Facultad 1

Título: Desarrollo de una ontología para las pruebas de usabilidad en el laboratorio de pruebas de la Universidad de las Ciencias Informáticas.

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

Autor(es): Iván Gutiérrez Del Toro
Tutor(es): Dr. C. Yaimí Trujillo Casañola
Ing. Dayamí Pompa Echavarría
Consultante: Ing. Leosmay Carrión Estradet

La Habana, 11 de 2021

Año 63 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con **título "Desarrollo de una ontología para las pruebas de usabilidad en el laboratorio de pruebas de la Universidad de las Ciencias Informáticas."**, *en* conceder a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los ___ días del mes de ___ del año ____.

Iván Gutiérrez Del Toro

_____
Firma del Autor

Dr. C. Yaimí Trujillo Casañola

_____
Firma del Tutor

Ing. Dayamí Pompa Echavarría

_____
Firma del Tutor

AGRADECIMIENTOS

A mi madre Yanelis Del Toro por brindarme su amor incondicional y apoyarme en todas las decisiones que he tomado.

A mi padre Iván Gutiérrez por ser un ejemplo a seguir y siempre estar presente brindarme una solución a cada problema.

A toda mi familia por el apoyo incondicional que me brindan.

A todos mis amigos que de una forma u otra aportaron un granito de arena por pequeño que fuese, en especial a Hugo Alberto, Eloy Cruz, Rpyland Rodriguez, Angel Yimi y Angel Hidalgo.

A mis tutores que sin ustedes este trabajo no hubiese sido posible.

DEDICATORIA

Este trabajo se lo dedico a mis padres, por siempre estar presente con su apoyo, con un consejo, con una sonrisa o con una solución en cada momento. Sé que jamás encontrare la forma de poder retribuirle todo lo que han hecho por mí, este logro es dedicados a ellos y quiero que sepan que todos mis esfuerzo y acciones son inspirados en ustedes.

RESUMEN

En el presente trabajo se ha establecido una ontología de dominio sobre las pruebas de usabilidad realizadas en la dirección de calidad de la Universidad de las Ciencias Informáticas, la cual se estructura sobre programa Protègè, y sirve de base para una mejor gestión del conocimiento del dominio, de los conceptos y formas de llevar a cabo esta actividad. Las validaciones demostraron que la solución favorece al proceso de de pruebas de usabilidad e incide sobre sus resultados garantizando eficiencia y eficacia en los procesos de planificación, seguimiento y control de esta actividad.

Palabras claves: ontología, pruebas de usabilidad, dominio, gestión del conocimiento.

ABSTRACT

In the present work, a domain ontology has been established on the usability tests carried out in the quality management of the University of Informatics Sciences, which is structured on the Protègè program, and serves as a basis for better management of domain knowledge, of the concepts and ways of carrying out this activity. The validations showed that the solution favors the usability testing process and affects its results, guaranteeing efficiency and effectiveness in the planning, monitoring and control processes of this activity.

Keywords: ontology, usability tests, domain, knowledge management.

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La industria del software es una de las que mayor desarrollo ha alcanzado en los últimos tiempos, debido a los constantes avances tecnológicos y una masiva tendencia hacia la automatización e informatización de tareas. Para lograr el desarrollo de un software es necesario transitar por una serie de etapas, constituyendo una buena práctica la realización de actividades de pruebas desde etapas tempranas. El desarrollo del software ha aumentado considerablemente y cada vez son más los sectores de la sociedad que automatizan sus procesos (la medicina, la aeronáutica, la educación, la economía, etc.), de ahí que la calidad en el software sea un elemento imprescindible, por su repercusión en los costos finales, como elemento diferenciador de la competencia y de imagen frente a sus clientes (Brito Riverol, Capote García y Febles Rodríguez 2015)

En la actualidad, las empresas que producen software tienen que velar necesariamente por la calidad de su producción, debido a la competitividad que existe en esta rama y a la velocidad de respuesta que es necesario brindar para aumentar la satisfacción del cliente(«EXPERIENCIAS EN LA IMPLANTACIÓN DE UN SISTEMA DE GESTIÓN DE LA CALIDAD PARA EL PROCESO DE PRODUCCIÓN DE SOFTWARE» 2004). Muchos de los desarrolladores de software piensan que la calidad solo es aplicable al producto, esta se debe medir desde su inicio hasta la entrega al usuario. En el concepto de calidad de un software, la usabilidad es una característica clave en el desarrollo del software, lo que permite medir la facilidad de uso, aprendizaje, así como la flexibilidad necesaria para atraer a los usuarios.

La usabilidad determina, qué el usuario busca, qué quiere, cómo lo quiere y qué tan rápido desea la información. No solo que la información se vea bonita o muy adornada, sino que esté disponible y organizada de manera que se pueda encontrar inmediatamente lo que se necesita y que la respuesta de esa información sea concisa. Otro de los objetivos que persigue es que no existan demasiados elementos que hagan a la persona perderse en la pantalla o que provoquen aburrimiento o fastidio. La finalidad de la usabilidad es que la persona vea la información de manera tan sencilla que desee seguir explorando y viendo todos los contenidos existentes (Corrales, Abascal y Real 2017).

Con el objetivo de garantizar la calidad en el proceso de desarrollo de software, la UCI basa su actividad productiva en el Modelo de Capacidad y Madurez Integrada (CMMI por sus siglas en inglés)y fue certificada con el nivel 2 en la variante de desarrollo de dicho modelo(«Proceso productivo de la UCI evaluado con CMMI nivel 2 | Universidad de las Ciencias Informáticas» 2015) y se encuentra en la realización de un proyecto para certificarse en el nivel 3, por lo que se definen actividades de aseguramiento de la calidad, verificación y validación (López, Pérez y Feria 2017). Dentro de los

subprocesos para la gestión de las actividades de calidad se encuentran la ejecución de las pruebas a nivel de proyectos, de centro y de gerencia. Las pruebas a nivel de gerencia se desarrollan en la Dirección de Calidad de Software de la Universidad, denominadas pruebas de liberación de producto dentro de estas se encuentran las de usabilidad.

Particularmente el proceso de evaluación de la usabilidad realizada en el laboratorio de la Dirección de Calidad de la UCI resulta complejo por la cantidad de personal involucrado de diferentes áreas del conocimiento, con diversos niveles de experiencia, además influye también el volumen de documentación generada. Todo ello, provoca dificultades en el entendimiento y comprensión total de la evaluación, fundamentalmente de los términos y conceptos utilizados. No existe consenso sobre la terminología utilizada y no se gestiona el conocimiento generado durante la ejecución de la misma. A partir de las insuficiencias anteriores surge la necesidad de encontrar técnicas y herramientas que proporcionen un vocabulario común, para resolver el problema de integridad e inconsistencia identificada entre el personal involucrado en las pruebas de usabilidad realizadas en la Dirección de Calidad de la Universidad de la Ciencias Informáticas (UCI).

Atendiendo la situación planteada y enfocándose en su solución, se propone como **problema de investigación:** ¿Cómo contribuir a la representación y reutilización del conocimiento generado en las pruebas de usabilidad ejecutadas en el laboratorio de pruebas de software en la UCI?

