre los conceptos Técnica de evaluación y Herramienta.	Relación
52	aplica criterio de criticidad	Es la relación que existe entre los conceptos Evaluación y Criterio de criticidad.	Relación
53	aplica técnica de evaluación	Es la relación que existe entre los conceptos Nivel de evaluación y Técnica de evaluación.	Relación
54	coordina artefacto	Es la relación que existe entre los conceptos Especialista y Artefacto.	Relación
55	cuantifica característica de calidad	Es la relación que existe entre los conceptos Medida de calidad y Característica de calidad.	Relación
56	cuantifica sub-característica de calidad	Es la relación que existe entre los conceptos Medida de calidad y Sub-característica de calidad.	Relación
57	gestiona documentación	Es la relación que existe entre los conceptos Especialista y Documentación.	Relación
58	tiene resultado	Es la relación que existe entre los conceptos Evaluación y Resultado.	Relación
59	tiene sub-característica de calidad	Es la relación que existe entre los conceptos Característica de calidad y Sub-característica de calidad.	Relación

60	tiene técnica de evaluación	Es la relación que existe entre los conceptos Característica de calidad y Técnica de evaluación.	Relación
61	tiene técnica de evaluación ²	Es la relación que existe entre los conceptos Evaluación y Técnica de evaluación.	Relación
62	tiene proyecto	Es la relación que existe entre los conceptos Actividad y Proyecto.	Relación

Tarea 2: Construir la taxonomía de conceptos.

Cuando el glosario de términos tenga una cantidad importante de elementos, se construye una taxonomía que defina la jerarquía entre los conceptos. Se debe evaluar que la taxonomía creada no contenga errores. Las taxonomías representan a través de las clases y las instancias propias de esas clases una realidad contextualizada que permitió la organización y recuperación efectiva de los conceptos representados pertenecientes a un determinado dominio. Para construir la taxonomía de conceptos, se seleccionaron del glosario de términos aquellos términos que son conceptos, por lo que hay que tener en cuenta que un concepto puede ser subclase de más de un concepto en la taxonomía (McGuinness, 2005). En la Figura. 3 se muestra la taxonomía para los conceptos Característica de calidad, Herramienta, Artefacto, Evaluación, Criterio de criticidad y Métrica, las restantes se pueden observar en el Anexo 1.

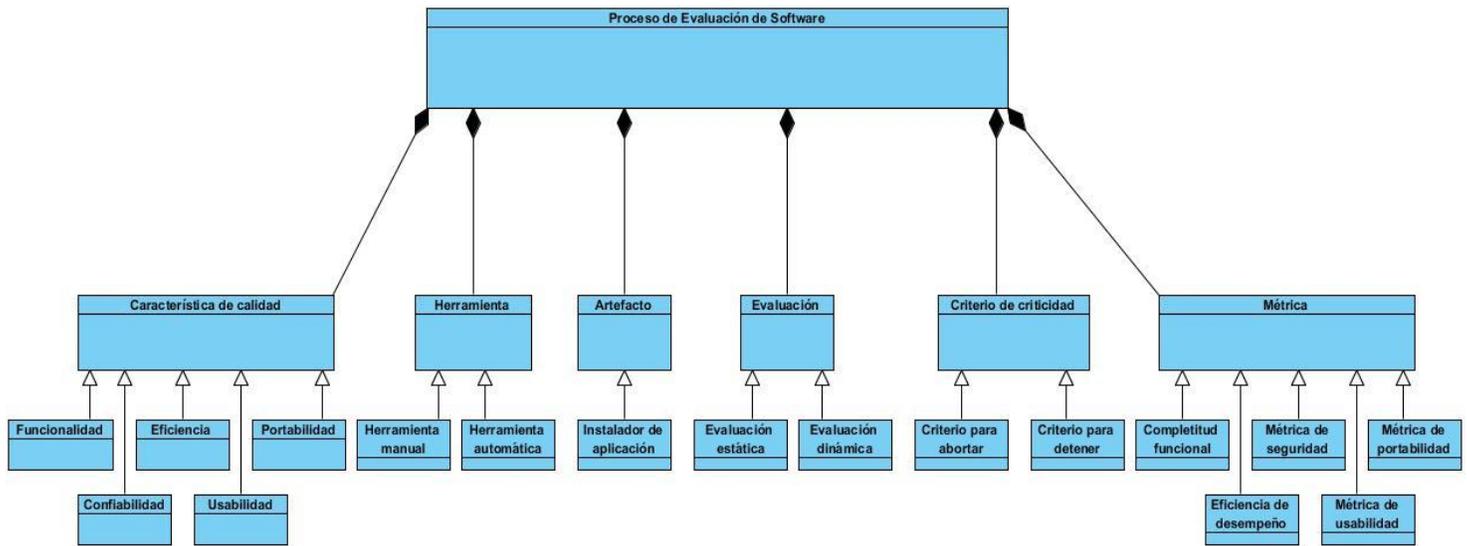


Figura. 3: Taxonomía de conceptos de la ontología Proceso de Evaluación de Software. Fuente: Elaboración propia.

Tarea 3: Construir un diagrama de relaciones binarias.

Luego de haber construido y evaluado la taxonomía, la actividad de conceptualización plantea construir diagramas de relaciones binarias. El propósito de este diagrama es definir las relaciones existentes entre conceptos de una o más taxonomías de conceptos. En el diagrama de clases mostrado anteriormente todas las relaciones que están son relaciones binarias, pero en la Figura. 4 se muestra una relación binaria inversa de uno de los conceptos del diagrama, Característica de calidad y Sub-característica de calidad.



Figura. 4: Diagrama de relaciones binarias de la ontología Proceso de Evaluación de Software. Fuente:Elaboración propia.

Tarea 4: Construir el diccionario de conceptos.

Después de crear las taxonomías de conceptos y los diagramas de relaciones binarias se especifica cuáles son las propiedades que describen cada concepto de la taxonomía. El diccionario de conceptos contiene los conceptos del dominio, sus relaciones, instancias, atributos de clases y atributos de instancias. Las relaciones,

atributos de instancias, y atributos de clases son locales al concepto, lo que significa que sus nombres pueden repetirse en diferentes conceptos (Ver tabla 3).

Tabla 3: Diccionario de conceptos de la ontología Proceso de Evaluación de Software.

Nombre del concepto	Instancia	Atributos de clase	Atributos de instancia	Relaciones
Evaluación	evaluación1		fecha Inicio fecha Fin	tiene actividad tiene modelo de calidad aplica criterio de criticidad tiene resultado tiene técnica de evaluación2 tiene productoE
Artefacto	artefacto1		denominación descripción documentación	tiene documentación
Sub-característica de calidad	subcaract1 subcaract2 subcaract3 subcaract4 subcaract5 subcaract6 subcaract7 subcaract8 subcaract9 subcaract10 subcaract11 subcaract12 subcaract13 subcaract14 subcaract15		denominación descripción	---

	subcaract16 subcaract17 subcaract18 subcaract19 subcaract20 subcaract21			
Métrica	métrica1		denominación descripción	---
Producto	producto1		denominación descripción	tiene métricaP
Modelo de calidad	mc1		denominación descripción	tiene característica de calidad
Criterio de criticidad	criterio1		denominación descripción	---
No Conformidad (NC)	nc1		descripción número de iteración	tiene Evaluación tiene Actividad
Medida de calidad	medida de calidad1		denominación descripción	cuantifica característica de calidad cuantifica sub-característica de calidad
Herramienta	herramienta1 herramienta2		denominación versión descripción	---
Especialista	especialista1 especialista2 especialista3 especialista4		nombre responsabilidad descripción	manipula herramienta coordina artefacto
Característica de calidad	cc1 cc2 cc3 cc4		denominación consecuencia descripción	tiene sub-característica de calidad

	cc5			
Nombre del concepto	Instancia	Atributos de clase	Atributos de instancia	Relaciones
Documentación	documentación1 documentación3 documentación2 documentación4		denominación descripción	---
Nivel de evaluación	nivel1		denominación descripción	Ejecuta
Proyecto	proyecto1		denominación descripción	tiene producto tiene especialista
Actividad	act1		denominación descripción fecha Inicio fecha Fin sesiones de trabajo responsable participante observación	tiene proyecto
Técnica de evaluación	técnica1 técnica2 técnica3 técnica4 técnica5 técnica6 técnica7 técnica8 técnica9 técnica10 técnica11 técnica12		denominación descripción	utiliza herramienta

	técnica13 técnica14 técnica15			
Resultado	resultado1		descripción	---

Tarea 5: Describir las relaciones binarias en detalle.

Después de haber construido el diccionario de concepto, se procedió a definir las relaciones binarias en detalle. El objetivo de esta tarea es detallar todas las relaciones binarias descritas en el diagrama de relaciones binarias e incluidas en el diccionario de conceptos. Para cada relación binaria, se especifica su nombre, los nombres de sus conceptos origen y destino, su cardinalidad y su relación inversa, si existe.

En la Tabla. 4 se definen las relaciones binarias de la ontología Proceso de Evaluación de Software.

Tabla 4: Relación binaria de la ontología Proceso de Evaluación de Software.

Nombre de la relación	Concepto Origen	Cardinalidad	Concepto Destino	Relación Inversa
aplica criterio de criticidad	Evaluación	1...n	Criterio de criticidad	es aplicado por la evaluación
aplica técnica de evaluación	Nivel de evaluación	1...n	Técnica de evaluación	es aplicado por el nivel de evaluación
coordina artefacto	Especialista	1...n	Artefacto	es coordinado por
cuantifica característica de calidad	Medida de calidad	1...n	Característica de calidad	es cuantificada por Medida de calidad
cuantifica sub-característica de calidad	Medida de calidad	1...n	Sub-característica de calidad	es cuantificada por medida de calidad
gestiona documentación	Especialista	1...n	Documentación	es gestionada por
manipula herramienta	Especialista	1...n	Herramienta	es manipulada por
tiene Actividad	No Conformidad	1...n	Actividad	es actividad de No Conformidad
tiene actividad	Evaluación	1...n	Actividad	es actividad de Evaluación
tiene característica de calidad	Nivel de evaluación	1...n	Característica de calidad	es característica de calidad de nivel de evaluación
tiene característica de	Modelo de calidad	1...n	Característica de	Es característica de

Nombre de la relación	Concepto Origen	Cardinalidad	Concepto Destino	Relación Inversa
calidad			calidad	calidad del modelo de calidad
tiene documentación	Artefacto	1...n	Documentación	es documentación de
tiene especialista	Proyecto	1...n	Especialista	pertenece al proyecto
tiene Evaluación	No Conformidad	1...n	Evaluación	es evaluación de No Conformidad
tiene evaluación	Proceso de evaluación de software	1...n	Evaluación	es evaluación del Proceso de evaluación de software
tiene métrica	Medida de calidad	1...n	Métrica	es métrica de la medida de calidad
tiene métricaP	Producto	1...n	Métrica	es métrica de producto
tiene modelo de calidad	Evaluación	1...n	Modelo de calidad	es modelo de calidad de
tiene producto	Proyecto	1...n	Producto	es producto de proyecto
tiene productoE	Evaluación	1...n	Producto	es producto de evaluación
tiene proyecto	Actividad	1...n	Proyecto	es proyecto de actividad
tiene resultado	Evaluación	1...n	Resultado	es resultado de
tiene sub-característica de calidad	Característica de calidad	1...n	Sub-característica de calidad	es sub-característica de calidad de
Tiene técnica de evaluación	Característica de calidad	1...n	Técnica de calidad	es técnica de evaluación de característica de calidad
tiene técnica de evaluacion2	Evaluación	1...n	Técnica de calidad	es una técnica de Evaluación
utiliza herramienta	Técnica de calidad	1...n	Herramienta	es utilizada por la técnica de evaluación

Tarea 6: Describir atributos de instancias.

El objetivo de esta actividad es describir todos los atributos de instancias incluidos en el diccionario de conceptos. Para ello, en cada atributo se identificaron los campos nombre, concepto al que pertenece (teniendo en cuenta que los atributos son locales al concepto), el tipo de valor, el valor del rango en caso de ser numérico y la cardinalidad (Ver tabla 5).

Tabla 5: Descripción de los atributos de instancias.

