



Ontología para representar el proceso de pruebas de liberación



Ontology to Represent the Release Testing Process

Castañeda Martínez, Aliuska; Fernández-Pérez, Yamilis

 Aliuska Castañeda Martínez
acastanedam@uci.cu
UNIVERSIDAD DE LAS CIENCIAS
INFORMÁTICAS, Cuba

 Yamilis Fernández-Pérez
yamilif@uci.cu
UNIVERSIDAD DE LAS CIENCIAS
INFORMÁTICAS, Cuba

Revista Cubana de Transformación Digital
Unión de Informáticos de Cuba, Cuba
ISSN-e: 2708-3411
Periodicidad: Trimestral
vol. 2, núm. Esp.4, 2021
rctd@uic.cu

Recepción: 31 Julio 2021
Aprobación: 03 Septiembre 2021

URL: <http://portal.amelica.org/ameli/jatsRepo/389/3892824001/index.html>

Resumen: En la actualidad, la implementación del software se encuentra en ascenso en la misma medida en que se torna más complejo. Es necesario, por tanto, lograr sistemas cada vez más precisos, fáciles de usar y mantener; eficientes, eficaces, seguros, portables, que cumplan, en fin, las funciones para las cuales fueron creados. Para entregar al cliente un sistema con calidad es necesario transitar por diversas etapas. Dentro de ellas se encuentran las pruebas de liberación, proceso complejo y costoso, dada la cantidad de personas, recursos y actividades involucradas. El objetivo de este artículo es describir el proceso de desarrollo de una ontología que contribuya a aumentar la comprensión del proceso de pruebas de liberación. El principal resultado es una ontología que describe el proceso de prueba de liberación e infiere nuevo conocimiento. En el método empleado para la valoración de la ontología, se miden las condiciones y propiedades como sistema lógico formal, el diseño estructural y el cumplimiento de los requerimientos para los cuales fue creada.

Palabras clave: Ontología, Calidad de software, Pruebas de liberación, Proceso de pruebas.

Abstract: Currently, software implementation is on the rise as it becomes more complex. Therefore, it is necessary to achieve ever more precise systems, easy to use and maintain, efficient, effective, safe, portable, that ultimately fulfill the functions for which they were created. In order to deliver a quality system to the client, it is necessary to go through various stages. Among them are the release tests, a complex and expensive process, given the number of people, resources and activities involved. The objective of this research is to develop an ontology that contributes to increasing the understanding of the release testing process. The main result is an ontology that describes the release testing process and infers new knowledge. In the method used for the ontology evaluation, the conditions and properties are measured as a formal logical system, the structural design and the fulfillment of the requirements for which it was created.

Keywords: Ontology, Software Quality, Release Test, Testing Process Ontology, Software Quality, Release Test, Testing Process.

INTRODUCCIÓN

El uso de aplicaciones informáticas incrementa la calidad de vida, disminuye el esfuerzo físico de la población y mejora la productividad del trabajo. Se estima que la conexión a Internet de millones de dispositivos, proporcionará a los seres humanos una serie de servicios y aplicaciones inteligentes sin precedentes (Fernández-Pérez, 2018). Esta situación crea un escenario de cambio permanente, donde la rápida capacidad de adaptación e innovación son la clave para el éxito de cualquier organización (Vázquez, 2017).

Con el fin de entregar al cliente un *software* con calidad es necesario desarrollar una serie de etapas donde se comprueban las características de calidad del sistema, tales como fiabilidad, eficiencia, portabilidad, compatibilidad, usabilidad, entre otras características. Por tanto, las pruebas constituyen un paso crucial para velar por el correcto desarrollo del sistema.

Las pruebas de *software* son un proceso costoso y complejo debido al gran volumen de documentación generada, la cantidad de personas, recursos, materiales y actividades involucradas (Fernández-Pérez, 2018). Desde el punto de vista de servicio, es importante la satisfacción de los clientes, entendidos no solo como los usuarios finales, sino también los desarrolladores o quienes desean diseñar nuevas versiones, al poder medir la calidad del producto con precisión.

De lo anterior, se deriva la importancia de entender y hacer eficiente el proceso de pruebas. Dentro de los estándares internacionales que guían y orientan los aspectos fundamentales de las pruebas, existen conceptos ambiguos y solapados. Además, es posible encontrar disímiles significados para un mismo término, lo que aumenta la problemática de entendimiento y comprensión del proceso. Existen inconsistencias y conflictos de terminología entre estándares de diferentes organizaciones e incluso en los de una misma organización. Su aplicación

suele centrarse en temas de un gran espectro; por tanto, no siempre están pensadas para ofrecer soluciones específicas, tornándose complejo entender e interpretar los mismos (ISO/IEC/IEEE., 2017). Esta complejidad provoca dificultades en el entendimiento y comprensión total del proceso de pruebas para una adecuada institucionalización. Dentro de los elementos que influyen en la comprensibilidad del proceso se encuentran:

- • La organización secuencial con un elevado número de tareas que generan varios artefactos.
 - La imprecisión y ambigüedad de la documentación que dificultan el aprendizaje y comprensión del proceso de pruebas.