Definiendo como **objeto de estudio:** la gestión del conocimiento basado en ontologías y enmarcando como **campo de acción** la ontología para gestionar el conocimiento en las pruebas de usabilidad en el laboratorio de pruebas de la Universidad de las Ciencias Informáticas.

Para dar solución al problema de investigación se propone como **objetivo general**: Desarrollar una ontología que permita representar y reutilizar el conocimiento generado en las pruebas de usabilidad ejecutadas en el laboratorio de pruebas de software en la UCI, permitiendo la consistencia, organización y comunicación del conocimiento existente y generado. Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos:**

1. Sistematizar los principales referentes teóricos sobre las ontologías y las pruebas de usabilidad.
2. Identificar la metodología, herramienta y tecnología a utilizar para el desarrollo de la ontología.
3. Desarrollar la ontología apoyándonos en las herramientas y tecnologías seleccionadas.

4. Validar la propuesta ontológica en función de la representación y reutilización del conocimiento existente y generado.

Para dar cumplimiento al objetivo trazado se plantean las siguientes **tareas de la investigación:**

1. Definición de los elementos conceptuales referentes a las pruebas de usabilidad y las ontologías.

2. Análisis de la gestión del conocimiento y formas de representar el mismo como base teórica en el desarrollo de la investigación.

3. Análisis de las tendencias actuales y metodologías empleadas en la elaboración de ontologías para conocer cuáles son la más adecuadas a la investigación.

4. Análisis de las ontologías existentes en el área de la calidad de software para identificar otras soluciones que se puedan reutilizar en el desarrollo de la ontología propuesta.

5. Especificación de los requisitos de la ontología para el desarrollo de la ontología yvalidación de la misma.

6. Elaboración del modelo conceptual para tener una descripción más detallada de los términos a utilizar así como conocer su tipo.

7. Implementación de la ontología siguiendo los pasos de la metodología seleccionada

8. Evaluación de la ontología para asegurar la calidad del producto final.

Métodos Teóricos:

- Histórico-Lógicos: para el estudio crítico de trabajos anteriores, y para utilizar estos como punto de referencia y comprobación de los resultados alcanzados.
- Análisis y Síntesis: Para descomponer el problema de investigación en elementos por separados y profundizar en el estudio de cada uno de ellos para luego sintetizar en la solución propuesta.
- Inductivo-Deductivo: para arribar a conclusiones sobre el problema de la investigación, a partir de la generalización y especificación de los resultados parciales que se obtengan.

Métodos Empíricos:

- Análisis documental: facilitó la investigación sobre el dominio de la bibliografía consultada, seleccionando información más certera sobre el tema.
- Observación: facilito la obtención de información, se utilizó para reconocer las particularidades del proceso de pruebas de usabilidad.

Métodos Matemáticos:

- Métodos estadísticos: Para el procesamiento de los datos arrojados por las entrevistas y encuestas.

CAPÍTULO I: FUNDAMENTACION TEORICA

En el presente capítulo se realiza un análisis de la gestión del conocimiento y las formas de representar el mismo. En correspondencia con el capítulo se analizan conceptos y definiciones sobre la calidad de software y el proceso de pruebas de usabilidad. Se examinan las tendencias actuales y metodologías empleadas en la elaboración de ontologías. Se realizó un análisis comparativo de algunas metodologías, tecnologías y herramientas para la creación de ontologías, con el fin de seleccionar las más adecuadas en el marco de la investigación.

1.1Calidad de software y Pruebas de Usabilidad

Los productos de software, sistemas y/o aplicaciones son creadas, desarrolladas e implementadas por seres humanos y por ende en cualquiera de sus etapas de creación se puede cometer un error, al generarse esa “error” se puede conllevar a un defecto en el software. Si no se ha identificado ese defecto y el software o la aplicación se ejecuta, hay un alto riesgo de que la aplicación no haga lo que debería hacer o el objeto para lo cual fue creada, es decir se genera un fallo o desperfecto, lo que podría generar una catástrofe(Julián Andrés Mera Paz 2015). Por tal motivo el software antes de ser liberado debe alcanzar un alto nivel de calidad.

El diccionario de la Real Academia de España hace referencia al término calidad como “Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”. Llevada esta definición al campo de la ingeniería de software, la IEEE Std 610, señala que “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”(IEEE Standard Computer Dictionary 1991).

Sommerville sostiene que la calidad del software es un concepto complejo que no es directamente comparable con la calidad de la manufactura de productos. En la manufacturación, la noción de calidad viene dada por la similitud entre el producto desarrollado y su especificación. En un mundo ideal, esta definición debería aplicarse a todos sus productos, pero, para sistemas de software, existen cuestiones específicas que impiden aplicar este mecanismo(Ian Sommerville 2005). Por su parte, Pressman define como calidad de software al " proceso eficaz de software que se aplica de manera que crea un producto

útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan "¨(Roger S. Pressman 2010).

Otra definición a evaluar es la que nos muestra la ISO (International Organization for Standardization – Organización Internacional para la Estandarización)define la calidad del software como la concordancia con los requerimientos funcionales cumpliendo con las políticas y el rendimiento establecidos por el negocio(Norma ISO/IEC 25000 2015)

Cuando se habla de calidad de software no podemos dejar de hacer referencia a las pruebas de software pues son las que nos permiten conocer la calidad del mismo, así como los posibles fallos que puedan existir a corto, medio o largo plazo con el fin de disminuir los costos finales. Según define la IEEE las pruebas son el proceso de operar un sistema o componente bajo condiciones específicas, observar o registrar el resultado y hacer una evaluación de algún aspecto del sistema o componente(IEEE Standard Glossary of Software Engineering Terminology 1990)

Existen diferentes tipos de pruebas de software, la que buscan probar una funcionalidad del software, las que buscan probar una característica no funcional como puede ser la fiabilidad y las que buscan probar una estructura de software. Teniendo en cuenta esto, vamos a diferenciar los tipos de pruebas de en tres puntos principales (José Manuel Sánchez Peño 2015).

- Pruebas funcionales
- Pruebas no funcionales
- Pruebas estructurales

Prueba funcional: Sirven para verificar los requerimientos funcionales (establecidos en las especificaciones, conceptos, casos de estudios, reglas de negocio o documentos relacionados). Se basan en funciones, prestaciones y en su interoperabilidad con sistemas específicos, y deben llevarse a cabo en todos los niveles de prueba. Es importante mencionar que se orientan en el comportamiento externo de un producto o aplicativo software, en las pruebas de caja negra (Julián Andrés Mera Paz 2016).

Pruebas no funcional: Evalúan las características no funcionales de los sistemas, se pueden realizarse en todos los niveles de pruebas. Hacen referencia a las pruebas necesarias para medir las características de los sistemas y software que pueden cuantificarse según una escala variable. Se debe tener en cuenta

que se orientan hacia el comportamiento externo del software y en la mayoría de los casos utilizan técnicas de diseño de pruebas de caja negra (Julián Andrés Mera Paz 2016).