Atributo de instancia	Concepto	Tipo de valor	Rango	Cardinalidad
Descripción	Actividad Artefacto Característica de calidad Documentación Criterio de criticidad Especialista Evaluación Herramienta Medida de calidad Modelo de calidad Métrica Nivel de evaluación No conformidad Producto Proyecto Resultado Sub-características de calidad Técnica de evaluación	string	-	1...1
Nombre	Especialista	string	-	1...1
Versión	Herramienta	string	-	1...1
Denominación	Actividad Artefacto Característica de calidad Documentación Criterio de criticidad Herramienta Medida de calidad Modelo de calidad Métrica Nivel de evaluación No conformidad	string	-	1...1

	Producto Proyecto Sub-características de calidad Técnica de evaluación			
número de iteración	No Conformidad	int	-	1...1
Documentación	Artefacto	string	-	1...1
fecha Inicio	Actividad Evaluación	string	-	1...1
fecha Fin	Actividad Evaluación	string	-	1...1
denominación descripción sesiones de trabajo responsable participante observación	Actividad	string	-	1...1

Tarea 7: Describir atributos de clases.

El objetivo de esta tarea es describir todos los atributos de clase que fueron especificados en el diccionario de conceptos. Los atributos de clases representan características genéricas de un concepto, es decir todas las instancias del concepto tendrán el mismo valor para ese atributo. Para cada atributo de clase se especificaron el nombre, concepto donde se definió, tipo de valor y cardinalidad. En el diccionario de concepto este aspecto no se encuentra reflejado, debido a que los conceptos a manejar en la presente investigación, solo van a estar identificados a través de las instancias/individuos presentes en cada uno de ellos y no es necesario emplear atributos para caracterizar los conceptos.

Tarea 8: Describir constantes.

El objetivo principal de esta actividad es describir cada una de las constantes identificadas en el glosario de términos, donde solamente fueron identificados términos de tipo concepto y relaciones. Para cada constante se especificó el nombre, tipo de valor, valor y unidad de medida (para constantes numéricas). En la propuesta

ontológica no se hace uso de constantes, debido a que el conocimiento almacenado en la misma, está estructurado a través de los conceptos identificados y sus relaciones, los cuales no hacen uso de constantes.

Tarea 9: Describir reglas o axiomas.

Ahora nos corresponde definir reglas, estas se utilizan normalmente para inferir conocimientos en la ontología, tales como valores de atributos, instancias de relaciones, entre otras. La metodología Methontology para cada regla propone especificar la siguiente información: nombre, descripción en lenguaje natural, expresión que describe formalmente la regla, conceptos, y relaciones utilizados en la regla (Ver tabla 6). En el presente trabajo los conceptos reglas y axiomas se solapan. Es decir, tienen el mismo significado.

Tabla 6: Ejemplo de una regla en la ontología Proceso de Evaluación de Software.

Nombre de la regla	Descripción	Expresión	Conceptos	Relaciones
Evaluación dinámica	Evaluación dinámica tiene algún Producto.	Evaluación and (tiene productoE some Producto)	Evaluación dinámica Producto	tiene productoE
Especialista Probador	Especialista Probador gestiona alguna Documentación.	Especialista and (gestiona documentación some Caso de prueba)	Probador Caso de prueba	gestiona documentación
Característica de calidad Confiabilidad	Característica de calidad Confiabilidad tiene sola una Sub-característica de calidad y tiene solo una Técnica de evaluación.	Característica de calidad and (tiene sub-característica de calidad only Disponibilidad) and (tiene técnica de evaluación only Modelo de crecimiento de la fiabilidad)	Confiabilidad Disponibilidad Modelo de crecimiento de la fiabilidad	tiene sub-característica de calidad tiene técnica de evaluación
Documentación Manual de Instalación	Documentación Manual de Instalación es documentación de solo un Artefacto.	Documentación and (es documentación de only Instalador de aplicación)	Manual de Instalación Instalador de aplicación	es documentación de
Especialista Coordinador de la evaluación	Especialista Coordinador de la evaluación gestiona	Especialista and (gestiona documentación only Expediente de la evaluación) and (manipula	Coordinador de la evaluación Expediente de la	gestiona documentación manipula

	sola una Documentación y manipula solo una Herramienta.	herramienta only (Herramienta automática)	evaluación Herramienta automática	herramienta
Especialista de tipos de pruebas de sistema específicos	Especialista de tipos de pruebas de sistema específicos gestiona solo una Documentación.	Especialista and (gestiona documentación only Plan de evaluación del producto)	Especialista de tipos de pruebas de sistema específicos Plan de evaluación del producto	gestiona documentación
Especialista del equipo de desarrollo	Especialista del equipo de desarrollo coordina solo un Artefacto.	Especialista and (coordina artefacto only Instalador de aplicación)	Especialista del equipo de desarrollo Instalador de aplicación	coordina

Tarea 10: Describir instancias.

Mediante una tabla de instancias, se definen las instancias relevantes del diccionario de conceptos. Para cada instancia se especifican el nombre, nombre del concepto al que pertenece y valores de los atributos de instancias, si se conocen. Esta tarea no se aplicó para este dominio.

Al finalizar la actividad de conceptualización la ontología cuenta con 106 clases, 52 relaciones, 47 atributos y 63 individuos.

2.2.3 Formalización

A partir de esta actividad, se transformó el modelo conceptual que se construyó en las actividades anteriores a un modelo formal o semi-computable. Es decir, llevar los conceptos y las relaciones, atributos e instancias entre ellos a dicho modelo. Para ello se hace uso del editor de ontologías Protégé 4.3.

Creación de las clases

Las clases son creadas según la jerarquía definida en la taxonomía de conceptos, partiendo desde la clase padre “Thing”, que es una palabra clave que hace referencia a la W3C². Cada clase presenta: anotaciones, clases equivalentes, sub-clases, instancias y restricciones. En la Figura. 5 se puede apreciar en la parte izquierda la creación de las clases.

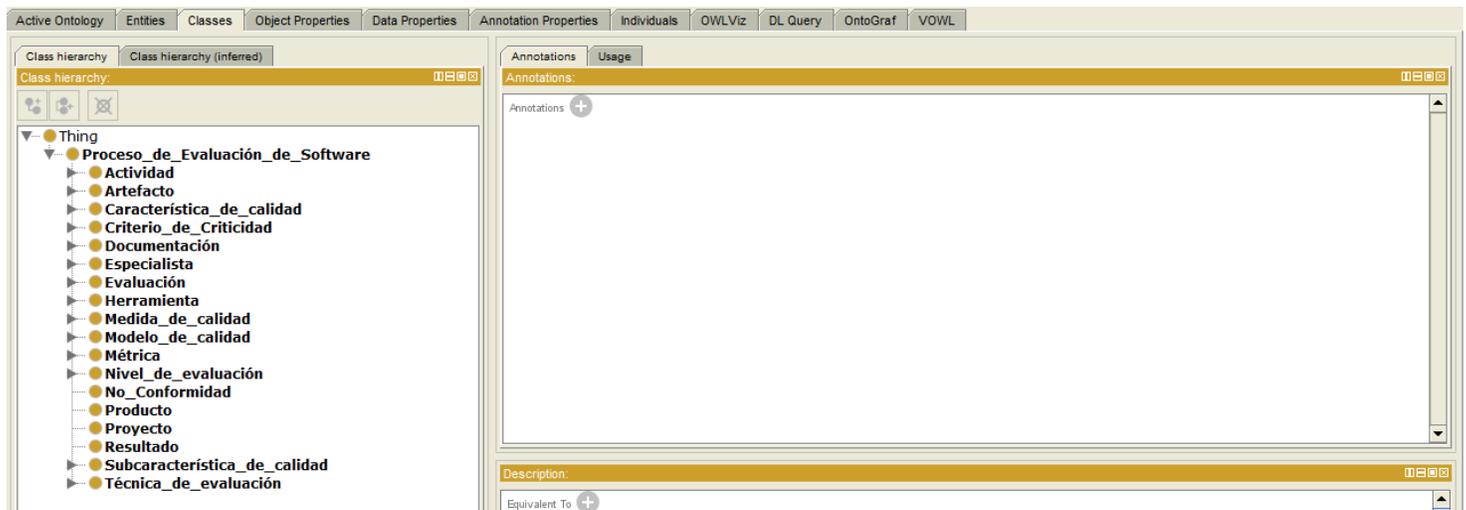


Figura. 5: Definición de las clases creadas en Protégé. Fuente: Elaboración propia.

Creación de las propiedades de las clases

Las propiedades son los elementos ontológicos que permiten definir las relaciones entre los conceptos del dominio de la ontología. Estas describen las cualidades internas de los conceptos y representan las propiedades diferentes de los objetos. Existen dos tipos principales de propiedades: los “*Object Properties*” y “*Datatype Properties*”. Los Object Properties sirven para definir propiedades que conectan una clase con otra, mientras que los Datatype Properties se utilizan para definir propiedades que conectan una clase con un tipo de dato. Mientras se añaden las Object Properties deben definirse las propiedades inversas de estas, también se especifica el dominio y rango, o sea las clases que se relacionan. En la Figura. 6 se pueden apreciar en la parte izquierda las propiedades de las clases y en la Figura. 7 la creación de las propiedades.

² World Wide Web Consortium, es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

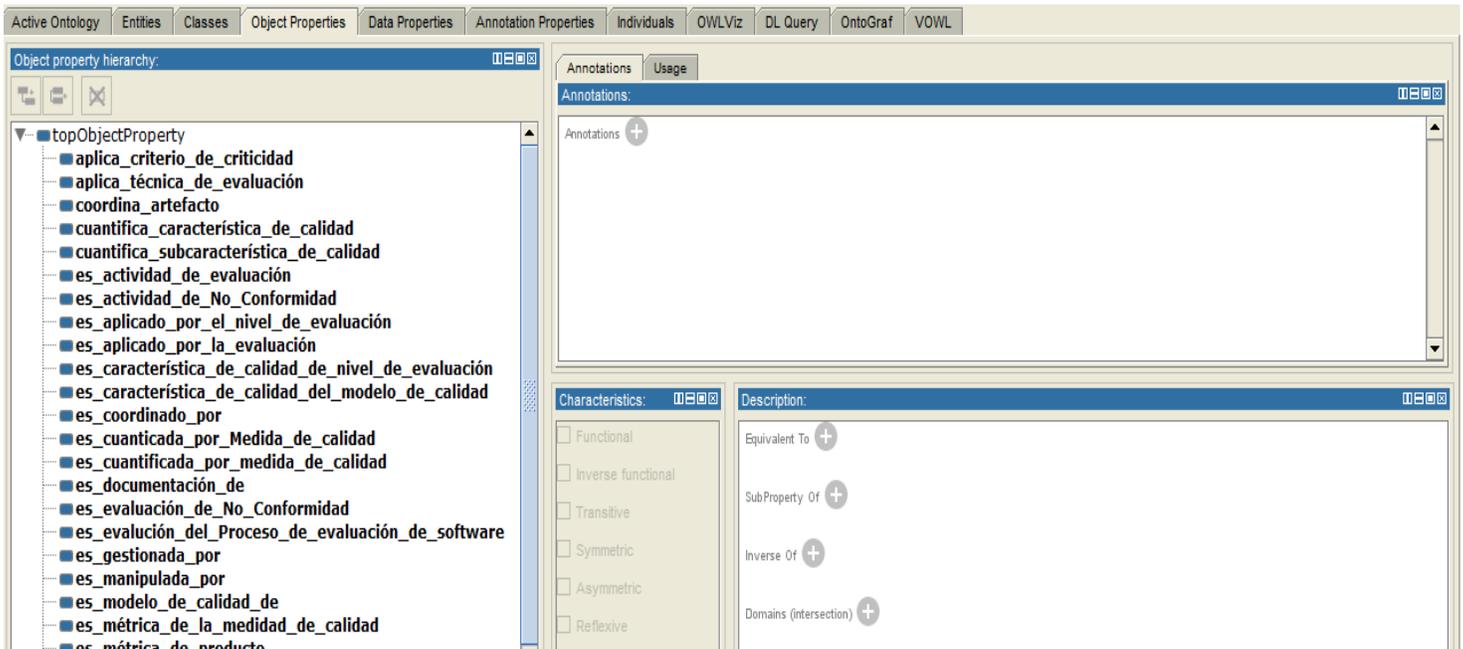


Figura. 6: Propiedades de las clases creadas en Protégé. Fuente: Elaboración propia.