Para abordar la heterogeneidad, falta de consistencia y ambigüedad de la información, en la literatura se reconoce que las ontologías son una solución recomendable. La razón de su amplia aplicación está sustentada en el propósito de alcanzar una comprensión común y compartida de algún dominio particular, donde puedan comunicarse las personas y las computadoras, lo cual permite considerar la automatización de procesos (Luna, J.A.G.B., 2012).

Considerando las notables ventajas de las ontologías, en esta investigación se adoptó un enfoque basado en estas para representar el proceso de pruebas de liberación, el cual contribuye a que la descripción del proceso sea más comprensible, fácil de entender y no presente inconsistencias o ambigüedades. Esta ontología también facilitaría un entendimiento común sobre el proceso y su ejecución y así compartir y analizar todo el conocimiento representado.

MATERIALES Y MÉTODOS

La revisión documental practicada durante la investigación ((Duarte, 2000; Echeverría Perez, 2011; Fernández Hernández, 2015; Ferreira, M., 2006; Lopez Rodriguez, 2017; Lugones, 2017; Souza, 2013)) arrojó un conjunto de ontologías aplicadas al dominio de calidad de *software*.

En Fernández (Fernández Hernández, 2015), la autora propone el desarrollo de un modelo para el diseño y construcción de un sistema de recuperación de información basado en ontologías. En el trabajo presentado por Felipe Alfonso (Felipe Alfonso, 2011) se hace referencia al uso de una ontología como herramienta para la gestión del conocimiento con el objetivo de apoyar la realización de un Modelo Cubano de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI) en la Industria Cubana de Software. Otro de los estudios relacionados con la temática es el presentado por Cristina Duarte (Duarte, 2000), desarrolla una ontología que describe la calidad de *software*. En ella solo aparecen elementos fundamentales de la calidad en general, sin abordar las pruebas de *software*, ni definir el modelo de calidad.

En la solución presentada por Ferreira (Ferreira, M., 2006) se desarrolla una ontología que caracteriza el proceso de medición. Es abarcadora y elimina problemas de homonimia, sinonimia y otras dificultades presentes entre estándares. No abarca el subproceso de prueba de *software*. Investigaciones como las de Echeverría (Echeverría Perez, 2011) y Sousa (Souza, 2013), abarcan las pruebas de *software* y presentan ontologías generalizadoras de este dominio, pero hay conceptos como rol y especialistas a nivel de proyecto que no se incluyen, lo que limita la toma de decisiones. Resulta interesante la propuesta presentada por Lugones (Lugones, 2017),

el trabajo consiste en el desarrollo de una ontología que modela: utilizar la información y el conocimiento generado en el proceso de medición y mejora definido en el MCDAI. Esta propuesta está encaminada al proceso de medición y análisis, pero se centra en los términos y características de esta área.

De manera general, a pesar de que existen varias propuestas de investigaciones relacionadas con el tema de pruebas de *software*, no se acertó una solución que esté estrechamente vinculada con el procedimiento de pruebas de liberación. Los enfoques analizados no abordan todos los términos o las relaciones surgidas en la Dirección de Calidad. Presentan ontologías generalizadoras de este dominio, pero hay conceptos a nivel de proyecto que no se incluyen, lo que limita la toma de decisiones. Existen diferencias entre ellas ya que corresponden a desarrollo e investigaciones efectuadas por separado, en distintos períodos de tiempo. Esta es la razón por la cual se requiere un esfuerzo de integración que proporcione una vista común en función de eliminar las diferencias y conflictos.

Las mismas, como solución a la problemática actual, son muy generales o incompletas, no abordan todos los términos o las relaciones; solo describen un grupo de aspectos del proceso y hay otros ausentes, como por ejemplo las herramientas necesarias para ejecutar una actividad, la experticia de los involucrados, etc. Por ello, se decide integrar y adaptar a las nuevas necesidades las ontologías ya existentes, estas se extienden con nuevos términos y relaciones, lo que conlleva a presentar una propuesta unificada con el objetivo de crear una terminología consistente para las pruebas de liberación. Para diseñar y desarrollar la ontología se utilizó la metodología de Alvarado (Alvarado, 2010), escogida por la simplicidad, claridad y objetividad de sus pasos. Se basa en lecciones aprendidas de metodologías anteriores, siendo considerada una de las metodologías más completas. Se emplea *OWL2* como lenguaje para desarrollar la propuesta, el cual ha sido ampliamente utilizado, debido, en parte, a sus resultados en la web semántica. *OWL* Permite definir clases mediante condiciones sobre sus miembros, combinación booleana de clases o por enumeración de las instancias pertenecientes a la clase.

Se utilizó *Protégé* (Horridge, M., 2009) como herramienta para implementar la ontología, la cual es una herramienta de código abierto de apreciable aceptación para la creación y edición de ontologías, y el razonador *Pellet*, teniendo en cuenta que es también de código abierto y permite validar con mayor profundidad la consistencia de la ontología.

RESULTADOS Y DISCUSIÓN

Desarrollo del modelo ontológico

El modelo ontológico desarrollado permite describir el proceso de prueba de liberación. La construcción de la ontología *OntoPL* se realiza teniendo en cuenta la correspondencia de la definición de las clases con los conceptos del dominio y sus relaciones, así como la correcta especificación de las propiedades y axiomas que permiten realizar inferencias a través de razonadores.