Pruebas estructurales: Se derivan de las pruebas basadas en la estructura interna o la implementación del sistema. La estructura interna puede incluir código, arquitectura, flujos de datos dentro del sistema. Pueden realizarse en todos los niveles de prueba. Son las más idóneas, después de las técnicas basadas en la especificación, para ayudar a medir la exhaustividad de las pruebas mediante una evaluación de la cobertura de un tipo de estructura. Para todo proceso de pruebas se debe tener clara la diferencia al clasificar los tipos de pruebas, esto contribuye a un análisis sólido del plan de pruebas y a estructurar los casos de pruebas y creación de la respectiva matriz; se tributa además a la eficiencia en el proceso de calidad del producto software (Julián Andrés Mera Paz 2016).

En esta investigación nos centraremos en un tipo específico de prueba no funcional ya que existen varios como las de Confiabilidad, Carga, Estrés, Usabilidad, Mantenibilidad entre otras. Estaremos abordando diferentes conceptos y definiciones sobre las pruebas de usabilidad. La característica de usabilidad según la ISO/IEC 25000, es la “Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones”. Esta característica a su vez se divide en las siguientes subcaracterísticas(Norma ISO/IEC 25000 2015):

- **Capacidad para reconocer su adecuación:** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje:** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado:** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario**: Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario:** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad:** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Por su parte (Abascal 2017) sostiene que las pruebas de usabilidad son un servicio de aseguramiento de calidad que consiste en probar un producto y detectar no conformidades sobre su facilidad de uso. El objetivo principal de estas pruebas es identificar los problemas de usabilidad, para mejorar el producto y la satisfacción de sus clientes. Proporcionar retroalimentación para mejorar el diseño. Valorar en qué medida se cumplen los objetivos de los usuarios y la organización y poder exponer cuantitativamente el grado de usabilidad que presenta el producto probado(Corrales, Abascal y Real 2017). Según el autor Sánchez Peño las pruebas de usabilidad miden la facilidad de uso, efectividad y satisfacción, siempre dentro de un grupo específicos de usuarios(José Manuel Sánchez Peño 2015).

Jakob Nielsenidentificado como el promotor de la usabilidad, ha definido 10 principios de usabilidad web con el objetivo de identificar posibles problemas de usabilidad. Los cuales se relacionan en la investigación con las subcaracterísticas de la usabilidad.

- **Visibilidad del estado del sistema.** El sitio web o aplicación debe mantener siempre informado al usuario de lo que está ocurriendo y brindarle una respuesta en el menor tiempo posible.
- **Relación entre el sistema y el mundo real.** El sitio web o aplicación debe utilizar el lenguaje del usuario, con expresiones y palabras que le resulten familiares. La información debe aparecer en un orden lógico y natural.
- **Libertad y control por parte del usuario.** En caso de elegir alguna opción del sitio web o aplicación por error, el usuario debe disponer de una “salida de emergencia” para abandonar el estado no deseado en que se halla. Debe poder deshacer o repetir una acción realizada.
- **Consistencia y estándares.** Los usuarios no tienen por qué saber que diferentes palabras, situaciones o acciones significan lo mismo. Es conveniente seguir convenciones**.**
- **Prevención de errores.** Es importante ayudarle al usuario a que no caiga en un error. La funcionalidad de autocomplete de los buscadores ayuda a que una persona no tenga que escribir toda la palabra y no se equivoque**.**
- **Reconocer antes que recordar.** Hacer visibles acciones y opciones para que el usuario no tenga qué recordar información entre distintas secciones o partes del sitio web o aplicación. Es importante mantener a nivel de diseño visual un estándar para que los elementos de interface sean consistentes en diferentes pantallas.

- **Flexibilidad y eficiencia en el uso.** Los aceleradores o atajos de teclado pueden hacer más rápida la interacción para usuarios expertos, de tal forma que el sitio web o aplicación sea útil tanto para usuarios básicos como avanzados.
- **Diseño estético y minimalista.** Las páginas no deben contener información innecesaria. Cada información extra compite con la información relevante y disminuye su visibilidad.
- **Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores.** Los mensajes de error deben estar redactados con un lenguaje simple, lo peor son errores como “Error 34-x1” que no le dicen nada al usuario de cómo puede recuperarse de ese error, deben ofrecerse alternativas para que el usuario pueda continuar realizando la tarea o recuperando lo último que hizo.
- **Ayuda y documentación.** Aunque es mejor que el sitio web o aplicación pueda ser usado sin ayuda, puede ser necesario proveer cierto tipo de ayuda. En este caso, la ayuda debe ser fácil de localizar, especificar los pasos necesarios y no ser muy extensa. en los sitios móviles se está utilizando un mini tutorial o tour por el App o el sitio en donde de manera sencilla se exponen las funcionalidades principales, no es tan aburrido como tener que leer documentos extensos de ayuda.

Los métodos de evaluación de usabilidad pueden ser **heurísticos** o **empíricos**. Los heurísticos (también llamados no empíricos), implican la participación de expertos especialistas en usabilidad. Los empíricos constan de técnicas e instrumentos que requieren de la participación de usuarios. Según el enfoque tradicional, las pruebas de usabilidad se aplican sobre el producto software para garantizar o determinar si el mismo alcanza un nivel aceptable de usabilidad (Mascheroni, M. et al. 2012).

La usabilidad es una cualidad que todos los productos desean tener pero que no todos consiguen, por esto las pruebas de usabilidad son cada vez más aceptadas por las empresas que someten a sus productos a unas pruebas de usabilidad cada vez mayores. Este tipo de pruebas consiste en la prueba del producto por parte de muchos usuarios para comprobar que el producto funciona correctamente cumpliendo el propósito para el cual fue diseñada(José Manuel Sánchez Peño 2015):

- **La eficiencia:** si el programa funciona con una rapidez suficiente para que el usuario lleve a cabo sus tareas eficientemente.
- **La efectividad:** los usuarios comprobaran que el producto funciona como esperaban.
- **La facilidad de manejo:** donde se comprobará si el usuario es capaz de operar con el producto con una cierta facilidad teniendo un periodo de formación.

- **La satisfacción:** satisfacción de los usuarios y la percepción y opiniones que tienen el producto en general.

Se hace necesario organizar la información y recuperar el conocimiento que se genera en el proceso de evaluación de usabilidad, para cumplir con los objetivos de este trabajo y tener una mejor comprensión del proceso de evaluación. Para esto se hizo un análisis de la gestión del conocimiento.

1.2 Gestión del conocimiento

El primer concepto a aclarar es qué se entiende por conocimiento. Sin entrar en profundidades filosóficas, que darían lugar a mucho más que un artículo, el conocimiento en una organización se produce cuando un individuo de la misma hace uso de lo que sabe y de la información que tiene disponible para la resolución de un problema o el desarrollo de un proyecto(Carlota Bustelo Ruesta y Raquel Amarilla Iglesias 2001).