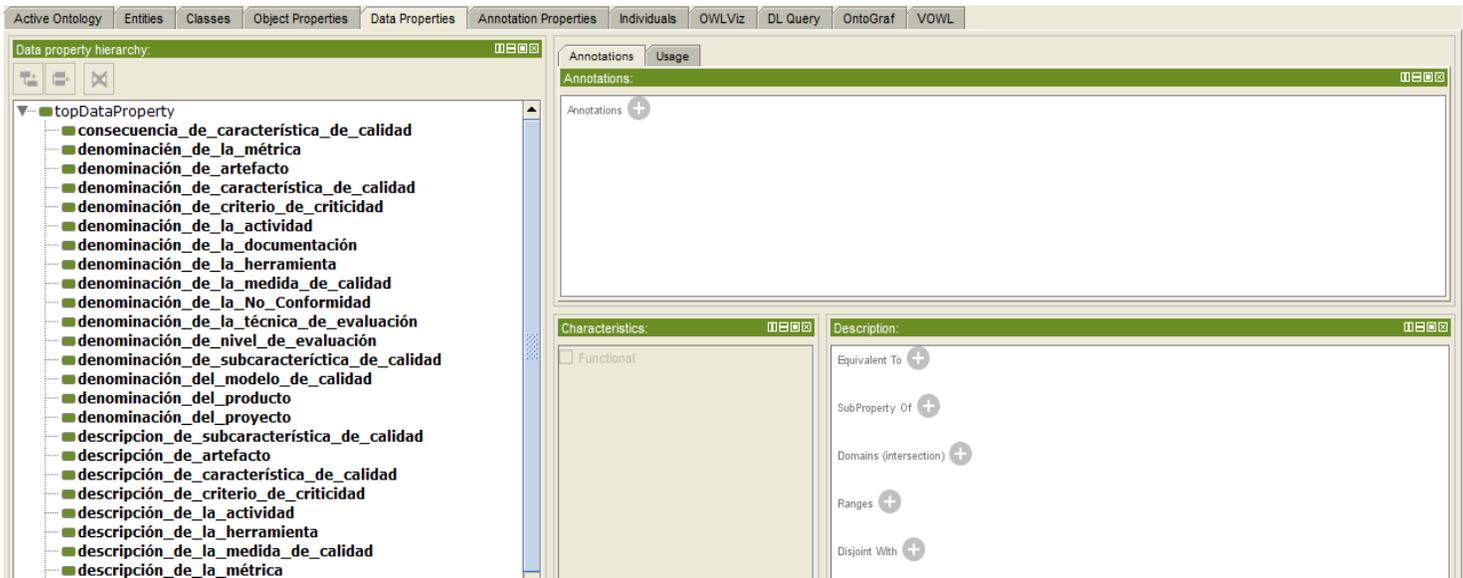


Figura. 7: Creación del tipo de datos de las propiedades en Protégé. Fuente: Elaboración propia.

Creación de las reglas

En la definición de las reglas se utilizaron las relaciones y las clases existentes en la ontología, cuantificándolas de forma universal y existencial. En la Figura. 8 se puede observar las reglas descritas para la clase Coordinador de la evaluación.

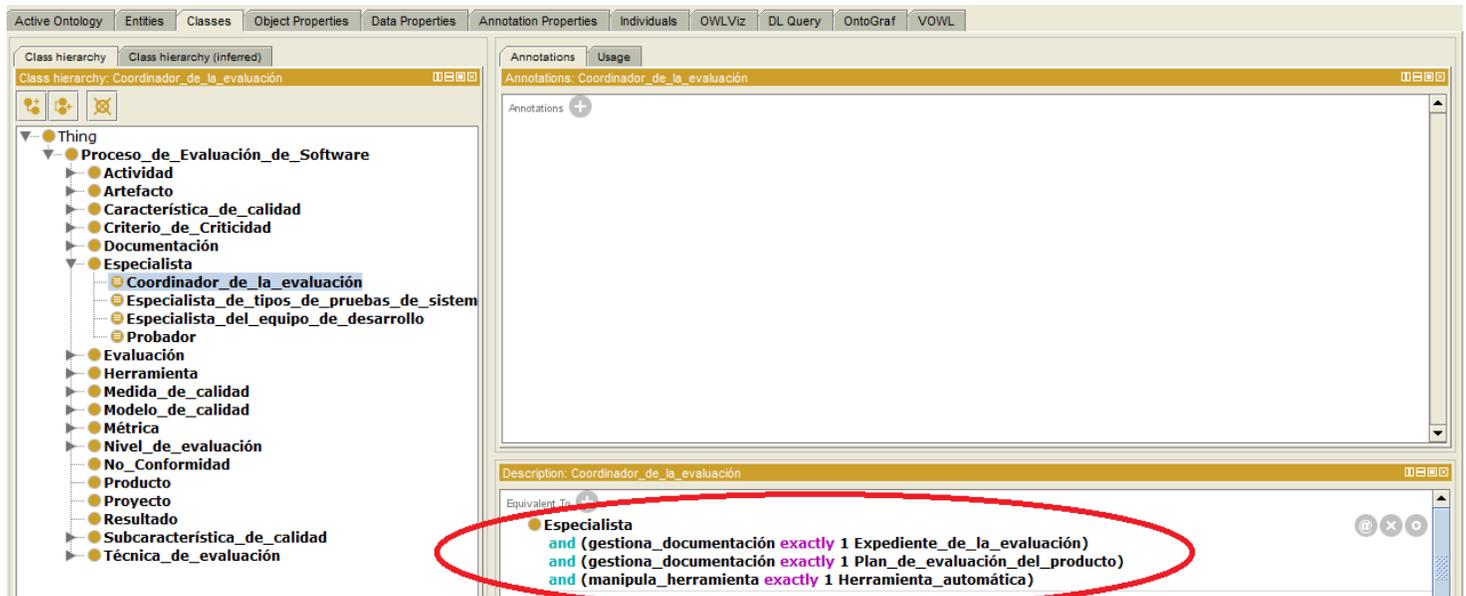


Figura. 8: Creación de las reglas. Fuente: Elaboración propia.

Creación de las instancias

Las instancias representan el conocimiento de la ontología, se usan para representar determinados objetos de un concepto y mediante estas se efectúa gran parte del proceso de razonamiento, además de que muestran en la práctica, la funcionalidad del sistema. Ejemplo de ellas se muestran en la siguiente figura:

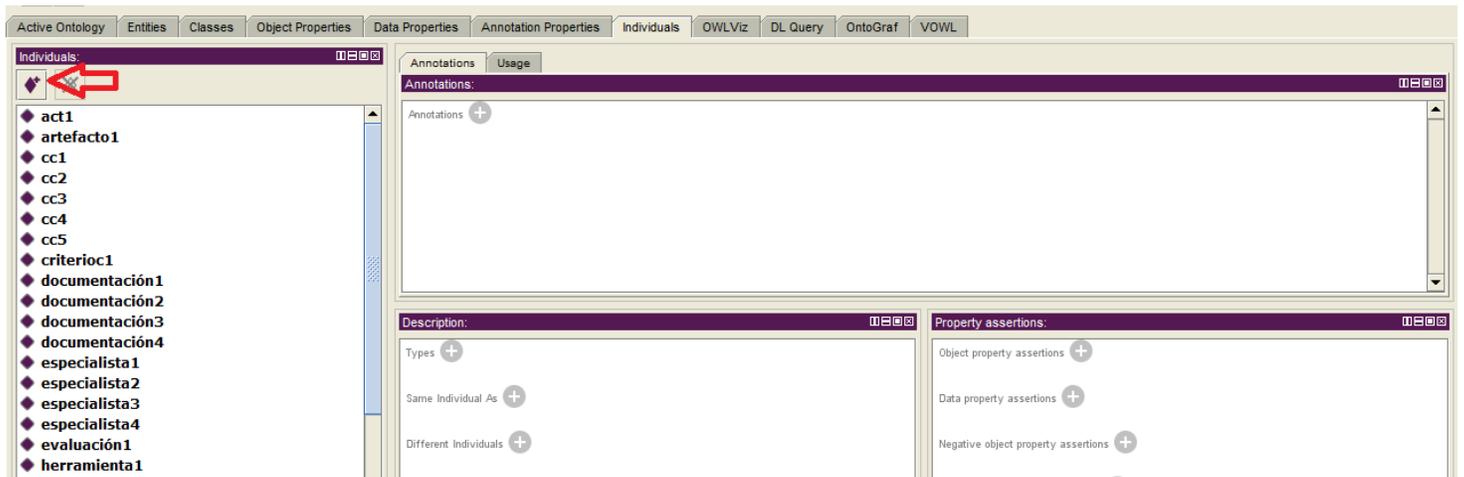


Figura. 9: Creación de las instancias con Protégé. Fuente: Elaboración propia.

OWL Viz: es un plugin para Protégé que permitió visualizar con grafos los conceptos y las relaciones que tienen creadas. En la Figura. 10 se muestra la gráfica de la ontología Proceso de Evaluación de Software.

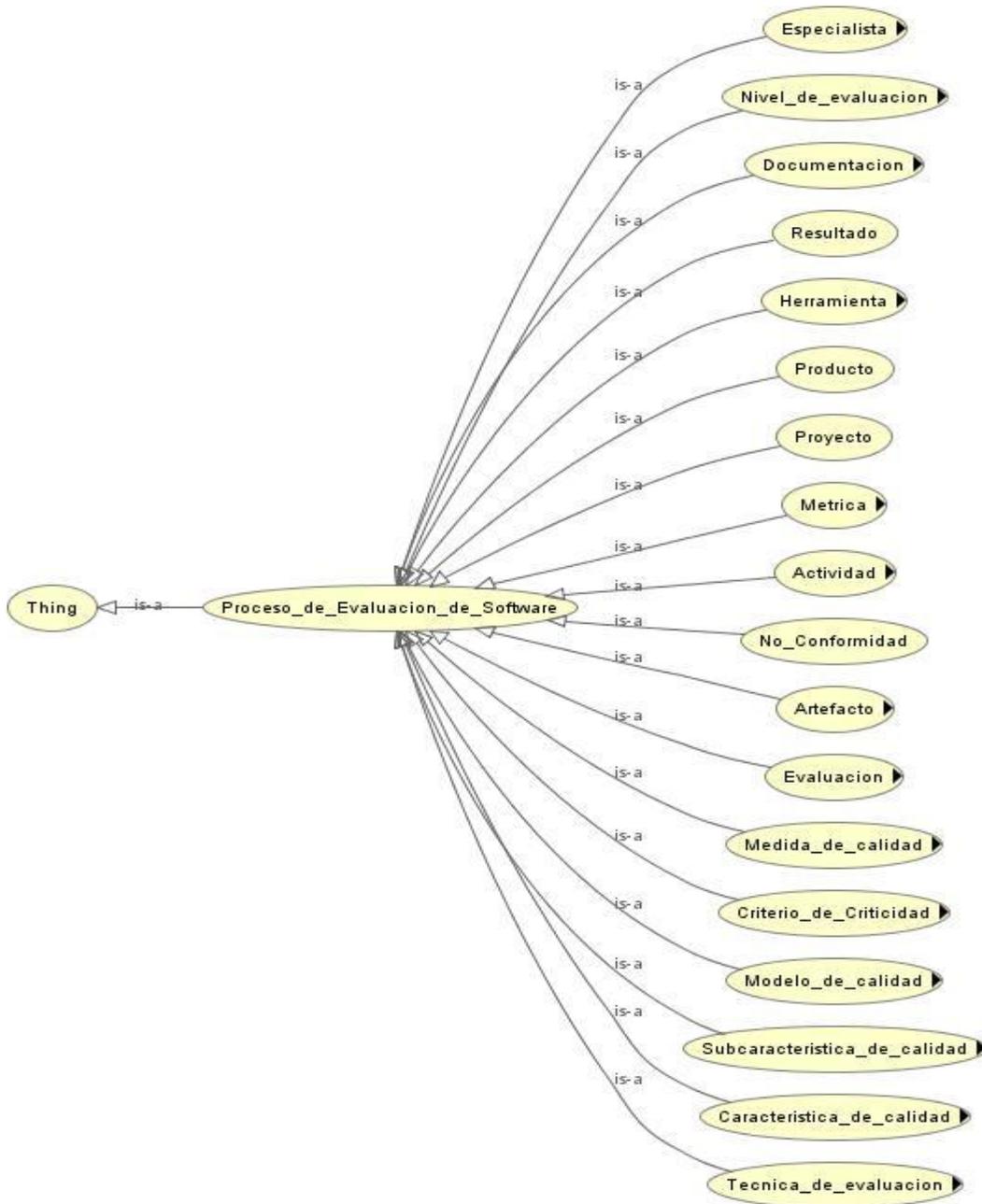


Figura 10: Pestaña OWL Viz del Protégé. Fuente: Elaboración propia.