Para guiar el proceso de desarrollo, fue utilizada la metodología de Alvarado, la cual propone cinco actividades fundamentales: determinación de los requerimientos, reutilización de

ontologías, elaboración del modelo conceptual, implementación y evaluación de las ontologías (Alvarado, 2010). A continuación, se explican los resultados alcanzados durante la ejecución de cada una de las actividades desarrolladas.

Actividad 1. Determinación de los requisitos

La ontología *OntoPL* tiene el objetivo de representar el proceso de pruebas de liberación de la Dirección de calidad de la UCI, teniendo en cuenta los conceptos relacionados al dominio, así como las relaciones entre ellos. Para ello se definen las siguientes preguntas:

- • ¿Qué dominio cubrirá la ontología?
 - ¿Para qué se va a emplear la ontología?
 - ¿Qué preguntas debería contestar la ontología?
 - ¿Quién utilizará y mantendrá la ontología?

Para precisar el alcance de la ontología se elaboró el Documento de Especificación de Requisitos de la Ontología (DERO), donde se precisa el propósito como la descripción del proceso de pruebas de liberación y de los requisitos funcionales y no funcionales. Se definen como requisitos no funcionales el lenguaje de implementación de ontologías, como OWL, usando la herramienta *Protégé5.0* y que la ontología debe estar escrita en idioma español. Los requisitos funcionales se detallaron a través de preguntas de competencia como, por ejemplo:

- • ¿El proceso de prueba de liberación cumple con los patrones básicos de flujo de trabajo?
 - ¿Qué actividades se realizan en el proceso?
 - ¿Qué actividades son realizadas por el rol X?
 - ¿Qué artefactos genera la actividad Y?
 - ¿Quién realiza la actividad J?
 - ¿Qué actividad genera el artefacto Z?

Estos requisitos serán utilizados en el desarrollo de la ontología; posteriormente, en la validación, se verificará el cumplimiento de los mismos.

Actividad 2. Reutilización de ontologías

Cuando se crea una nueva ontología es conveniente comprobar si se pueden reutilizar conceptos de otras ya existentes, por ello la metodología de Alvarado establece como segundo paso la reutilización.

Para la creación de *OntoPL* fueron reutilizadas las siguientes ontologías:

- • *OntoCIMMejorada.owl* (Grass, 2018): describe los aspectos fundamentales de un marco de trabajo cuyo fin es mejorar la descripción de procesos de negocio y habilitar su validación semántica. Para la generación de la ontología se aplican definiciones del paradigma de desarrollo de *software* dirigido por modelos. Esta ontología es una versión enriquecida de *OntoCIM*, la cual se diseñó fundamentalmente para representar los procesos de negocio en *OWL* y considerando conceptos

propios del dominio de la gestión empresarial (N. Silega, 2014); además, es una ontología creada en español.

- *PruebadSoftware.owl* (Castañeda Martínez et al., 2018): describe el subproceso de pruebas de software en la UCI, favoreciendo la organización y comunicación del conocimiento generado y contribuyendo a la verificación de la consistencia de las descripciones. Incluye el conocimiento tácito del personal involucrado en las pruebas, así como el conocimiento explícito representado en diferentes fuentes, como el estándar NC ISO / IEC 25000 y los procedimientos definidos en la entidad. Proporciona un vocabulario común y soluciona los problemas de inconsistencias e integridad detectados en el subproceso; sin embargo, no tiene en cuenta términos y relaciones presentes en el proceso de prueba de liberación.

- *OntoEva.owl* (Estradet, 2018): contribuye a la organización y recuperación del conocimiento generado en el proceso de evaluación de productos de *software*. Describe de forma clara los principales conceptos, términos y las relaciones entre los mismos dentro del proceso de evaluación de *software*.

En resumen, de *OntoCIMMejorada* se reutilizan términos para la representación de los procesos de negocio utilizados, tales como: Proceso, Actividad, Evento, Computa, Paso y Elemento de flujo. Por su parte, *PruebadSoftware* se reutiliza por contener conceptos y clases relacionadas con las pruebas de *software* que sirven de base para el desarrollo de *OntoPL* y cumplir con los requisitos descritos en el DERO. Finalmente, de *OntoEva* se reutilizaron los conceptos asociados al subproceso de evaluación de *software*, el cual forma parte del subproceso de prueba de liberación descrito en *OntoPL* (ver Figura 1).

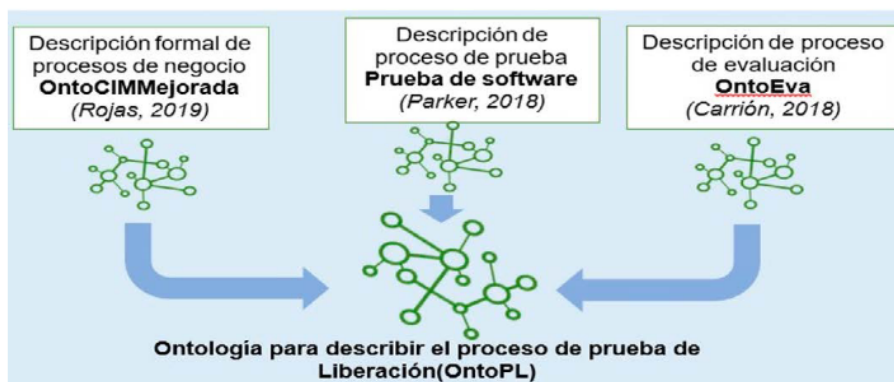


Figura 1. Reutilización de ontologías.