El conocimiento es considerado uno de los activos más valiosos en las organizaciones, aún por encima de los activos tangibles tales como la infraestructura o los recursos financieros, ya que éste está intrínseco en el recurso humano pues se origina y reside en las mentes de las personas y se desarrolla en las habilidades y experiencia para la ejecución de sus actividades. En este sentido no puede perderse de vista que lo más importante en una empresa son las personas que participan en su engranaje (Casttrillon y Eucario 2018). Partiendo de esta definición podemos comprender el papel fundamental que juega el conocimiento dentro de cualquier actividad, proyecto o proceso. Según (Manjón 2010), “el conocimiento lo entendemos como el conjunto de saberes y habilidades (saber hacer) que atesora la empresa, sobre los cuales basa la organización su proposición de valor para el mercado”.

En este punto ya estamos en condiciones de abordar los conceptos y definiciones de diferentes autores sobre lo que es la gestión del conocimiento. Para(Carlota Bustelo Ruesta y Raquel Amarilla Iglesias 2001) la gestión del conocimiento es todo el conjunto de actividades realizadas con el fin de utilizar, compartir y desarrollar los conocimientos de una organización y de los individuos que en ella trabajan, encaminándolos a la mejor consecución de sus objetivos.

Por su parte (Mario Pérez-Montoro Gutiérrez 2008) sostiene que la gestión del conocimiento es la disciplina que se encarga del estudio del diseño y de la implementación de sistemas cuyo principal

objetivo es identificar, capturar y compartir sistemáticamente el conocimiento involucrado dentro de una organización, de forma que este pueda ser convertido en valor para ella. De otra manera podemos ver que la gestión del conocimiento es considerada el conjunto de procesos que permiten utilizar el conocimiento como factor clave para añadir y generar valor a la organización(Pérez y Dressler 2007).

Todas estas definiciones anteriores nos hacen llegar a la conclusión que la gestión del conocimiento es el proceso por el cual una organización, facilita la trasmisión de la información y el uso del conocimiento, de una manera sistemática y eficiente. El conocimiento no siempre está disponible cuando es necesario para la organización. Para tratar este problema han surgido nuevas formas de gestionarlo. El objetivo es que todo el conocimiento que reside en una organización pueda ser utilizado por quien lo necesite para actuar de manera adecuada en cada momento. Para lograr este cometido intervienen otras disciplinas como los sistemas de información y la organización del conocimiento.

La Organización del Conocimiento(OC) estudia las diferentes fórmulas de trabajo existentes, en especial en lo que tiene que ver con las fórmulas analíticas y de representación y con los lenguajes documentales, sin desdeñar aspectos como los metadatos, los enlaces, los nuevos desarrollos existentes, su relación con la arquitectura de la información, etc. (Jiménez 2003). Según los autores(Mora y Arias 2018)la OC como una disciplina dentro de las ciencias de la información, “se encarga de estudiar el tratamiento y la recuperación del conocimiento, así como la construcción y control de los lenguajes e instrumentos utilizados en los procesos de representación de los documentos que son producto del conocimiento humano, permitiendo así su posterior recuperación por parte de los usuarios”. Por su parte (Luna González 2015) sostiene que la organización del conocimiento “establece los sistemas de representación de la realidad del autor de dicho conocimiento, con la finalidad de que el usuario final pueda acceder y comprender el contexto en el que fue creado un conocimiento específico”.

Dentro de los beneficios de la OC se encuentran capturar y compartir buenas prácticas, proporcionar formación y aprendizaje, gestionar las relaciones con los usuarios, desarrollar inteligencia competitiva, proporcionar un espacio de trabajo, gestionar la propiedad intelectual, realizar las publicaciones web y reforzar la cadena de mando retención de los conocimientos del personal, mejoramiento de la satisfacción de los usuarios y/o clientes, incremento de los beneficios y acortamiento de los ciclos de desarrollo de productos(Amanda Valdivieso Martinez 2020).

Al tener en la problemática que se aborda en la investigación, el volumen de artefactos e información que se generan en el proceso de pruebas de usabilidad, el número de pruebas que se ejecutan

constantemente, superan la capacidad humana para procesarlos. El propósito es estructurar la información del proceso de pruebas para que la organización y su personal puedan reutilizarlo. 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)

Los Sistemas de Organización del Conocimiento (SOCs) apoyan la recuperación efectiva de los contenidos digitales. Estos sistemas se enfocan a todos los tipos de proyectos para organizar información y para promover el manejo del conocimiento, además incluyen esquemas de clasificación para organizar los materiales a un nivel general, encabezamientos de materia que otorgan un acceso más detallado y archivos de autoridad que controlan varias versiones de información clave (nombres geográficos y personales). También incluyen proyectos menos tradicionales, como ontologías y redes semánticas(Vargas y Santamaría 2008).

En cuanto a estructura, los SOC son esquemas gráficos y/o textuales del universo de conocimiento, fundamentados en tres aspectos(Sánchez 2017):

- **Conceptos** como elementos representadores.
- **Categorías** para establecer niveles entre los elementos conceptuales.
- **Relaciones** entre las entidades conceptuales que los forman.

Para describir a los Sistemas de Organización del Conocimiento (SOC) tenemos que tomar en cuenta las características de estructura y complejidad, las relaciones entre términos y su función. Por ello, podemos decir que los sistemas se agrupan dentro de tres categorías generales(Vargas y Santamaría 2008):

- **Listas de términos** (entre las listas de términos se encuentran los archivos de autoridad, glosarios, diccionarios y diccionarios gráficos), en éstas se pone énfasis en listas de términos frecuentemente con definiciones. Si bien esta categoría es de uso ya tradicional en las bibliotecas –sean o no digitales−, en la actualidad cobran mayor realce, sobre todo al momento de plantear la posibilidad de crear redes de bibliotecas digitales en donde será necesario hacer precisiones de idioma, región, etcétera.
- **Clasificaciones y categorías** (se encuentran los Encabezamientos de materia, Esquemas de Clasificación, Taxonomías y Esquemas de Categorización), que ponen énfasis en la creación de grupos de términos.

- **Listas de relaciones** (se encuentran los tesauros, redes semánticas y ontologías), en éstas se pone énfasis en las relaciones entre términos y conceptos.

Las últimas dos décadas del pasado siglo han sido variables, ya que el organizar por temas la información se ha insertado en un nuevo contexto imbuido en las tecnologías, los recursos digitales, el internet y la búsqueda en línea. Derivado de ello, a los SOCs formales o tradicionales se han incorporado otras herramientas de arreglo temático, como son las folksonomías, las taxonomías digitales y las ontologías(Vargas y Santamaría 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. Son una manera de representar el conocimiento en una determinada disciplina(Carrión Estradet 2018).

1.3 Ontología

A la luz de la literatura especializada, con la palabra ontología se está apuntando al mismo tiempo a cosas que, aun estando relacionadas en muchos sentidos, presentan diferencias evidentes. En cualquier caso este trabajo no se interesa por el término ontología en su derivación filosófica inicial. Desde el campo de la informática las ontologías son herramientas que especifican el conocimiento y la dinámica de un dominio de interés, por medio de modelos, aceptados de manera universal, que faciliten su comprensión (Checa Rojas y Rojas Alvarado 2014).Una de las definiciones de ontología más utilizada en el área de la informática es la de Gruber el cual plantea que una otología es una especificación explícita de una conceptualización(Gruber 1995). Por su parte Guarino define la ontología como un producto de ingeniería consistente en un vocabulario específico usado para describir una realidad más un conjunto de asunciones relacionadas con el significado del vocabulario(Guarino 1998).