2.2.4 Implementación

Esta actividad consistió en codificar el modelo generado en un lenguaje ontológico; para esto se seleccionó la herramienta de edición Protégé, versión 4.3, a través de la cual, al definir dicha ontología, se genera la codificación en el lenguaje ontológico OWL. El Lenguaje de Ontologías Web (OWL) provee un lenguaje para definir ontologías. OWL trata de facilitar un lenguaje para ser usado con el fin de describir las clases, la relación entre las clases y las propiedades de las clases. Este lenguaje se construye sobre RDF, quien ofrece la base apropiada para desarrollar ontologías. Las ontologías basadas en RDF podrán ser distribuidas en numerosos sistemas y serán compatibles con otros estándares web. (Lugones, 2017).

A continuación, se muestran algunos fragmentos que describen la manera en que se codifican los datos de la propuesta ontológica.

La declaración de las clases se hace de la siguiente manera, donde se declara la clase Actividad (Ver tabla 7):

Tabla 7: Declaración de las clases.

```
<Declaration>
  <Class IRI="#Actividad"/>
</Declaration>
```

La declaración de una subclase de otra clase, se realiza de la siguiente manera, mostrando que la clase, Evaluación dinámica es subclase de la clase Evaluación (Ver tabla 8):

Tabla 8: Declaración de una subclase de otra clase.

```
<SubClassOf>
  <Class IRI="#Evaluación_dinámica"/>
  <Class IRI="#Evaluación"/>
</SubClassOf>
```

La declaración de las relaciones se realiza de la siguiente manera, donde se describe la relación que existe en la propuesta ontológica, tiene_resultado (Ver tabla 9):

Tabla 9: Declaración de las relaciones.

```
<Declaration>
  <ObjectProperty IRI="# tiene_resultado "/>
</Declaration>
```

En este otro fragmento de código se puede observar la creación de los individuos de las clases, en este caso la creación del individuo act1, el cual es un individuo de la clase Actividad (Ver tabla 10).

Tabla 10: Creación de los individuos de las clases.

```
<Declaration>
  <NamedIndividual IRI=" #act1 "/>
</Declaration>
```

SPARQL

Una de las ventajas que tiene el editor Protégé es que permite realizar ontologías con extensión OWL y RDF, debido a su entorno gráfico basado en plugins. Como ejemplo de estos plugins tenemos el lenguaje SPARQL. A continuación se muestra como se codificaron los datos para dar respuesta a una pregunta de competencia realizando una consulta SPARQL (Ver tabla 11).

Pregunta de competencia: ¿Qué especialista manipula la herramienta1?