FIGURA 1.
Reutilización de ontologías

sf

Actividad 3. Elaboración del modelo conceptual

Como tercer paso, la metodología define la elaboración del modelo conceptual. A partir de la revisión bibliográfica y del análisis documental se identificaron los conceptos del dominio, según se muestra en la Figura 2.

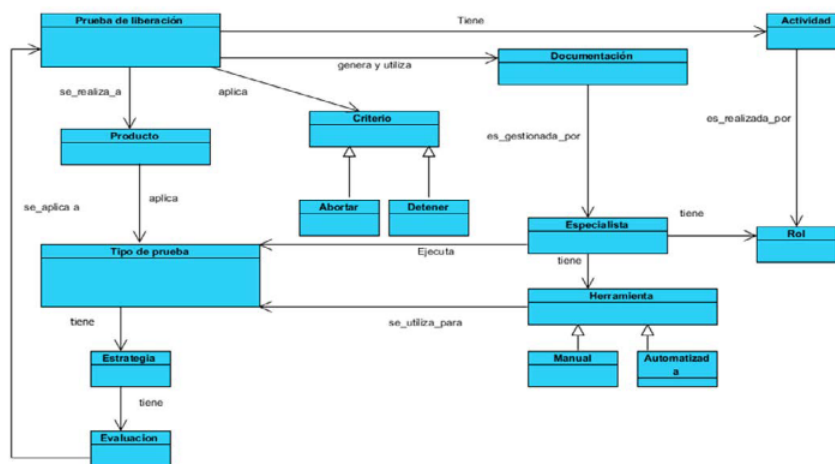


Figura 2. Modelo conceptual.

FIGURA 2.
Modelo conceptual.

sf

Definir las clases y la jerarquía de clases

Para la definición de las clases de la ontología se tuvieron en cuenta tres tipos de conceptos. En primer lugar, los conceptos asociados a la descripción de procesos de negocio; en este caso se reutilizaron los conceptos definidos en *OntoCIMMejorada.owl*; por ejemplo, **Proceso**, **Actividad**, **Evento**, **Compuerta**. A su vez, en esta ontología se incluyen clases para salvar la limitación que tiene *OWL*, el cual no posee constructores nativos de representación de procesos de negocio (N. Silega, 2014). Entre estas clases se encuentra **Paso** y **Elemento-Flujo**. En segundo lugar, los conceptos asociados al proceso de pruebas de liberación. En este caso se reutilizaron los conceptos definidos en *PruebadSoftware.owl*. Algunas de las clases fundamentales son: **Producto**, **Especialista**, **Rol**, **Documentación**, **Herramienta**. En tercer lugar, los conceptos asociados al proceso de evaluación; por ejemplo, **Evaluación**, **TipodePrueba** y **Criterio**. Luego de la identificación de las clases, se establece la jerarquía de las mismas.

Actividad 4. Implementación de la ontología

En esta actividad, se transforma el modelo conceptual construido en las actividades anteriores a un modelo formal o semi-computable, transformando los conceptos y relaciones, atributos e instancias entre ellos a dicho modelo, haciendo uso del editor de ontologías *Protégé*. A continuación, se muestran cada uno de los pasos desarrollados.

Creación de las clases

Las clases son creadas según la jerarquía definida en la taxonomía de conceptos. En la Figura 3, se muestra la definición de las clases de la ontología desarrollada.

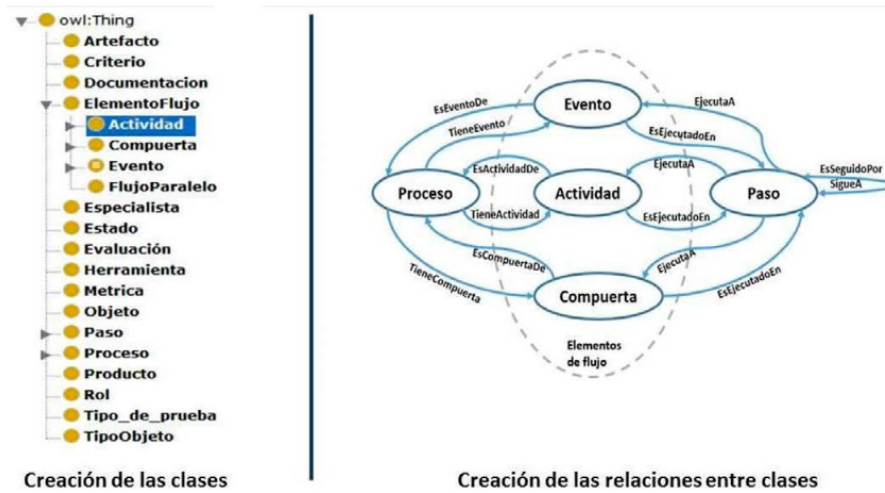


Figura 3. Clases y relaciones entre clases.