Según (Rojas Grass 2018) en la actualidad las ontologías cumplen diversas funciones esencialmente se manifiestan en la teoría de contenido ya que su contribución principal es identificar clases específicas de objetos y relaciones de un dominio. A continuación se mencionan las utilidades que tienen las ontologías en la actualidad según la autora antes mencionada.

- Sirven para entender como diferentes sistemas comparten información.
- Se utilizan para descubrir distorsiones que puedan presentarse en los procesos cognitivos de aprendizaje en un mismo contexto.

- Sirven para formar patrones para el desarrollo de Sistemas de Información. En el ámbito del software se ha utilizado para describir las propiedades del software (componentes, arquitecturas, lenguajes de definición).
- Permiten compartir y reutilizar conocimiento común.
- Ayudan a establecer comunicación entre personas y organizaciones con el fin de unificar diferentes áreas de investigación.

Las características y ventajas de las ontologías, mencionadas anteriormente fundamentan la decisión de utilizarlas en la presente investigación como base del método propuesto para la gestión del conocimiento arrojado en las pruebas de usabilidad.

Existen diferentes maneras de clasificar las ontologías, una de ellas es la planteada por (Guarino 1998), quien atendiendo el nivel de generalidad plantea 4 tipos de ontologías:

- **Ontologías de alto nivel:** describen conceptos generales, como espacio, tiempo, materia, objeto, etc., que normalmente son independiente de un dominio o problema particular.

- **Ontologías de dominio:** proporcionan un vocabulario genérico para describir conceptos, y relaciones entre ellos, de un dominio en específico. Los conceptos de una ontología de dominio normalmente son especializaciones de conceptos que aparecen en ontologías de alto nivel. Lo mismo ocurre con las relaciones entre ellos.

- **Ontologías de tareas:** prescriben el vocabulario relativo a una tarea genérica o actividad, como por ejemplo, diagnosticar, planificar y vender. Para ello al igual que las ontologías de dominio, se especializan los términos introducidos en ontologías de alto nivel.

- **Ontologías de aplicación:** describen conceptos que dependen de dominios y tareas particulares, y que son especializaciones de ambos tipos de ontologías relacionadas. Con frecuencia, estos conceptos corresponden a roles desempeñados por las entidades de un dominio al realizar ciertas tareas.

Para garantizar la representación del conocimiento, las ontologías están compuestas por los componentes que a continuación se detallan(Kotsiantis et al. 2009):

- **Conceptos:** ideas básicas que se intentan formalizar. Pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, entre otros.
- **Relaciones:** representan la interacción y enlace entre los conceptos de dominio. Suelen formar la taxonomía de dominio: “subclase de”, “parte de”, “parte exhaustiva de” y “conectado a”.
- **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. Por ejemplo, pueden aparecer funciones como categorizar clase, asignar fecha.
- **Axiomas:** proposiciones que se declaran sobre las relaciones que deben cumplir los elementos de la ontología. Los axiomas, junto a la herencia de conceptos, permiten inferir el conocimiento oculto detrás de una taxonomía de conceptos.
- **Instancias:** se utilizan para representar objetos que pertenecen a un concepto determinado.

Para la realización de una ontología con los componentes antes expuestos es necesario realizar una secuencia de actividades y describirlas en un lenguaje para su desarrollo.

1.3.1Lenguajes para desarrollar ontologías

Existen numerosos lenguajes para el desarrollo de ontologías a continuación se presentan algunos de ellos analizados por (Ramos y Núñez 2007):

- **XML (eXtensible Markup Language)** es una meta-lenguaje derivado de SGML, que permite la definición de lenguajes de marcado adecuados para usos específicos. Es un lenguaje basado en marcas tipo etiquetas. XML es una manera flexible de crear formatos de información comunes y compartir tanto los formatos como los datos entre sistemas de computación. XML es un estándar W3C (World Wide Web Consortium).
- **RDF (Resource Description Framework)** RDF es una recomendación de la W3C para representar metadatos en la Web. Proporciona un medio para agregar semántica a un documento sin referirse a su estructura. RDF es una infraestructura para la codificación, intercambio y reutilización de metadatos estructurados.
- **XOL (Ontology Exchange Language)** Es un lenguaje para el intercambio de ontologías. Fue diseñado para ser utilizado como un lenguaje intermedio para permitir la transferencia de las

ontologías entre diferentes sistemas de bases de datos, herramientas de desarrollo de ontologías o aplicaciones. Aunque XOL fue diseñado para el intercambio de ontologías bioinformáticas, puede ser utilizado para ontologías de cualquier dominio.

- **OIL (Ontology Interface Layer)**Es un lenguaje de representación de ontologías basado en Web y capas de inferencia, que combina las primitivas de representación de conocimiento de los lenguajes basados en marcos con la semántica formal y los servicios de razonamiento proporcionados por la lógica descriptiva. Incluye una semántica precisa para describir el significado de los términos.
- **DAML+OIL**Es un lenguaje de marcado semántico para recursos Web. Es un estándar propuesto por la W3C para la representación de ontologías y metadatos. DAML (DARPA Agent Markup Language) fue transformado a DAML+OIL a través de la inclusión de algunas características de OIL al lenguaje. DAML consiste en un formalismo que permite a los agentes de software interactuar entre ellos.

  DAML+OIL se construye sobre RDF y RDFS, pero proporciona primitivas de representación más ricas, comúnmente encontradas en la lógica descriptiva. Algunas de las ideas basadas en marcos proporcionadas por OIL fueron eliminadas.

  DAML+OIL soporta tipos de datos complejos, a diferencia de OIL que sólo soporta el tipo de dato string, los tipos de datos utilizados provienen de XML Schema. El lenguaje tiene una semántica bien definida.
- **OWL (Web Ontology Language)** Es un lenguaje de marcado semántico desarrollado por la W3C para publicar y compartir ontologías sobre el World Wide Web. Es una extensión del vocabulario de RDF y se deriva de DAML+OIL. OWL está diseñado para ser utilizado por aplicaciones que necesitan procesar el contenido de la información en lugar de sólo presentarla a las personas. OWL proporciona tres sub-lenguajes diseñados para ser utilizados por comunidades específicas de desarrolladores y usuarios. La característica que define a cada lenguaje es su expresividad.

Para la creación de la ontología pertinente a nuestra investigación se consideró utilizar OWL ya que permite publicar y compartir datos mediante el uso de ontologías y emplea RDF. Además de ser un lenguaje diseñado para representar conocimiento complejo acerca de cosas, grupos de cosas y relaciones entre las cosas.