Tabla 11: Consulta SPARQL.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX base: <http://www.semanticweb.org/carrion/ontologies/2017/10/untitled-ontology-16#>
SELECT?subject?object
  WHERE { ?subject base: manipula_herramienta base:herramienta1}
```

2.2.5 Mantenimiento

El mantenimiento de software se define como “cualquier modificación de un producto de software, después de su entrega, para corregir errores, mejorar el rendimiento u otros atributos, o a la acción de adaptar el producto a un entorno que cambia” (Montoya, 2013). Gran parte de los sistemas desarrollados, inevitablemente sufren cambios para su duración y utilidad. Algunas partes del mismo tienen que modificarse para corregir errores

detectados en su funcionamiento, adaptarlo a una nueva plataforma, mejorar su rendimiento, entre otras características no funcionales (Alfonzo, 2012).

Esta actividad fue empleada en todo el proceso de diseño y desarrollo de la ontología, donde se fueron agregando nuevas clases, atributos, relaciones e instancias. Además en esta ontología se utilizaron anotaciones para definir el significado de los términos analizados y cuenta con una documentación que apoya el proceso de actualización.

2.3 Conclusiones del capítulo

A partir del diseño y desarrollo de la ontología Proceso de Evaluación de Software se arribaron a las siguientes conclusiones:

- ✚ Mediante el estudio de la NC 25040 se extrajeron los conceptos de mayor énfasis en el dominio del proceso de evaluación de software donde se obtuvo una taxonomía genérica.
- ✚ Se pudo caracterizar el proceso de evaluación de producto de software determinando los principales conceptos y sus relaciones.
- ✚ Aplicando la metodología Methontology y utilizando el editor Protégé se diseñó y desarrolló la ontología para representar el conocimiento generado en apoyo al proceso de evaluación de software definido en la Dirección de la Calidad.

Capítulo 3 Validación y prueba de la ontología

Introducción

En este capítulo se valida la ontología propuesta, basado en un modelo que consta de cuatro fases, comprobando el alcance y objetivo de la ontología desarrollada. La última fase muestra cómo la herramienta Protégé responde a las preguntas de competencias antes recogidas, validando la capacidad de inferencia de la ontología.

3.1 Validación de la ontología

Para las ontologías es importante comprobar según la metodología propuesta (Rodríguez, 2017):

- Sus condiciones y propiedades como sistema formal.
- Su diseño estructural.
- Que cumpla con los requerimientos para los cuales fue creada.

El proceso de validación de las ontologías no es un hecho puntual dado del ciclo de vida. Tiene lugar a lo largo de todo el ciclo de vida, realizándose comprobaciones de las propiedades lógico-formales por medio de razonadores que garantizan aislar los errores en contextos más reducidos (Proenza, y otros, 2012).

Las propiedades lógico-formales de una ontología son críticas con el fin de lograr un grupo de posibilidades inferenciales tales como:

- Determinar si es posible que una clase tenga instancias.
- Inferir cuáles son las clases a las que directamente pertenece una instancia. Si además se utiliza la jerarquía inferida mediante la clasificación anterior, es posible obtener todas las clases a las que indirectamente pertenece una instancia dentro de la ontología.
- Clasificar la ontología a partir de las relaciones de subclase entre todos los conceptos declarados explícitamente para construir la jerarquía de clases.
- Chequear la consistencia de una ontología para asegurar que no contiene hechos contradictorios.

Para que se cumplan estas posibilidades es necesario que la ontología tenga calidad y que en cada fase del ciclo de vida de desarrollo, se evalúen los resultados parciales.

La calidad de una ontología se puede valorar examinando un conjunto mínimo de criterios: que la taxonomía de conceptos sea completa, sin redundancia y consistente, que las preguntas de competencias se respondan correctamente y que el vocabulario utilizado para representar el conocimiento tenga cobertura suficiente del corpus³.

Según Ramos y otros (Ramos, et al., 2009) el modelo para validar ontologías de dominio consta de cuatro fases:

- Fase 1: Uso correcto del lenguaje.
- Fase 2: Exactitud de la estructura taxonómica.
- Fase 3: Validez del vocabulario.
- Fase 4: Adecuación a requerimientos.

3.1.1 Fase 1. Uso correcto del lenguaje

Para evaluar el uso correcto del lenguaje se comprobó que este cumpla con los estándares para el desarrollo ontológico. En esta investigación se utilizó el lenguaje OWL el cual cumple con dichos estándares. Este es un lenguaje consistente y completo, permitiendo la aplicación de métodos de razonamiento sobre la ontología de manera satisfactoria.

De la misma manera, es importante resaltar que la escritura está libre de errores y de inconsistencias sintácticas. Además, se verificó que las palabras utilizadas en la ontología estuvieran bien escritas según la Real Academia Española y NC ISO 25000. Se comprobó que cada palabra fuera lo más justo posible a su rol dentro del proceso de evaluación de productos de software, logrando obtener una ortografía de óptima calidad.

3.1.2 Fase 2. Exactitud de la estructura taxonómica

La evaluación taxonómica considera el chequeo de inconsistencias, completitud y redundancia de los términos de la taxonomía (Ramos, y otros, 2009).

³ Conjunto más extenso y ordenado posible de datos o textos científicos, literarios, que pueden servir de base a una investigación.

En esta fase al realizar una comprobación manual de cada concepto y sus relaciones, no se encontraron inconsistencias tales como: una misma clase es definida como subclase y superclase al mismo tiempo en diferentes niveles de la taxonomía, existencia de más de un concepto principal, falta de conocimiento disjunto, poca especificación o delimitación de las propiedades que provoca un pobre razonamiento y existencia de conceptos repetidos.

3.1.3 Fase 3. Validez del vocabulario

Para validar el vocabulario se chequea que los términos codificados en la ontología existan y sean significativos en otras fuentes de conocimiento independientes, como por ejemplo, el conocimiento contenido en el corpus del dominio (Ramos, y otros, 2009).