FIGURA 3.
Clases y relaciones entre clases.

sf

Creación de las relaciones entre clases

Además de la definición de las clases, la definición de las propiedades es otro de los pasos cruciales durante el desarrollo de una ontología. Las propiedades contribuyen a responder las preguntas de competencia, a la vez que permiten caracterizar los individuos. Las mismas pueden ser de dos tipos: propiedades de objetos (*object properties*) y de datos (*data properties*).

Las propiedades de objeto permiten establecer relaciones binarias entre individuos. Cada propiedad de objeto posee una propiedad inversa. Por lo tanto, se definieron en la ontología un conjunto de propiedades de objetos para relacionar los conceptos identificados para describir el PPL. Por ejemplo, como se muestra en la figura 4, el **Proceso de prueba de liberación** *TieneActividad* y una **Actividad** es *EsRealizadaPor* un **Rol**. A su vez un **Rol** *EsRolDe* un **Especialista** quien *Gestiona* la **Documentación** que *Genera* y *Utiliza* el proceso. Si una **Actividad** *EsRealizadaPor* un **Rol**, se infiere que un **Rol** *Realiza* una **Actividad**.

La expresividad de *OWL* permite detallar las propiedades definidas. Se puede establecer si una propiedad es funcional, transitiva, simétrica, etc. En *OntoPL* se definió que la propiedad *EjecutaA* es funcional debido a que en un **Paso** solo se puede ejecutar un **ElementoFlujo**.

Definición de atributos

Los atributos presentes en el modelo conceptual fueron representados en la ontología con el objetivo de describir cada uno de los conceptos. Es importante representar además del nombre, el dominio a la hora de agregar un nuevo atributo para garantizar la consistencia de los datos de la ontología (ver figura 5).

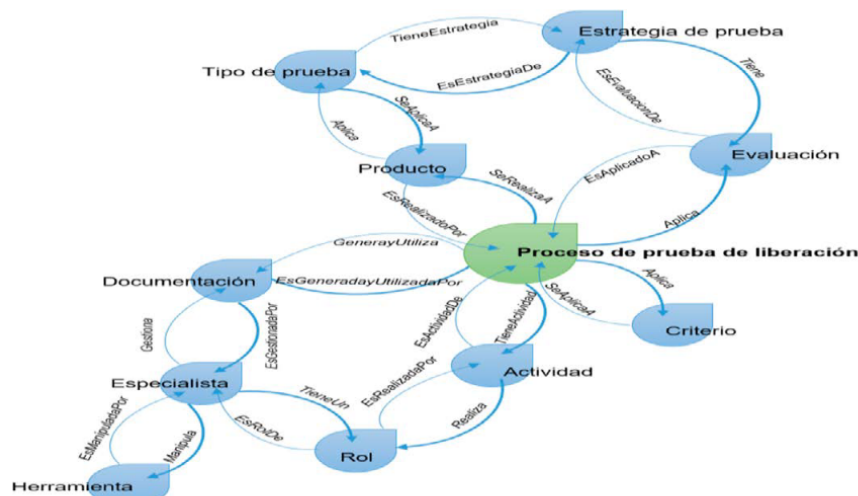


Figura 4. Dimensión de restricciones de procesos de negocio.

FIGURA 4.
Dimensión de restricciones de procesos de negocio.

sf

Declaración de 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 (ver figura 5).

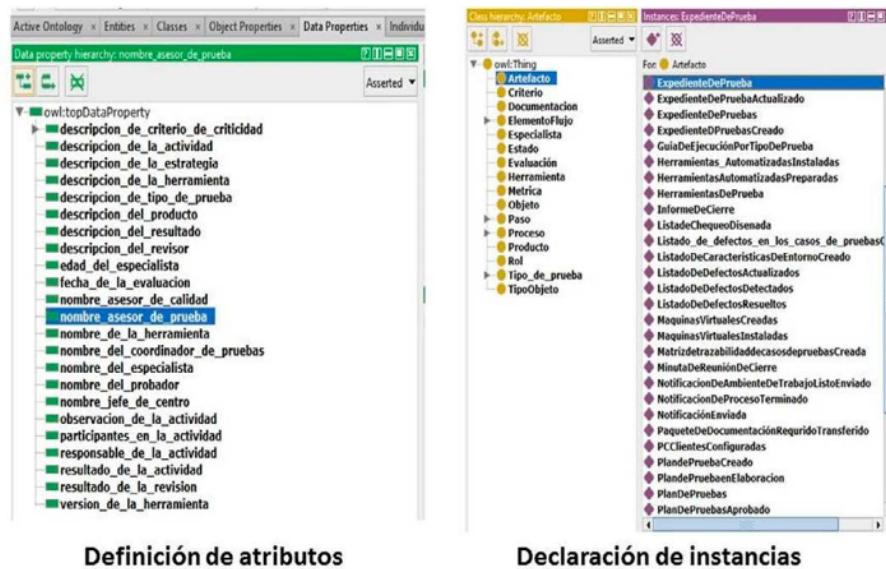


Figura 5. Propiedades de objeto en *OntoPL*.