1.3.2 Metodologías para el desarrollo de ontologías

Para el diseño de cualquier ontología es necesario contar con una metodología específica. Son muchas las propuestas existentes. De entre ellas podemos destacar algunas como:

- **CYC** publicada por Lenat y Guha desde 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(Luna, Bonilla y Torres 2012).
- **USCHOLD Y KING** creada en 1995. Sostiene 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: (1) identificar el propósito; (2) capturar los conceptos y relaciones entre estos conceptos y los términos utilizados para referirse a estos conceptos y relaciones; (3) codificar la ontología. La ontología debe ser documentada y evaluada, y se pueden usar otras ontologías para crear la nueva(Luna, Bonilla y Torres 2012).
- **GRÜNINGER Y FOX c**uyo 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(Luna, Bonilla y Torres 2012).
- **METHONTOLOGY** esta es una de las propuestas más completas 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, la documentación, etc. 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, implementación, etc.), 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: (1) especificación, (2) conceptualización, (3) formalización, (4) implementación y (5) mantenimiento(Luna, Bonilla y Torres 2012).

- **SENSUS** esta constituye un enfoque top-down para derivar ontologías específicas del dominio a partir de grandes ontologías. En esta metodología se identifican un conjunto de términos semilla que son relevantes en un dominio particular. Tales términos se enlazan manualmente a una ontología de amplia cobertura. Los usuarios seleccionan automáticamente los términos relevantes para describir el dominio y acotar la ontología Sensus. Consecuentemente, el algoritmo devuelve el conjunto de términos estructurados jerárquicamente para describir un dominio, que puede ser usado como esqueleto para la base de conocimiento(Luna, Bonilla y Torres 2012).
- **ON-TOKNOWLEDGE** esta aplica ontologías a la información disponible electrónicamente para mejorar la calidad de la gestión de conocimiento en organizaciones grandes y distribuidas. Además, incluye la identificación de metas que deberían ser conseguidas por herramientas de gestión de conocimiento y está basada en el análisis de escenarios de uso y en los diferentes papeles desempeñados por trabajadores de conocimiento y accionistas en las organizaciones(Luna, Bonilla y Torres 2012).

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.

1.3.3Herramientas para la creación de ontologías

Una vez que se haya seleccionado la metodología y el lenguaje para representar la ontología que se desee desarrollar, es necesario utilizar una herramienta que facilite a los desarrolladores o usuarios la creación, uso y mantenimiento de las mismas. Por tal motivo se han desarrollado un gran número de herramientas al respecto. Estas herramientas se pueden clasificar atendiendo a dos criterios: el mecanismo de representación en el que están basadas, o la finalidad que tengan las mismas. Sobre el primer criterio es posible encontrar *(Díaz Díaz 2009)*:

- Herramientas que se corresponden o representan directamente utilizando un lenguaje de representación de ontologías (OilEd, RdfEdit, SWOOP, etc.)
- Herramientas integradas, cuya característica principal es tener una arquitectura extensible y su representación del conocimiento es independiente de cualquier lenguaje (Protégé, WebOde, OntoEdit, etc.).

Dependiendo de la finalidad que tengan estas herramientas:

- Herramientas de desarrollo o representación de ontologías.
- Herramientas de evaluación de ontologías.
- Herramientas de integración y alineamiento de ontologías.
- Herramientas para anotar ontologías.
- Herramientas de consulta e inferencia.
- Herramientas de evolución y versionado.
- Herramientas de generación de ontologías (learning tools).
- Herramientas de integración con bases de datos.

Las herramientas para el desarrollo de ontologías, permiten la creación, uso y mantenimiento de las mismas siguiendo el ciclo de vida de desarrollo. Estas herramientas, se caracterizarán por la capacidad para ajustarse a una metodología concreta, o bien seguir cada una de las opciones del ciclo de vida de una ontología, desde la definición de la ontología, creación de instancias, traducción a un lenguaje de representación formal y mantenimiento. La mayoría de estas herramientas se basan en el formalismo de representación de marcos (frames). Entre las herramientas de representación de ontologías más conocidas, están aquellas que únicamente permiten editar o definir ontologías en un lenguaje determinado, pero también existen herramientas que permiten la construcción de ontologías, guiando a los usuarios en el proceso de desarrollo*(Díaz Díaz 2009)*. A continuación, se caracterizan algunas de estas herramientas:

- **WebODE:** es una herramienta para modelar el conocimiento usando ontologías. Fue desarrollada en la Escuela Técnica de Informática (FI) en Madrid y es el equivalente Web de ODE (Ontology Desing Environment). Su interfaz basada en formularios y editores gráficos permite guiar la conceptualización y editar taxonomías. Su arquitectura ofrece chequeo completo de la consistencia e importación avanzada de términos *(Barrera, Núñez y Ramos 2012)*. Esta herramienta no cuenta

con soporte desde el año 2006, elemento que limita el uso de la misma en la presente investigación.

- **Swoop:** proyecto de código abierto desarrollado en el laboratorio de Mindswap de la Universidad de Meryland. Se trata de una herramienta para la creación, edición y depuración de ontologías OWL, basado en conceptos de diseño y navegación hipermedia. Su aspecto es similar a un navegador web, pudiendo recorrer los elementos de las diferentes ontologías cargadas a base de activar hipervínculos. Esta herramienta admite varias sintaxis de representación, desde OWL a la sintaxis habitual RDF/XML y permite la búsqueda semántica. Utiliza anotación colaborativa mediante el entorno de trabajo Annotea. Es fácilmente extensible por herramientas de anotación semántica tipo SMORE; además incluye el razonador Pellet integrado, con la posibilidad de depurar la ontología al tiempo que se está creando*(Horridge et al. 2014)*. No permite el empleo de otros razonadores que no sea el Pellet, limitando la comprobación de las condiciones lógicas formales con diferentes razonadores.
- **WebOnto:** desarrollado por el Knowledge Media Institute (KMI) de la Open University (Reino Unido). Aunque es de libre acceso, para acceder y disfrutar de sus potencialidades de edición, los usuarios deben solicitar una cuenta a los administradores. Es un applet (pequeña aplicación que permite obtener una gran variedad de efectos en las páginas web) de Java, trabaja junto a un servidor web personalizado que permite a los usuarios navegar, crear y editar ontologías sin problemas de interfaz. WebOnto ofrece la gestión gráfica de ontologías e inspección de elementos, teniendo en cuenta la herencia de propiedades y el chequeo de consistencia; además, permite la generación automática de instancias a partir de definiciones de clases *(Barrera, Núñez y Ramos 2012)*.Solo cuenta con soporte para el lenguaje OCML, lo que limita su empleo como herramienta para el desarrollo de la ontología para gestionar el conocimiento en las pruebas de usabilidad, que será desarrollada con OWL como se mencionó anteriormente.
- **Protégé:** es un software libre de código abierto implementado en Java, desarrollado en la Universidad de Stamford, que permite la construcción de ontologías de dominio. Es capaz de operar como una plataforma para acceder a otros sistemas basados en conocimiento o aplicaciones integradas, o como una librería que puede ser usada por otras aplicaciones para acceder y visualizar bases de conocimiento. La herramienta ofrece una interfaz gráfica que permite al desarrollador de ontologías enfocarse en la modelación conceptual sin que requiera de conocimientos de la sintaxis de los lenguajes de salida *(Ramos y Núñez 2007)*. Es reconocida como una de las herramientas

más utilizadas para la ingeniería ontológica *(Rodríguez y Martínez 2017)*, *(Zhao, Zhang y Zhao 2012)*y existen varios estudios que así lo avalan *(Silega et al. 2017)*, *(Horridge et al. 2014)*. Cuenta con soporte para OWL y permite integrar a su solución varios razonadores como el Pellet, HermiT y Fact++