Las actividades realizadas en esta fase fueron las siguientes: se analizó el corpus del dominio identificando los términos significativos a partir de los documentos; por otra parte, se evaluó el vocabulario considerando medidas de calidad de resultados usadas en escenarios de recuperación de información tales como la precisión y la exhaustividad.

Calcular la precisión nos brinda el porcentaje de los términos de la ontología que aparecen en el corpus con relación a la cantidad total de términos de la ontología, utilizando la siguiente expresión:

$$\text{Precisión} = \frac{CO_C}{COnto} \quad (1)$$

Donde:

- CO_C : Cantidad de términos que se solapan entre la ontología y el corpus
- $COnto$: Cantidad total de términos de la ontología.

En este caso la cantidad de términos que se solapan entre la ontología y el corpus es 106, al igual que la cantidad total de términos de la ontología.

$$\text{Precisión} = \frac{106}{106} = 1$$

Esto nos demuestra que el 100% de los términos existentes en la ontología se encuentran en el corpus del dominio.

Calcular la Exhaustividad nos brinda el porcentaje de términos del corpus que aparecen en la ontología con relación al total de términos en el corpus, utilizando la siguiente expresión:

$$Exhaustividad = \frac{CO_C}{CCorp} \quad (2)$$

- *CCorp*: Cantidad total de términos del corpus, el cual su valor es 106.

$$Exhaustividad = \frac{106}{106} = 1$$

Esto nos demuestra que el 100% de los términos del corpus del dominio aparecen en la ontología.

3.1.4 Fase 4. Adecuación a requerimientos

En esta fase para verificar que la ontología puede ser utilizada según lo previsto se utilizaron las preguntas de competencia. Para esto se aplicaron casos de prueba con la siguiente estructura: pregunta de competencia que aborda, escenario de prueba, resultado esperado y resultado obtenido. En correspondencia con las preguntas de competencia se aplicaron los casos de prueba.

Caso de prueba 1

Pregunta de competencia: ¿Qué tipo de evaluación tiene el producto1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra en la siguiente tabla (Ver tabla 12).

Tabla 12: Ejemplo de instancia de la clase Evaluación.

Clases	Instancias	Propiedades	Valor de las Propiedades
Evaluación	evaluación1	tiene productoE	producto1

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen la evaluación1 debe clasificarse Evaluación dinámica (Ver figura 11).

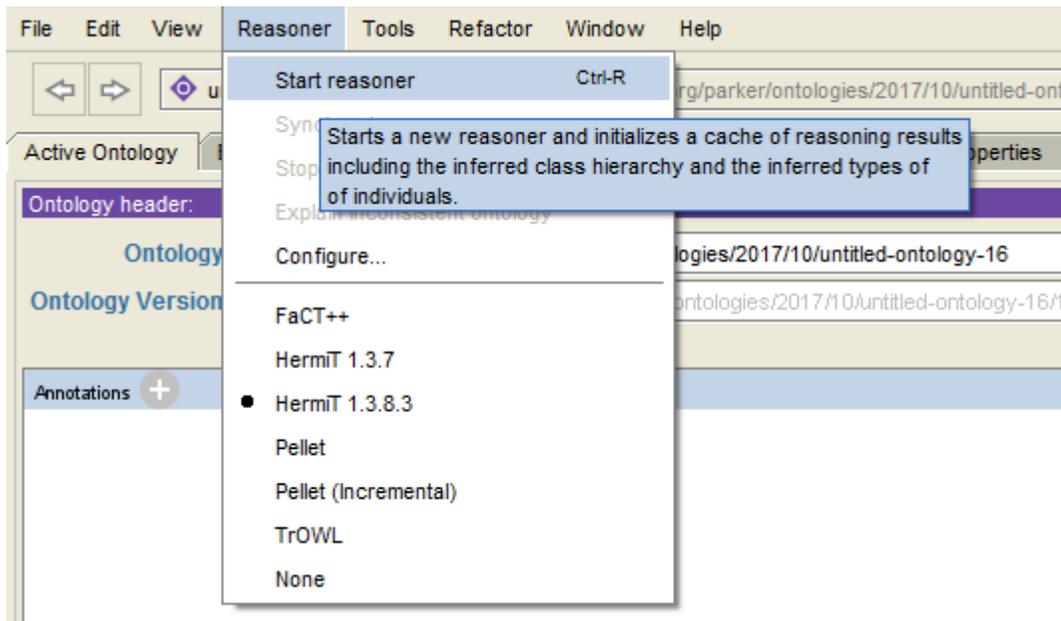


Figura 11 : Aplicación del razonador. Fuente: Elaboración propia.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido (Ver figura 12).

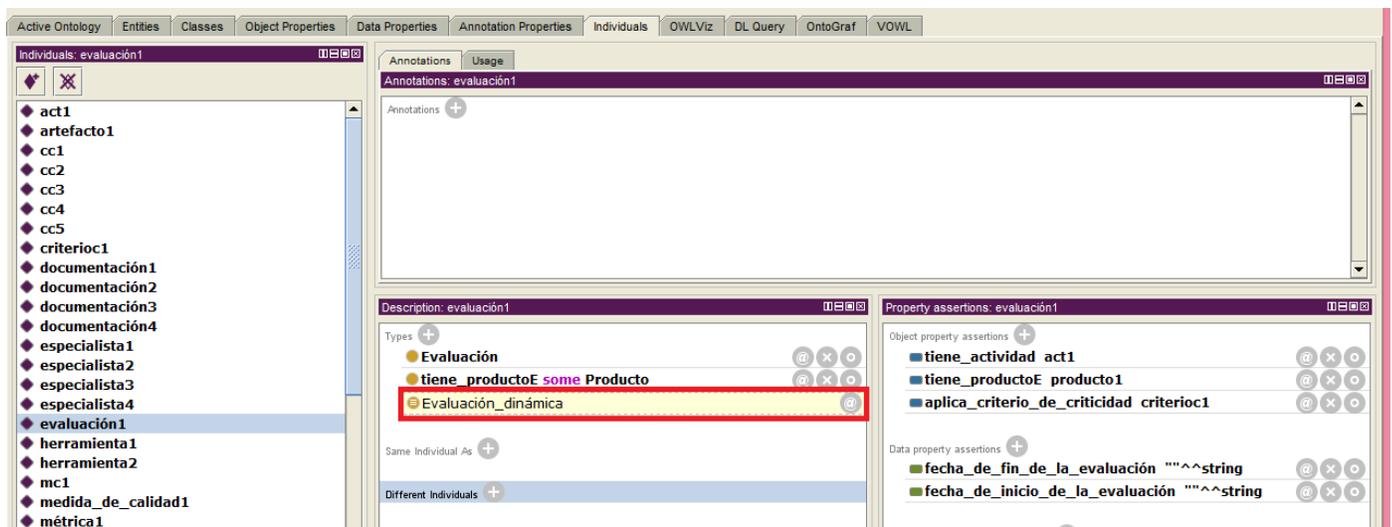


Figura 12: Resultado de la evaluación1. Fuente: Elaboración propia.

De la misma manera, se aplicarán los casos de pruebas restantes (Ver anexo 2).

Durante el desarrollo de la investigación, el profesor que está al frente de la asignatura PID2 en la UCI Yoan Antonio López Rodríguez se mantuvo verificando la realización de la misma hasta su culminación. Avalando de esta manera la aplicación de la propuesta ontológica en el dominio que la refiere.

3.2 Conclusiones del capítulo

Luego del desarrollo del capítulo se arribaron a las siguientes conclusiones:

- ✚ Se validó la propuesta ontológica en función de la organización y recuperación de la información del proceso de evaluación de productos de software, demostrando que se utilizó el lenguaje correcto y que la estructura taxonómica no contiene inconsistencias ni redundancias.
- ✚ A través del razonador y con el uso de las preguntas de competencia, se consultó y encuestó la ontología de forma satisfactoria permitiendo validar su alcance, capacidad de inferencia y recuperación de conocimiento.

Conclusiones generales

A partir del desarrollo de la presente investigación se arribaron a las siguientes conclusiones:

- ✚ El estudio de las las soluciones existentes demostró que, aunque se encontraron algunas propuestas de soluciones relacionadas con el área de la calidad, no cubrían todo el conocimiento generado en el proceso de evaluación de productos de software. Por lo que fue necesario desarrollar una ontología que apoyara dicho proceso en este departamento.
- ✚ La metodología Methontology resultó de gran utilidad para la construcción de la ontología. La misma permitió estructurar el conocimiento adquirido mediante técnicas, tablas y diagramas que pueden ser entendidos por los expertos del dominio.
- ✚ El estudio de las herramientas y tecnologías para el desarrollo de ontologías permitió la creación de una ontología como herramienta de apoyo para organizar y recuperar el conocimiento generado en la Dirección de la Calidad de Software. Dicha ontología ayudará a la toma de decisiones, haciendo inferencias a partir del conocimiento representado en este dominio.
- ✚ El método propuesto para validar la ontología demostró que, la ontología desarrollada se elaboró correctamente y que se cumplieron los requisitos para los cuales fue creada.
- ✚ La ontología propuesta permitió organizar y recuperar los términos utilizados en el proceso de evaluación de productos de software, así como los significados y las relaciones de los mismos. Además, contribuyó a la unificación de los conceptos de dicha área del conocimiento.

Recomendaciones

- ✚ Integrar la ontología desarrollada con la ontología de apoyo al proceso de pruebas de software en la Dirección de la calidad UCI.
- ✚ Poblar la ontología a partir de los datos existentes en la herramienta Gespro mediante un proceso automatizado.

Referencias bibliográficas

1. Abascal, Anaivys Vázquez. 2017. Método para la evaluación de las características de calidad según la ISO/IEC 25000.
2. Alfonzo, S. I. Mariño and M.V.Godoy. 2012. Propuesta de aplicación de SCRUM para gestionar el proceso de mantenimiento del software: estudio preliminar. Técnica Administrativa.
3. Barité, Mario. 2014. El control de vocabulario en la era digital: revisión conceptual.
4. Broughton, Vanda. 2008. Faceted Classification as the Basis of a Faceted Terminology: Conversion of a classified structure to thesaurus format in the Bliss Bibliographic Classification. 2008. págs. 193-210.
5. Cardoso, Lieny Díaz y Pérez, Raúl Alejandro García. 2013. Aplicación Informática para gestionar ontologías representativas del conocimiento en la web. La Habana.
6. Carrión, J. S. 2011. Repositorios Semánticos de Objetos de Aprendizaje.