FIGURA 5.
Propiedades de objeto en *OntoPL*.

sf

Definición de axiomas

El número total de axiomas se va incrementado en la medida que se utilice la ontología. En la figura 6, se representa la estadística relacionada con la ontología desarrollada. Hasta el momento existen un total de 900 axiomas, entre ellos 55 clases, 64 relaciones, 64 atributos y 156 individuos.



Figura 6. OntoPL, su estadística.

FIGURA 6.
OntoPL, su estadística.
sf

Se muestran en la figura 7 los axiomas correspondientes a la clase Artefacto, reflejando entre otros aspectos la definición de la clase, sus atributos e individuos representados hasta el momento.

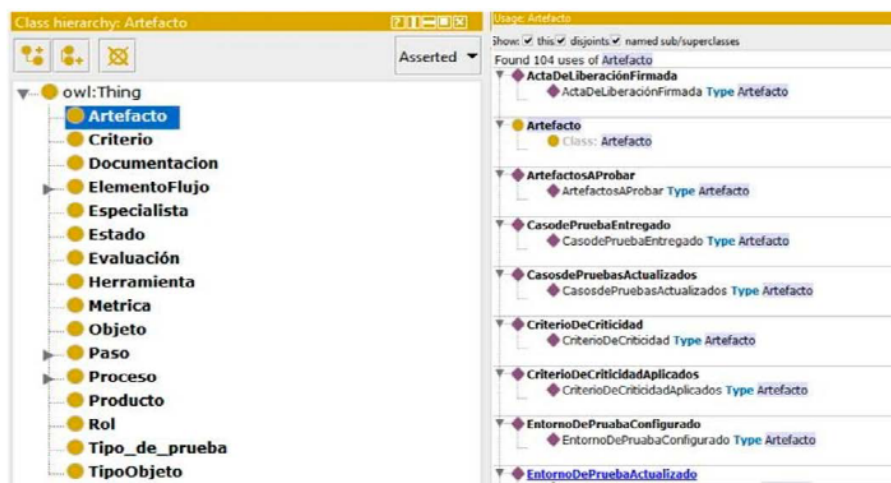


Figura 7. Axiomas de la clase Artefacto.

FIGURA 7.
Axiomas de la clase Artefacto.
sf

Validación de la ontología

Para realizar esta actividad se analizaron varias investigaciones (García, 2009; López et al., 2016; Nemury Silega et al., 2017) y aplicó la propuesta de López (López et al., 2016), por su completitud y multidimensionalidad de las características a analizar. El método propone la comprobación de los siguientes elementos:

- Las condiciones y propiedades como sistema lógico formal.
- El diseño estructural.

- El cumplimiento de los requerimientos para los cuales fue creada.

Por medio de los razonadores, se realizan comprobaciones de las propiedades lógico-formales de las ontologías durante todo el ciclo de vida, garantizando aislar los errores en contextos más reducidos. El diseño estructural se verifica a través de una lista de chequeo, la cual permite evaluar desde etapas tempranas del desarrollo de la ontología y puede ser actualizada a partir de los errores detectados, permitiendo así su evolución. La utilización de casos de prueba permite verificar el cumplimiento de los requerimientos de la ontología expresados en las preguntas de competencia.

A continuación, se detalla la validación de la ontología, teniendo en cuenta los tres aspectos mencionados.

Verificación de las propiedades lógicas formales

El empleo de los razonadores desde el inicio de la implementación de la ontología, garantizó además que se fueran resolviendo de manera inmediata cada una de las inconsistencias presentadas. La figura 8, muestra el resultado que presenta el razonador Pellet al clasificar la ontología desarrollada, mostrando que no existen problemas de inconsistencia en la misma, por lo que se considera una verificación satisfactoria, y las actividades relacionadas con el proceso de prueba de liberación, una vez que inicia el razonador detecta que no existen inconsistencias y se infiere conocimiento a partir de los datos registrados en la base de conocimiento.

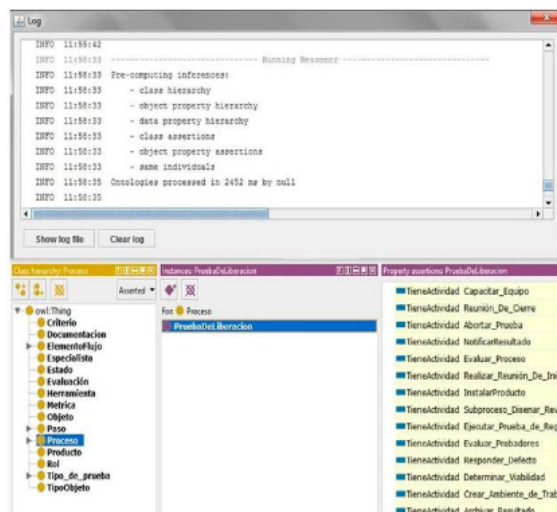


Figura 8. Resultado del razonador en la clasificación de la ontología y un ejemplo.

FIGURA 8. Resultado del razonador en la clasificación de la ontología y un ejemplo.

sf

Verificación del diseño

Para validar este aspecto, se usó la lista de chequeo que se muestra en la tabla 1, elaborada a partir de los errores más comunes detectados en el diseño de ontologías. Ha sido utilizada en otra validación realizada sobre ontologías (López et al., 2016).