- **OntoEdit:** Herramienta de edición de ontologías que apoya el desarrollo y mantenimiento de las mismas, utilizando medios gráficos en un entorno Web. La herramienta representa gráficamente las ontologías, las almacena y posteriormente manipula en una base de datos relacional. Su interfaz permite incorporar plugins y exporta a la mayoría de lenguajes de ontologías. Está basado en la metodología ONTo-Knowledge*(Díaz Díaz 2009)*.
- **KAON:** Editor de ontologías que permite la creación y mantenimiento a través de un navegador Web. Sólo representa RDF(s) y almacena la información en una base de datos*(Oberle et al. 2004)*.
- **OilEd:** Editor de ontologías que usa el lenguaje OIL (precursor del OWL). No es una herramienta que permita desarrollar todo el proceso de vida de desarrollo de una ontología, pero permite definir axiomas, importar y exportar a la mayoría de los lenguajes conocidos*(Díaz Díaz 2009)*.

Teniendo en cuenta los elementos planteados anteriormente se decidió utilizar en la presente investigación como herramienta para el desarrollo de ontologías el Protégé ya que funciona como una aplicación local o a través de un cliente en una comunicación con un servidor remoto. El navegador web de Protégé permite a los usuarios compartir, navegar y editar sus ontologías utilizando un navegador web estándar, lo que proporciona un ambiente de colaboración que ayuda a las comunidades en el desarrollo de ontologías. La comunidad de usuarios de Protégé regularmente contribuye a mejorar la calidad del software y participa en grupos de discusión en línea dedicados a formular preguntas, realizar peticiones de nuevas características y cuestiones de soporte técnico. Protégé presenta una gran capacidad de extensión, debido al soporte de conectores (plugins), que son aditivos que se adquieren de manera individual y se acoplan al entorno de trabajo de Protégé para añadirle funcionalidad.

1.3.4 Razonadores:

Para extraer más conocimiento que los conceptos almacenados en una ontología, son necesarias otras herramientas como un razonador, los cuales permiten deducir nuevas relaciones o conceptos no explícitos en el modelo original de la ontología, permitiendo además realizar consultas sobre ella(Barrera, Núñez y

Ramos 2012). Existen una amplia variedad de razonadores para los diferentes lenguajes de ontologías. Muchos de estos de forma libre. A continuación se describen los razonadores más representativos e importantes disponibles basado en OWL y extensiones del mismo(Reuco 2008).

Jena

Jena fue originalmente una API de Java para RDF que ahora también soporta OWL. Proporciona los métodos para la entrada y salida de modelos OWL de y para archivos, bases de datos o la Web. Jena contiene interfaces para representar modelos, recursos, propiedades, literales, sentencias y todos los demás conceptos claves de RDF y OWL. Además, Jena provee los métodos para navegar y consultar modelos OWL. También define algunas operaciones básicas sobre modelos como la unión, intersección y diferencia de dos modelos.

Jena suministra un juego de razonadores para hacer inferencias y validaciones de modelos OWL:

- Razonador transitivo: suministra las reglas de inferencias básicas para determinar la jerarquía de clase y propiedades, con el objetivo de mejorar el rendimiento y disminuir el consumo de espacio. Es usado en el razonador de reglas RDFS para llevar a cabo la clausura transitiva de propiedades rdfs: SubClassOf y rdfs: subPropertyOf.
- Razonador de reglas RDFS: tiene tres modos distintos que el usuario puede seleccionar: Completamente (Full), Predeterminado (Default) y Simple.
- 9 Razonador de OWL FB: implementa un subconjunto de inferencia OWL-Lite.

FaCT++

FaCT++ es un razonador basado en C++, libre y de código abierto, implementa procedimiento de decisión tableaux para la lógica descriptiva SHOIQ, con soporte adicional para tipos de datos como cadenas de caracteres y enteros. El sistema emplea un avanzado desempeño de optimización en las técnicas estándares absorción y combinación de modelos y otros como heurísticas de ordenamiento y clasificación taxonómica. FaCT++ a través de interfaces DIG (DL Implementation Group) puede utilizarse para servicios de razonamiento para herramientas de diseño de ontologías que soporten el lenguaje OWL-DL.

Para el razonamiento, FaCT++ primeramente realiza un preprocesamiento sobre la ontología cargada en el razonador, es normalizada y transformada en una representación interna. Durante este proceso algunas optimizaciones son aplicadas, así como clasificaciones.

RacerPro

RacerPro (Renamed ABox and Concept Expression Reasoner Professional) es un razonador basado en LISP, es comercial pero con trials gratuitos y licencias de investigación. Como el nombre indica, sus orígenes están dentro del área de la lógica descriptiva aunque puede ser usado para lógicas modales. Aborda lenguajes estandarizados de ontologías.

RacerPro puede procesar ontologías en OWL y en los sublenguajes Lite y DL, pero con algunas restricciones. También los tipos de datos definidos por el usuario no son soportados, realiza chequeo de consistencia. Implementa procesos de optimización e incluye razonamiento algebraico.

RacerPro implementa el quasi estándar DIG basado en http para la interconexión de sistemas DL con interfaces y aplicaciones usando un protocolo basado en XML. Es como una combinación de lógicas descriptivas y algebras relacionales específicas.

Pellet

Pellet, desarrollado en la Universidad de Maryland, comenzó como una demostración para ayudar a comprender la implementación de los requerimientos para OWL. Se ha convertido desde entonces en una herramienta de alto desempeño, completa, práctica y popular para el razonamiento con OWL. Pellet ha sido el primer razonador que soporta todo el OWL-DL. Pellet es implementado en Java, es un software libre y de código abierto. Brinda grandes facilidades incluyendo tipos de datos definidos por usuarios, respuestas a consultas conjuntivas, soporta reglas, razonamiento con múltiples ontologías usando E-Connections, trabajo con axiomas, técnicas de optimización para DL, depuración, entre otras. Para hacer fácilmente asequible para los usuarios estas capacidades de razonamiento, Pellet proporciona varias interfaces incluyendo una interfaz de línea de comando, un formulario Web interactivo, implementación en servidor DIG, aplicación independiente, integración con editores de ontologías como Protégé y otras.

Para la realización de inferencias se propuso como razonador a utilizar Pellet ya que este posee múltiples ventajas pues esta implementado en Java, lo cual permitió hacer inferencias y consultas basadas en onto-

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

1.4 Reutilización de ontologías

Durante el proceso de desarrollo de la ontología es importante tener en cuenta la posibilidad de la reutilización de alguna ontología o cualquier otra fuente de información ya existente 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 (Echeverría Pérez y Fernández Pérez 2013),(Castañeda Martínez et al. 2018) y(Martínez Valdivieso 2020) 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 pruebas de usabilidad 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 anteriormente planteadas, se decidió desarrollar la ontología desde cero y no reutilizar algunas de las propuestas ya reflejadas.