7. CASE, Herramienta. 2010. Análisis y diseño de sistemas.
8. Dahlberg, Ingetrau. 2012. Knowledge organization: A new science?.
9. Española, Real Academia. 2017. Diccionario de la lengua española . Edición del Tricentenario. [En línea] 2017. [Citado el: 2 de enero de 2017] <http://dle.rae.es/?id=6nVpk8P|6nXVL1Z>.
10. Gómez, Majela Guzmán. 2008. Sistemas de organización del conocimiento y transdisciplinariedad: un acercamiento desde el enfoque de los niveles integrativos. [En línea] noviembre de 2008. [Citado el: 23 de febrero de 2018] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352008001100007.
11. Gruninger, Michael y Fox, Mark S. 1995. Methodology for the Design and Evaluation of Ontologies. Toronto.
12. Guarino, Nicola. 1998. Formal Ontology and Information Systems. Amsterdam, IOS Press. Padova : págs. 3-15.

13. Guzmán Luna, J. A., López Bonilla, Mauricio, Durley Torres, Ingrid . 2012. Metodologías y métodos para la construcción de ontologías.
14. Hernández, Anisleiby Fernández. 2015. Modelo ontológico de recuperación de información para la toma de decisiones en Gestión de Proyectos. Granada.
15. Hjørland, Birger. 2004. Arguments for philosophical realism in library and information science.
16. Huertas, G.Rodríguez & María José López. 2013. Saaty's analytic hierarchies method for knowledge organization in decision making.
17. Lapuente, María Jesús Lamarca. 2013. Hipertexto, el nuevo concepto de documento en la cultura de la imagen. Madrid.
18. Lenat, Guha. 1990. Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project, Addison-Wesley.
19. López-Huertas, María J. 2008. Some Current Research Questions in the Field of Knowledge Organization.
20. Lugones, Mario Luis García. 2017. Ontología para el proceso de medición y mejora definido en un Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas. La Habana.
21. Manchester, T. U. O. 2016. Updated List of OWL reasoners. Information Management Group at the School of Computer Science, The University of Manchester, United Kingdom.
22. María Aurora Soto, Norma M Barrios. 2009. Gestión del conocimiento. Parte I. Revisión crítica del estado del arte . [En línea] 2009. [Citado el: 26 de octubre de 2017] http://bvs.sld.cu/revistas/aci/vol14_2_06/aci04206.htm.
23. Marín, Yisel Padrón y Álvarez, Leonel Montero. 2011. Desarrollo de una ontología para el proceso de pruebas de software del Grupo de Calidad del Centro FORTES.
24. McGuinness, Natalya F Noy and Deborah L. 2005. Ontology development 101: A guide to creating your first ontology.

25. Montoya, Serna. 2013. Acercamiento ontológico a la gestión del conocimiento en el mantenimiento del software.
26. Morón, Claudia Pio. 2015. Ontología para la gestión del conocimiento y la toma de decisiones en el sistema gestión de proyectos XEDRO GESPRO. La Habana.
27. Navarro, Juan Francisco García. 2012. Analítica Visual aplicada a la Ingeniería de Ontologías. Salamanca.
28. Norma Cubana 25040. 2016. Ingeniería de software y sistemas . Requisitos de la calidad y evaluación de software y Sistemas (SQuaRE). Proceso de evaluación.
29. Paradigm, Visual. 2012. Visual Paradigm. Visual Paradigm. [En línea] 2012. [Citado el: 11 de noviembre de 2017] <http://www.visual-paradigm.com>.
30. Pérez, D, and Dressler, M. 2007. Tecnologías de la información para la gestión del conocimiento.
31. Pérez, Yamilis Fernández, Corona, Carlos Cruz y Verdegay, José Luis. 2016. Método multicriterio basado en Soft Computing para evaluar la calidad de software. Granada.
32. Pérez, Yamilis Fernández, Ronquillo, Dayris Espinosa y Martínez, Aliuska Castañeda. 2017. Diseño de una ontología para la evaluación de productos de software en la UCI. La Habana.
33. Proenza, Rey Segundo Guerrero y Martínez, Andrés García. 2012. Metodología para el diseño y desarrollo de ontologías en el campo de la educación. La Habana. págs. 1-11. Vol. 6.
34. Ramos, Esmeralda, Núñez, Haydemar y Casañas, Roberto. 2009. Esquema para evaluar ontologías.
35. Ricardo, Edgar Rojas, y otros. 2012. Aplicación informática para gestionar repositorios, establecer semejanzas y caracterizar ontologías. La Habana.
36. Rivero Amador, J. López Huerta and M. Pérez Díaz. 2013. La interdisciplinariedad de la ciencia y la organización del conocimiento en los sistemas de gestión de la información curricular. Revista Cubana de Información en ciencias de la Salud.

37. Rodríguez, Yoan Antonio López. 2017. Método para la integración de ontologías en sistemas relacionales para la evaluación de créditos. La Habana.
38. Sanchez, Sonia. 2009. Organización del conocimiento, Sistema de KOS.
39. Santiago, Yanet Riquelme. 2015. Sistema de información basado en ontología para la gestión del conocimiento y la toma de decisiones en el sistema de gestión de proyectos XEDRO GESPRO. La Habana.
40. Tello, Pablo Gabriel. 2016. Evaluación de Calidad de un Producto de Software.
41. Uschold, Mike y King, Martin. 1995. Towards a Methodology for Building Ontologies. IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing.
42. Vanguardia, La. 2015. La Vanguardia. [En línea] 2015. [Citado el: 23 de noviembre de 2017] <http://www.lavanguardia.com>.
43. Vitelli, Iván Flores. 2011. Aplicación de methontology para la construcción de una ontología en el dominio de la Microbiología. Caracas.

Anexos

Anexo 1. Taxonomía de conceptos de la ontología Proceso de Evaluación de Software

Taxonomía 2

Taxonomía para los conceptos Documentación, Medida de calidad, Modelo de calidad, Nivel de Evaluación, y Especialista.

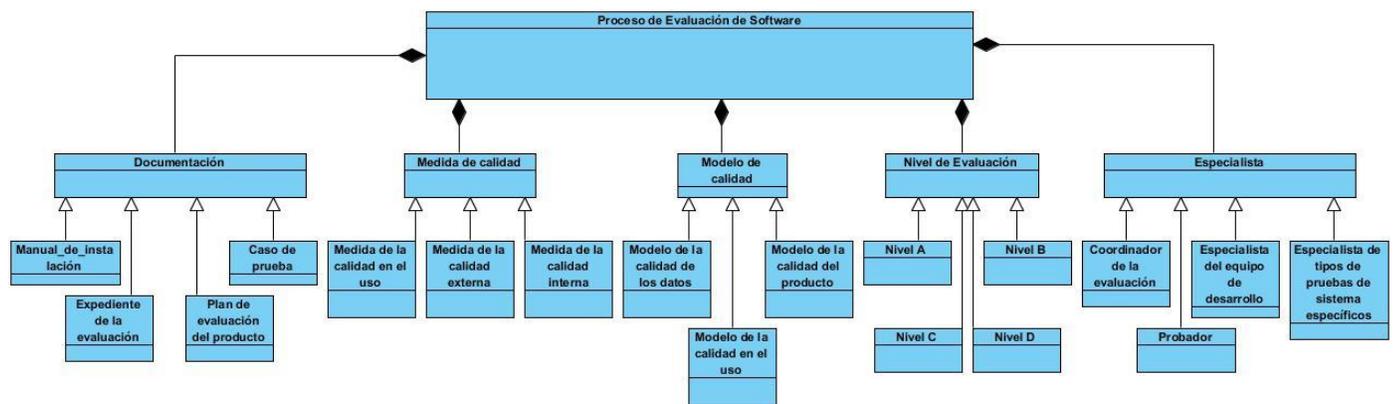


Figura 13: Taxonomía 2. Fuente: Elaboración propia.

Taxonomía 3

Taxonomía para el concepto Sub-característica de calidad.

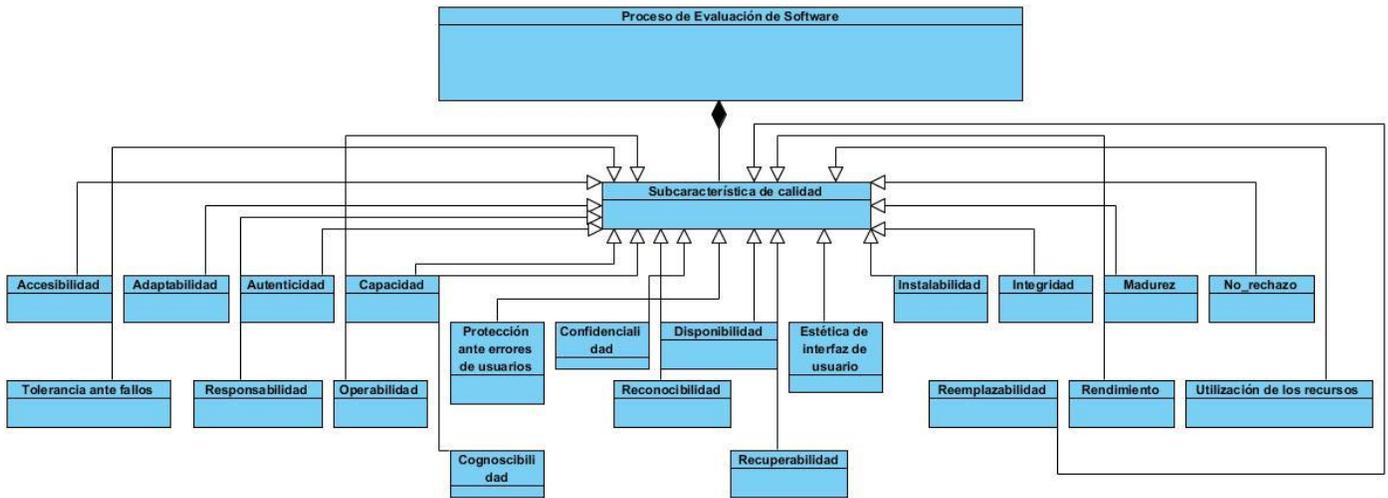


Figura 14: Taxonomía 3. Fuente: Elaboración propia.

Taxonomía 4

Taxonomía para el concepto Técnica de evaluación.

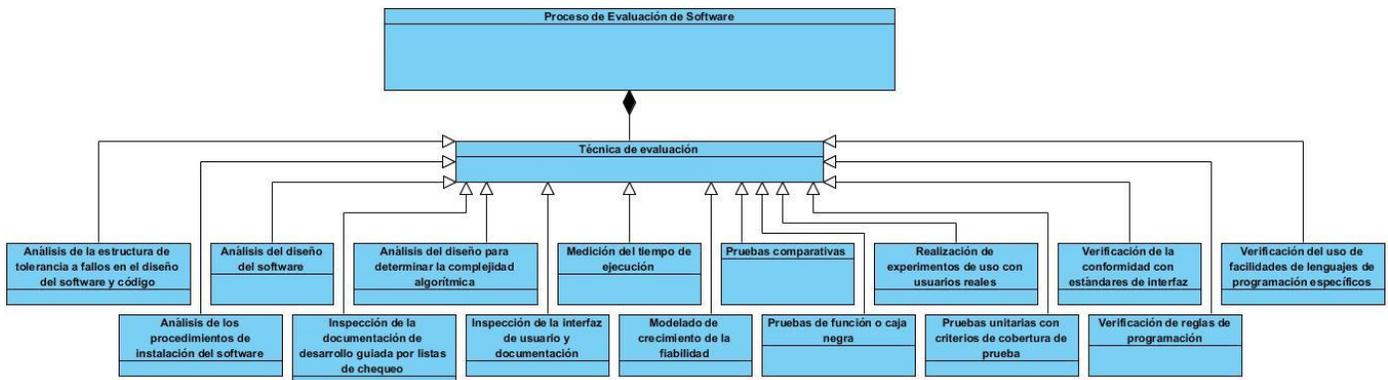


Figura 15: Taxonomía 4. Fuente: Elaboración propia.

Taxonomía 5

Taxonomía para el concepto Actividad.

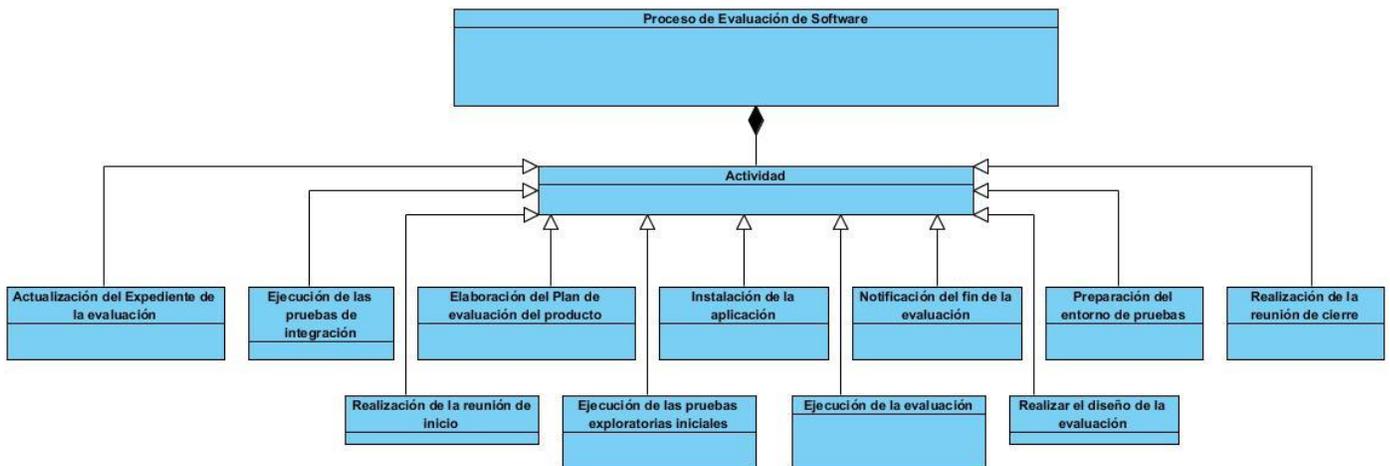


Figura 16: Taxonomía 5. Fuente: Elaboración propia.

Anexo 2. Casos de prueba de la ontología Proceso de Evaluación de Software

Caso de Prueba 2

Pregunta de competencia: ¿Qué especialista manipula la herramienta1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 13).