Tabla 1: Lista de chequeo para verificación del diseño estructural

Tipo de error	Elemento a verificar
I.	Una misma clase es definida como subclase y superclase al mismo tiempo en diferentes niveles de la taxonomía.
II.	Uso excesivo de la relación es un.
III.	Existencia de más de un concepto principal.
IV.	Existencia de clases incompletas que provocan ambigüedad por no estar correctamente documentadas.
V.	Falta de conocimiento disjunto. No se declara la disyunción entre conceptos provocando una incorrecta formalización del conocimiento.
VI.	Falta de exhaustividad. Se declaran subclases sin tener en cuenta la división completa de los conceptos en partes.
VII.	Existencia de conceptos repetidos.
VIII.	Poca especificación o delimitación de las propiedades que provoca un pobre razonamiento.
IX.	No se corresponden los elementos del dominio con los conceptos declarados o no se corresponde el conocimiento del dominio con los conceptos, relaciones y axiomas declarados.
X.	Existencia de redundancia entre las extensiones disjuntas de un concepto.
XI.	No se tiene en cuenta la traducción de los conceptos de la taxonomía a otros idiomas.
XII.	Falta de estandarización. Los nombres de los términos no siguen un estándar.

TABLA 1:
Lista de chequeo para verificación del diseño estructural

sf

En la primera iteración fueron detectados errores de tipo III, VIII, XII, los que fueron corregidos. Dentro de las insuficiencias detectadas se encontró que la ontología no contaba con un solo concepto principal, por lo que luego de realizar un análisis de cada uno de los conceptos se decidió que el concepto principal fuera Proceso de Prueba de Liberación y asociados a él el resto de los conceptos. Existían pocas propiedades asociadas a los conceptos, para darle solución a este error se analizó el modelo conceptual diseñado y agregaron atributos al mismo que posteriormente se convirtieron en propiedades de las clases o conceptos. Fue detectado el nombre de un concepto y una relación que no estaban descritos de manera estándar. En una segunda iteración se detectaron errores de tipo V y XI, que también fueron solucionados. Se consideró que el error de tipo XI no aplica para la presente investigación, pues desde un inicio se definió que la ontología solo sería en español; será tenido en cuenta como elemento a mejorar con el objetivo de extender la solución a otros entornos de trabajo. Finalmente, en la tercera revisión no se detectaron insuficiencias, considerándose que la ontología desarrollada presenta un diseño correcto (satisfactorio).

Verificación de requisitos

Para la verificación de la ontología pueden ser utilizadas las preguntas de competencia (o requisitos funcionales) del DERO. En la investigación se elaboraron y aplicaron diez casos de prueba para verificar el cumplimiento de los requisitos de la ontología, los cuales se aplicaron en correspondencia con las preguntas de competencia y presentan la siguiente estructura:

Para la verificación de la ontología pueden ser utilizadas las preguntas de competencia (o requisitos funcionales) del DERO. En la investigación se elaboraron y aplicaron diez casos de prueba para verificar el cumplimiento de los requisitos de la ontología, los cuales se aplicaron en correspondencia con las preguntas de competencia y presentan la siguiente estructura:

pregunta de competencia que aborda, escenario de prueba, resultado esperado y resultado obtenido. A modo de ejemplo, se detalla a continuación un caso de prueba (ver tabla 2).

Tabla 2: Ejemplo de Caso de prueba.

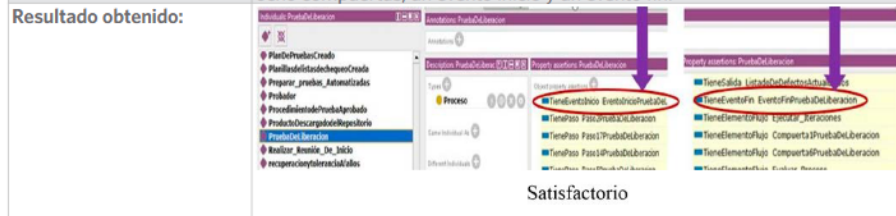
Pregunta de competencia:	¿El proceso de prueba de liberación cumple con los patrones básicos de flujo de trabajo?
Escenario:	En la herramienta Protégé fueron creados los elementos del proceso.
Resultado esperado:	Al aplicar el razonador Pellet el proceso cumple con las restricciones definidas. El proceso Prueba de Liberación tiene 32 pasos, 37 actividades, ocho compuertas, un evento inicio y un evento fin.
Resultado obtenido:	 <p>Satisfactorio</p>

TABLA 2
Ejemplo de Caso de prueba.
sf

Los resultados de los casos de pruebas aplicados fueron satisfactorios; por tanto, la ontología responde a los requisitos para los que fue creada. El nivel de integridad es satisfactorio, teniendo en cuenta los resultados obtenidos en la validación de la misma.