Tabla 13: Ejemplo de instancia de la clase Especialista.

Clases	Instancias	Propiedades	Valor de las Propiedades
Especialista	especialista1	manipula herramienta	herramienta1

Resultado esperado: Al aplicar un razonador el especialista1 debe clasificarse Probador.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

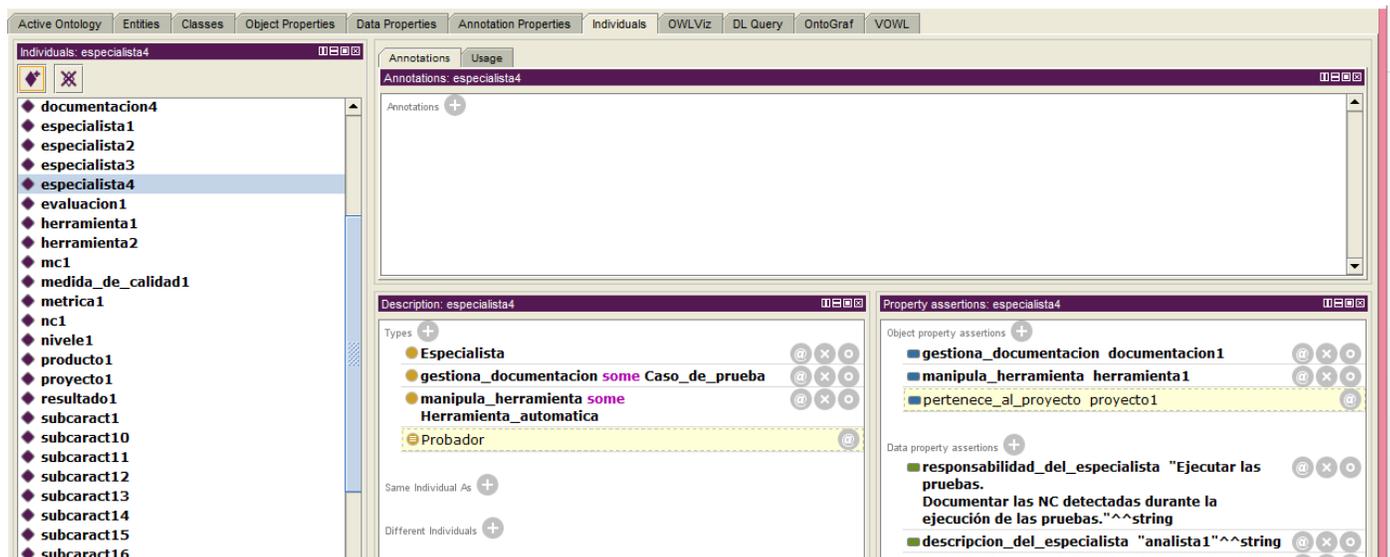


Figura 17: Caso de prueba 2. Fuente: Elaboración propia.

Caso de Prueba 3

Pregunta de competencia: ¿Qué especialista coordina el artefacto Instalador de aplicación?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 14).

Tabla 14: Tabla 14: Ejemplo de instancia de la clase Especialista.

Clases	Instancias	Propiedades	Valor de las Propiedades
Especialista	especialista3	coordina artefacto	artefacto1

Resultado esperado: Al aplicar un razonador el especialista3 debe clasificarse en Especialista del equipo de desarrollo.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

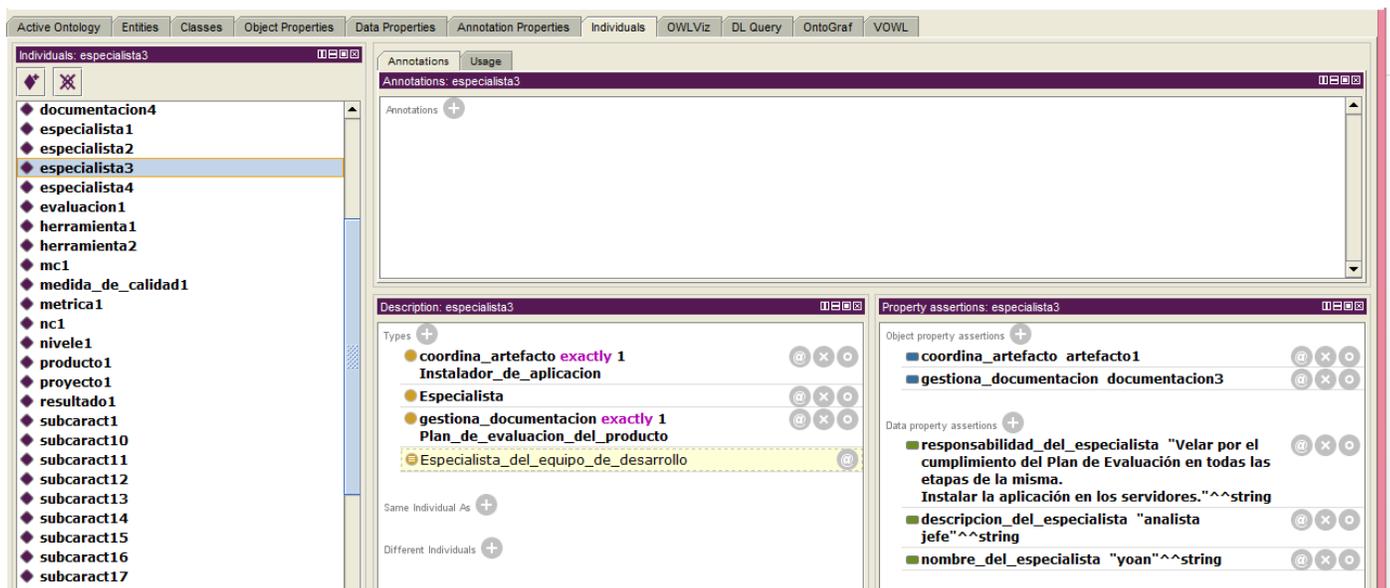


Figura 18: Caso de prueba 3. Fuente: Elaboración propia.

Caso de Prueba 4

Pregunta de competencia: ¿Qué actividad se realiza en la evaluación1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 15).

Tabla 15: Ejemplo de instancia de la clase Actividad.

Clases	Instancias	Propiedades	Valor de las Propiedades
Actividad	actividad1	es actividad de evaluación	evaluación1

Resultado esperado: Al aplicar un razonador la actividad1 debe clasificarse en Actualización del Expediente de la evaluación.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

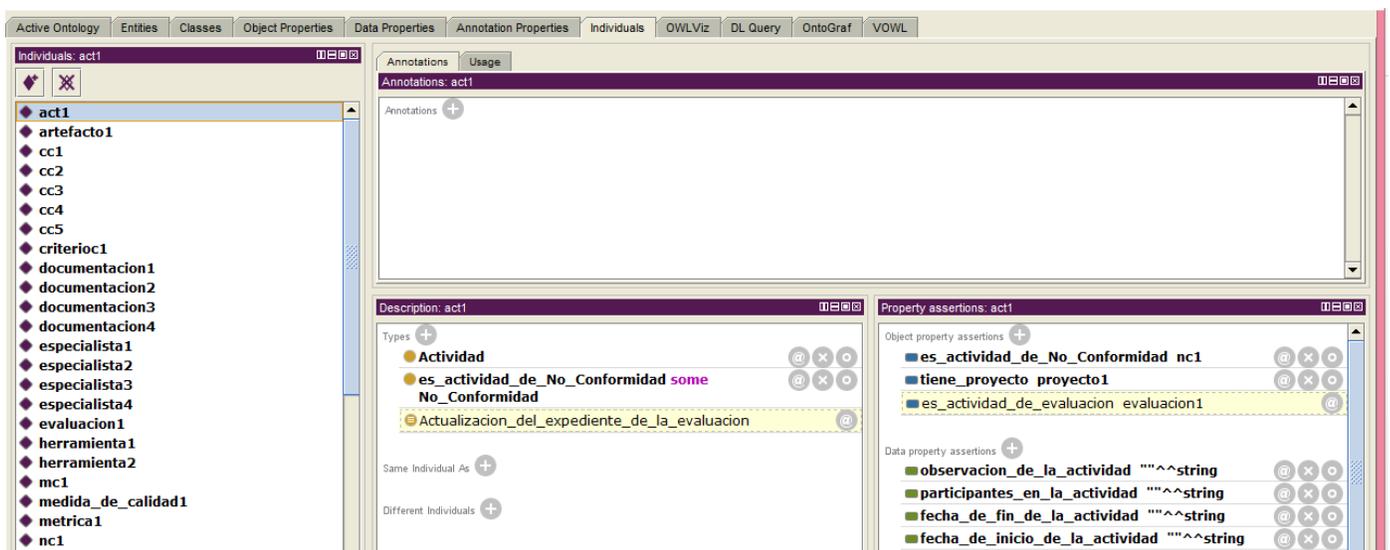


Figura 19: Caso de prueba 4. Fuente: Elaboración propia.

Caso de Prueba 5.

Pregunta de competencia: ¿Qué especialista gestiona la documentación2 y 4?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 16).

Tabla 16: Ejemplo de instancia de la clase Especialista.

Clases	Instancias	Propiedades	Valor de las Propiedades
Especialista	especialista2	gestiona documentación	documentación2 documentación4

Resultado esperado: Al aplicar un razonador el especialista2 debe clasificarse en Especialista de tipos de pruebas de sistema específicos.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

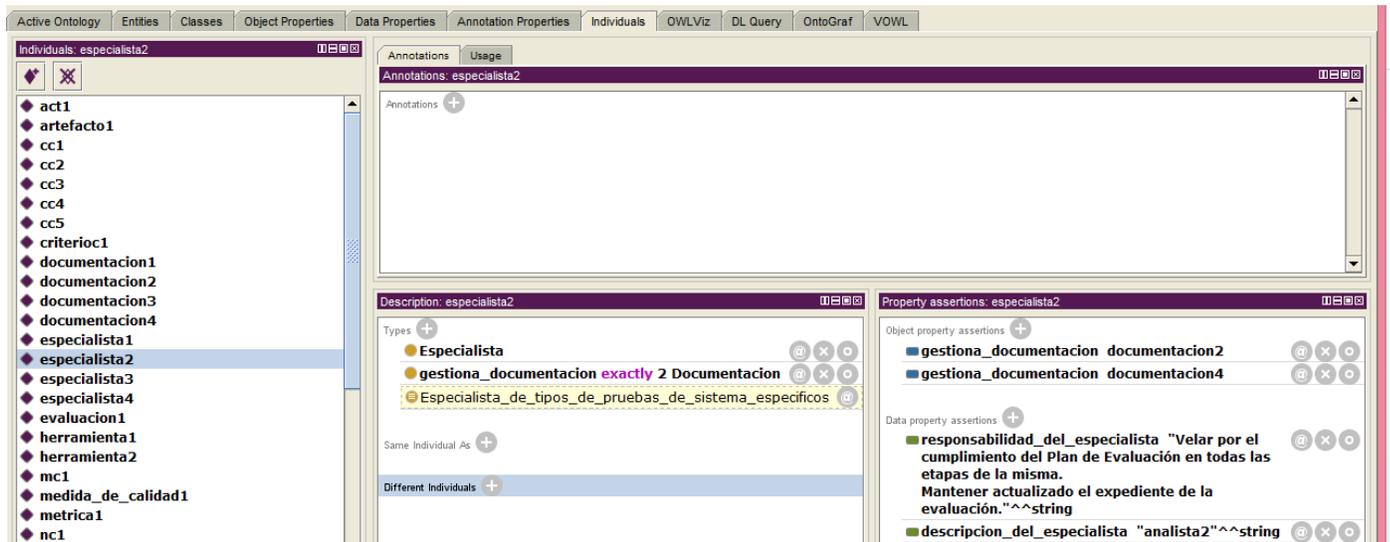


Figura 20: Caso de prueba 5. Fuente: Elaboración propia.

Caso de Prueba 6

Pregunta de competencia: ¿Qué tipo de métrica utiliza el producto1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 17).

Tabla 17: Ejemplo de instancia de la clase Completitud funcional.

Clases	Instancias	Propiedades	Valor de las Propiedades
Completitud funcional	métrica1	es métrica de producto	producto1

Resultado esperado: Al aplicar un razonador la métrica1 debe clasificarse en Cobertura funcional.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

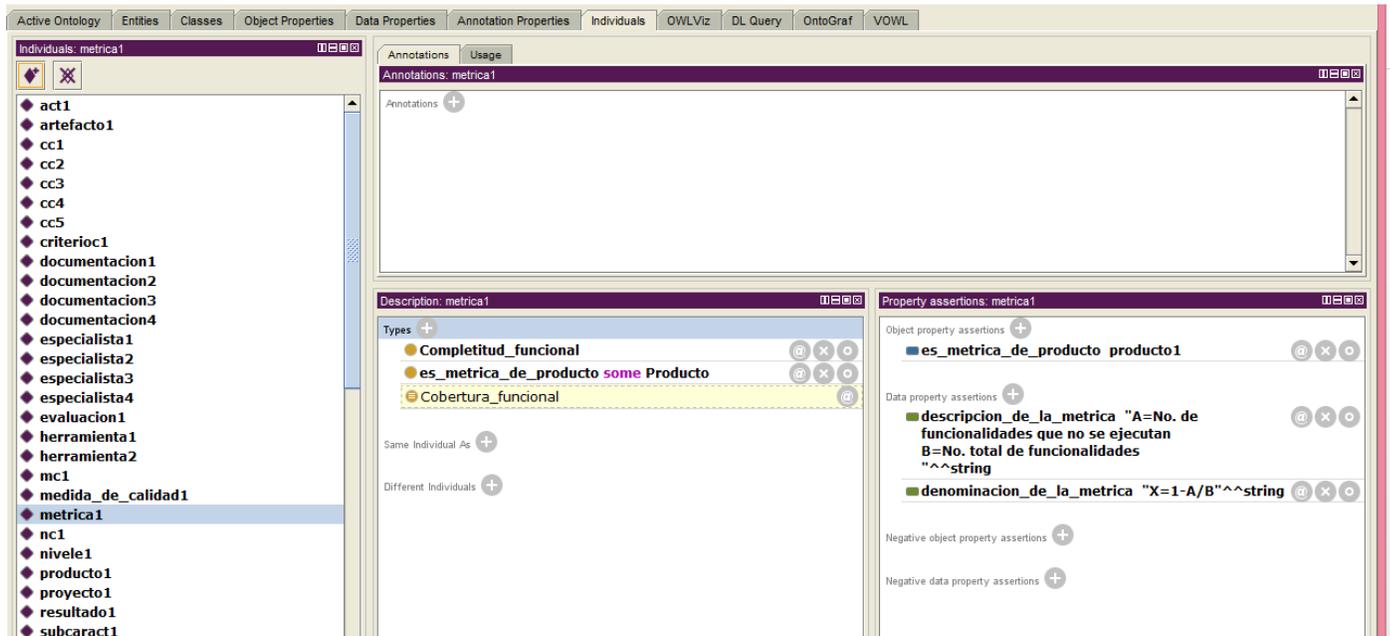


Figura 21: Caso de prueba 6. Fuente: Elaboración propia.

Caso de Prueba 7

Pregunta de competencia: ¿En cuál nivel de evaluación se aplica la técnica de evaluación Verificación del uso de facilidades de lenguajes de programación específicos?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 18).

Tabla 18: Ejemplo de instancia de la clase Nivel de evaluación.

Clases	Instancias	Propiedades	Valor de las Propiedades
Nivel de evaluación	nivel1	aplica técnica de evaluación	técnica1

Resultado esperado: Al aplicar un razonador la técnica1 debe inferir que es aplicado por el nivel de evaluación nivel1.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

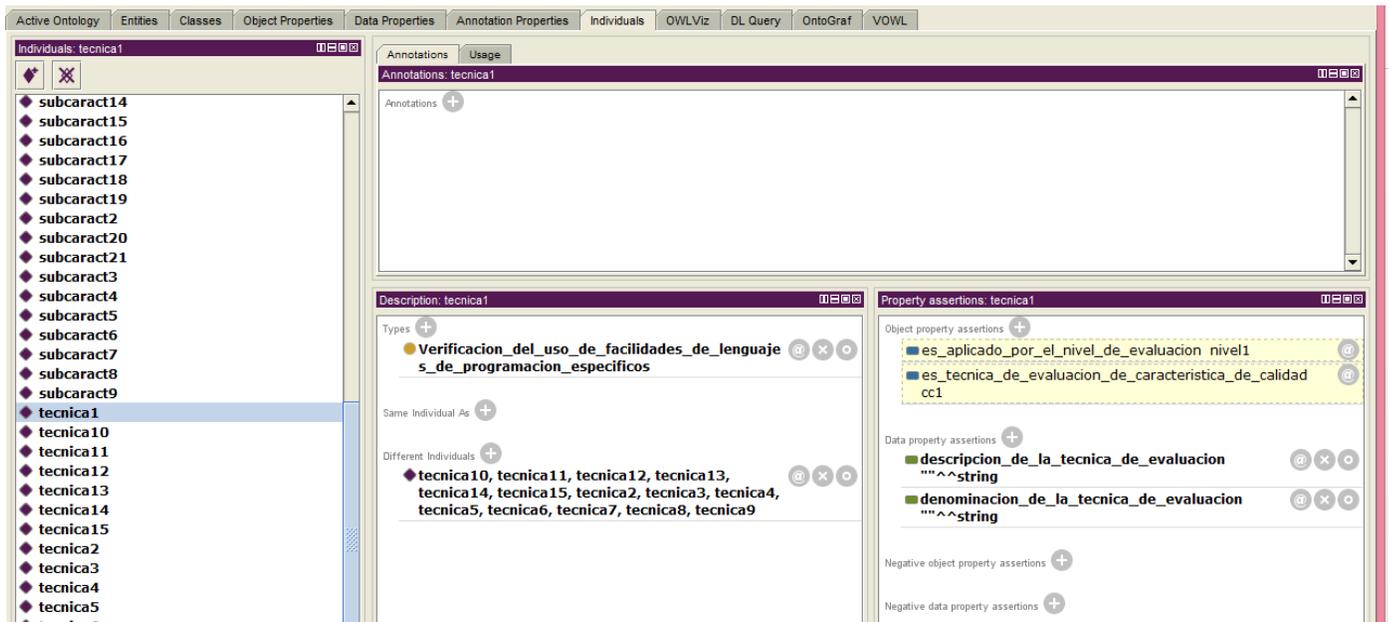


Figura 22: Caso de prueba 7. Fuente: Elaboración propia.

Caso de Prueba 8

Pregunta de competencia: ¿Qué tipo de Actividad tiene la nc1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 19).

Tabla 19: Ejemplo de instancia de la clase Actividad.

Clases	Instancias	Propiedades	Valor de las Propiedades
Actividad	act1	es actividad de No Conformidad	nc1

Resultado esperado: Al aplicar un razonador la nc1 debe inferir que tiene Actividad act1.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

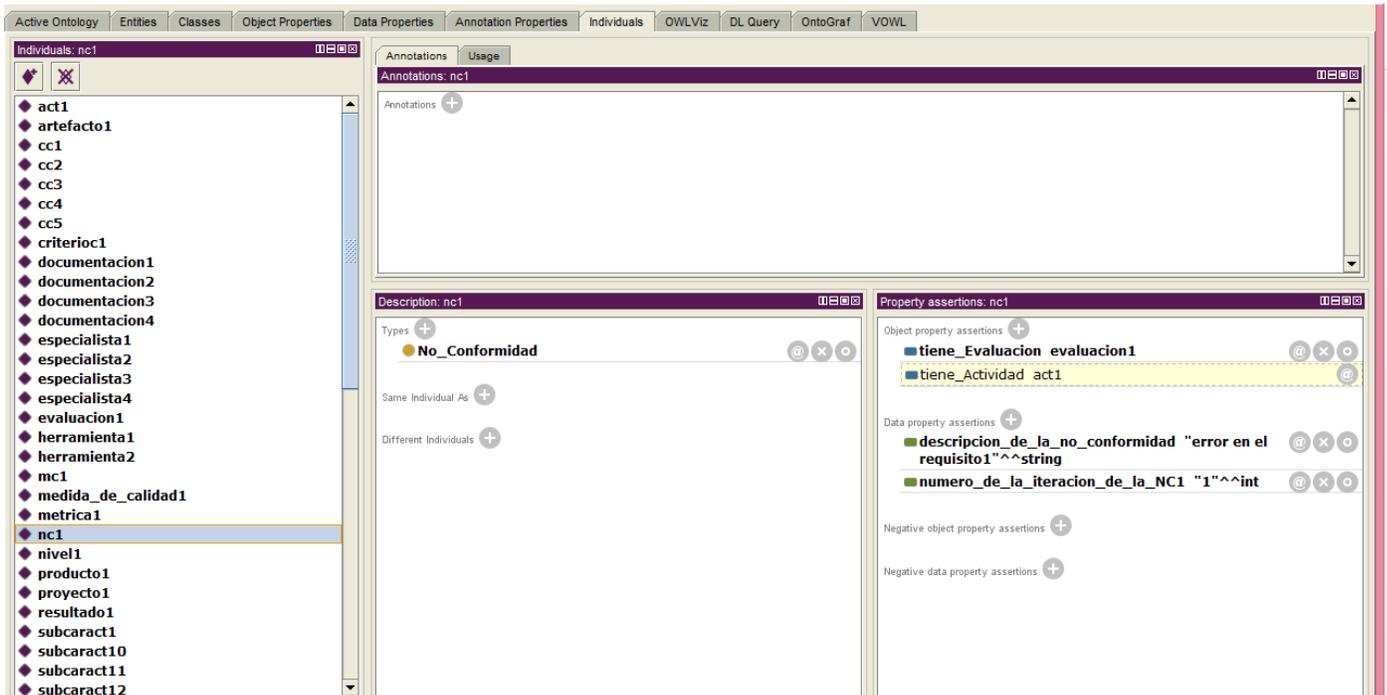


Figura 23: Caso de prueba 8. Fuente: Elaboración propia.

Caso de Prueba 9

Pregunta de competencia: ¿Cuál es el proyecto al que pertenece el producto1?

Escenario: En la herramienta Protégé se crearon las instancias de la ontología como se muestra a continuación (Ver tabla 20).

Tabla 20: Ejemplo de instancia de la clase Proyecto.

Clases	Instancias	Propiedades	Valor de las Propiedades
Proyecto	proyecto1	tiene producto	producto1

Resultado esperado: Al aplicar un razonador producto1 debe inferir que es producto de proyecto proyecto1.

Resultado obtenido: Satisfactorio. La figura siguiente muestra el resultado obtenido.

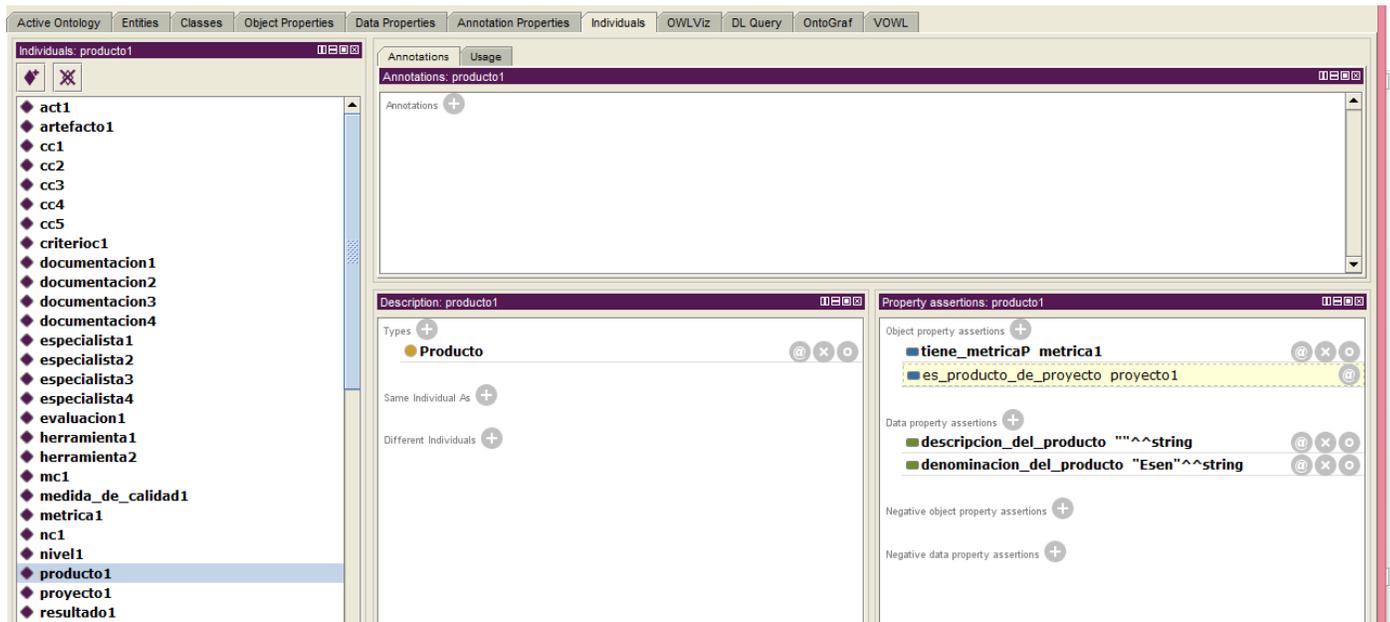


Figura 24: Caso de prueba 9. Fuente: Elaboración propia.

Anexo 3. Diseño de la entrevista

Estimado(a):

Con la entrevista se pretende visualizar la importancia de desarrollar una ontología para el proceso de evaluación de productos de software. Muchas Gracias por su colaboración.

Nota: El objetivo de esta entrevista es solamente investigativa.

- 1) ¿Cuáles son las áreas de proceso relacionadas con la calidad de CMMI que la UCI tiene definida?
- 2) ¿A qué niveles se realizan estas actividades de calidad?
- 3) ¿Dónde se encuentra la documentación de las áreas del proceso mencionadas anteriormente?
- 4) ¿Cuál es el flujo de actividades que se ejecutan al evaluar un producto de software?
- 5) ¿Dónde se almacena la información generada de este proceso?
- 6) ¿Qué técnicas de verificación se utilizan?
- 7) ¿Cuáles son los sistemas informáticos que apoyan a dicho proceso?
- 8) ¿Cuáles son las clasificaciones de la lista de defectos?
- 9) ¿Qué estándares se utilizan para el proceso de evaluación de productos de software?
- 10) ¿Qué herramientas manuales y automáticas se llevan a cabo para la realización del proceso de evaluación de productos de software?
- 11) ¿Otro comentario que usted quisiera realizar?

Anexo 4. Expertos entrevistados

Grupos de expertos al que se le aplicó la técnica de entrevista

- ✚ MSc. Anaivys Vázquez Abascal: Especialista de la empresa XETID. Vicepresidente del subcomité 7 de la ISO en Cuba.
- ✚ MSc. Aymara Marin Díaz: Miembro del subcomité 7 de la ISO en Cuba.
- ✚ MSc. Surima Gé Pérez: Jefa del laboratorio de pruebas en Centro Nacional de Calidad de Software.