La ontología esta accesible en: <https://drive.google.com/file/d/1w2mrcm-HYrDoufc6dqA-crF3KVGR7Tpr6/view?usp=sharing>

CONCLUSIONES

Al culminar la investigación, se arribó a las siguientes conclusiones:

- El análisis bibliográfico realizado evidenció la necesidad de implementar una ontología para representar el proceso de prueba de liberación.
 - El empleo de herramientas, lenguajes y una metodología, permitió la creación de una ontología para la representación del proceso de pruebas de liberación, contribuyendo a una descripción clara y precisa por lo que eleva su comprensión.
 - La validación de la ontología permitió verificar la correctitud de su diseño, condiciones y propiedades como sistema formal, así como el cumplimiento de los requisitos para los que fue creada. Se recomienda como trabajo futuro:
 - Continuar enriqueciendo la ontología, de manera que permita agregar valor semántico a la misma, introduciendo la mayor cantidad de instancias y relaciones para que se haga un uso más efectivo y real de la misma como parte del mantenimiento.
 - Implementar un Sistema de Información Basado en Ontologías que permita consultar la información disponible en la ontología.
 - Extender el modelo ontológico para que pueda ser utilizada por diversas entidades relacionadas con la evaluación de productos en la industria cubana de *software*.

AGRADECIMIENTOS

AGRADECIMIENTOS

Agradecemos a la Dirección de Calidad de la UCI, al Grupo de Investigación Ingeniería y Calidad de *Software* (Grisoft) y al proyecto de investigación “Modelo para un Laboratorio Especializado de Pruebas de *Software*”.

REFERENCIAS

- Alvarado, R. D. (2010). *Metodología para el desarrollo de ontologías*. [https://Es.Slideshare. Net/Iceman1976/ Metodologia-Para-Ontologias?Qid=b1d3dc5d-7f43-4035-8cc2- 80fef4e1a- c69&v=&b=&from_search=1](https://Es.Slideshare.Net/Iceman1976/Metodologia-Para-Ontologias?Qid=b1d3dc5d-7f43-4035-8cc2-80fef4e1a-c69&v=&b=&from_search=1).
- Castañeda Martínez, A., Parker Leyva, C., Fernández Pérez, Y. and López Rodríguez, Y. (2018). Ontología de apoyo a las pruebas de software en la UCI. *Revista Cubana de Ciencias In- formáticas*, 12(Supl. 1, 222-235).
- Duarte, K. C. and R. A. F. (2000). Uma Ontologia de Qualidade de *Software*. *Repositorio Ins- titucional de Universidade Federal Do Espírito Santo*.
- Echeverría Perez, D. and Y. F. P. (2011). *Desarrollo de una ontología de apoyo al procedimien- to del Departamento de Pruebas de Software* (Tesis de Máster).
- Estradet, L. C. (2018). *Ontología de apoyo al proceso de evaluación de productos de software en la UCI*.
- Felipe Alfonso, R. (2011). *Implementación de una ontología para el proceso de ingeniería de requisitos del modelo cubano de calidad para el desarrollo de aplicaciones informáticas en la industria cubana de software*. UCI.
- Fernández-Pérez, Y. (2018). *Modelo computacional para la evaluación y selección de produc- tos de software* (Tesis doctoral).
- Fernández Hernández, A. (2015). *Modelo Ontológico de recuperación de información para la toma de decisiones en Gestión de Proyectos* (Tesis doctoral).
- Ferreira, M. (2006). Medición del *Software* Ontología y Metamodelo. *Informe Técnico Univer- sidad de Castilla- La Mancha*.
- GARCÍA, M. N. (2009). *Modelado y análisis de sistemas CSCW siguiendo un enfoque de in- geniería dirigida por ontologías* (Tesis Doctoral).
- Grass, O. Y. R. (2018). *Método para la descripción y validación de procesos de negocio de ges- tión empresarial basado en ontología* (Tesis de Máster).
- Horridge, M., *et al.* (2009). A practical guide to building owl ontologies using protégé 4 and co-ode tools edition 1. 2. *The University of Manchester*,.
- ISO/IEC/IEEE. (2017). ISO/IEC/IEEE 24765 Systems and Software Ingeneering- Vocabulary. Switzerland: ISO.
- Lopez Rodriguez, Y. (2017). *Método para la integración de ontologías en sistemas relacionales para la evaluación de créditos* (Tesis de Máster).
- López, Y., Hidalgo, Y. and Silega, N. (2016). Método para la integración de ontologías en un sistema para la evaluación de créditos. *Revista Cubana de Ciencias Informáticas*, 10(4), 97-111.
- Lugones, M. L. G. (2017). *Ontología para el proceso de medición y mejora defnido en un Mo- delo de Calidad para el Desarrollo de Aplicaciones Informática* (Tesis de Máster).
- Luna, J.A.G.B., M. L. (2012). Metodologías y métodos para la construcción de ontologías. UTP Universidad Técnica de Pereira, 20(50).
- Silega, N. (2014). Método para la transformación automatizada de modelos de procesos de ne- gocio a modelos de componentes para Sistemas de Gestión Empresarial. *Centro de Infor- matización de Entidades*, 5.
- Silega, Nemury, Teresa-Loureiro, T., Noguera, M. and Pedro-Febles, J. (2017). Framework ba- sado en ontología para la descripción y validación de procesos de negocio. *Ingeniería In- dustrial*, 38(3), 276–288.
- Souza, E. F. (2013). Using Ontology Patterns for Building a Reference Software Testing Onto- logy. *17th IEEE International Enterprise Distributed Object Computing Conference Wor- kshops*, 21–30.
- Vázquez, A. A. (2017). Método para la evaluación de las características de calidad según la ISO/IEC 25000.