1.5 Conclusiones del capítulo

- Las pruebas de Usabilidad aportan ventajas significativas respecto a mejorar el producto y la satisfacción de sus clientes. Además nos permiten conocer si el producto es de fácil comprensión y aprendizaje a la hora de ser utilizado por usuarios inexpertos en Informática y nos permiten conocer la aceptación que tendrá el producto de software.
- Para la representación y reutilización del conocimiento que se genera de los procesos de pruebas de usabilidad es necesario apoyarse en la gestión del conocimiento, los sistemas de organización del conocimiento, específicamente las ontologías de dominio.
- Se identificó que existe consenso en la comunidad científica en cuanto al empleo de OWL como lenguaje para la representación y Protégé como herramienta para la creación; así como

el uso de Methontology como metodología para guiar el desarrollo de la ontología de la presente investigación.

- Las ontologías existentes aplicadas a las pruebas de software estudiadas poseen alto valor desde el punto de vista teórico pero son incompletas, generalizadoras de este dominio, no abordan todos los términos o las relaciones surgidas en la Dirección de Calidad, específicamente las que abarcan las pruebas de usabilidad.

CAPÍTULO II: Diseño de la propuesta ontológica

En este capítulo se realiza el diseño e implementación de la propuesta ontológica, haciendo uso de las herramientas descritas en el capítulo anterior. Se siguen los pasos de la metodología Methontology hasta lograr la implementación de la ontología. Se especifican los requisitos y se elabora el modelo conceptual además de implementar la solución con el uso de los lenguajes y las herramientas seleccionadas.

2.1 Descripción de la propuesta de solución

La propuesta ontológica pretende representar y reutilizar el conocimiento existente y generado en el proceso de pruebas de usabilidad de la Dirección de la Calidad de Software y lograr con esto un mayor nivel de comprensión de dicho proceso por las personas involucradas en él. 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 de los datos del dominio en cuestión. Además, permitirá representar el conocimiento a tratar dentro del mismo. A continuación, se muestra una breve descripción con las principales clases y relaciones del proceso de evaluación de usabilidad:

En el proceso de pruebas de usabilidad interviene el producto a evaluar y se realiza sobre la base de un modelo o estándar de calidad, el cual descompone la usabilidad en subcaracterísticas de usabilidad, las cuales se ven cuantificadas por las medidas de usabilidad. En este proceso además se involucran especialistas, estos coordinan artefactos y en este accionar manipulan herramientas y consultan documentación. En las evaluaciones de usabilidad existen tres tipos de pruebas como las automatizadas, con usuarios reales o con usuarios experto las cuales utilizan técnicas de evaluaciones específicos. 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 y se hace 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.

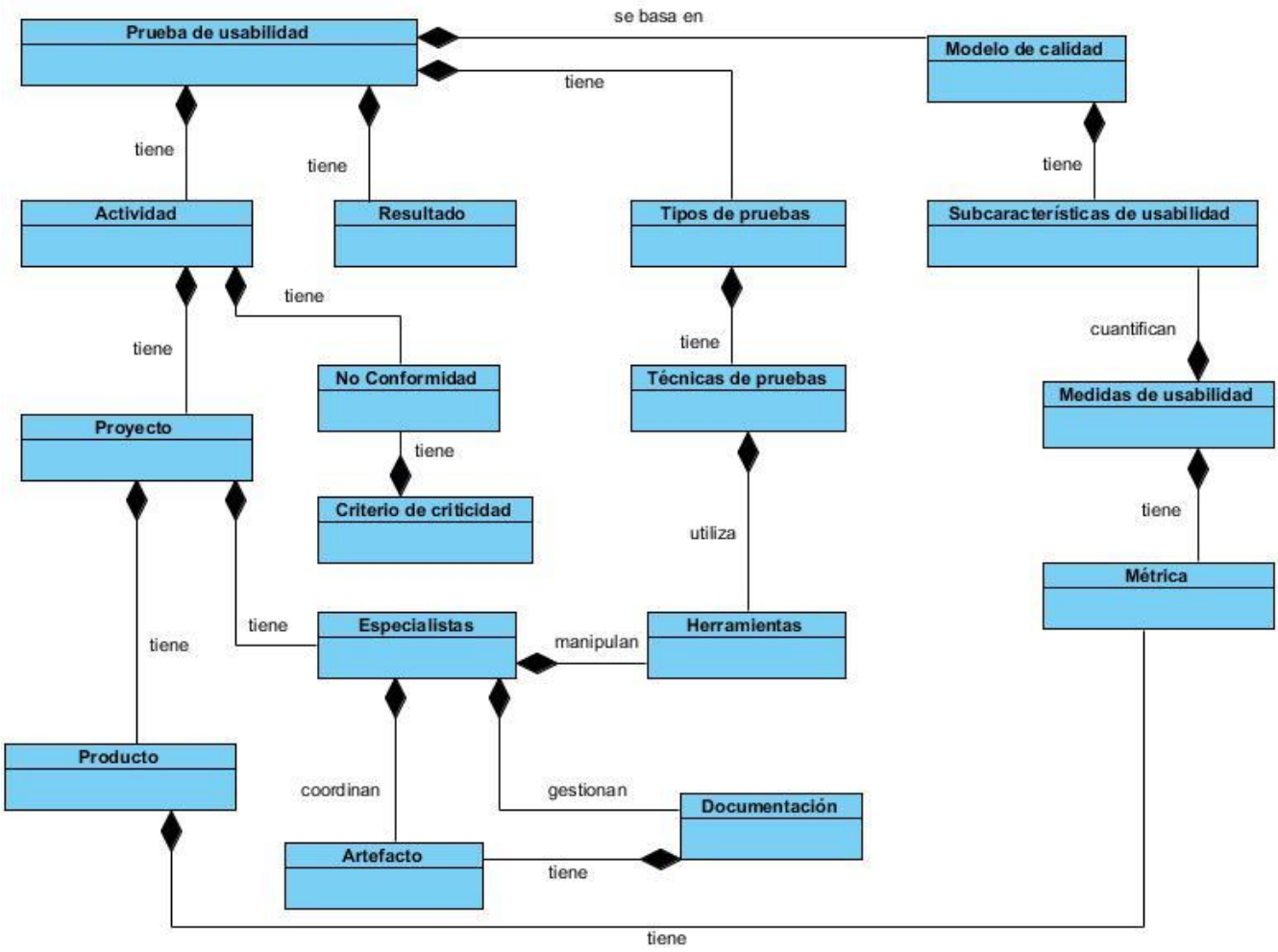

Figura 1: Modelo conceptual del Proceso de prueba de usabilidad. Fuente: Elaboración propia.

2.2 Aplicación de la metodología Methontology

Como se menciona en los fundamentos teóricos trabajaremos sobre las actividades de la metodología Methontology. A continuación se presentan los resultados de ejecutar las actividades de Especificación, Conceptualización, Formalización e Implementación.

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. En el Tabla 1 muestra dicha plantilla para esta ontología.

Tabla 1: Especificación de Requerimientos.

Dominio	Proceso de Pruebas de usabilidad en la Dirección de la Calidad UCI.
Fecha	5 de Agosto del 2021
Desarrollador(es)	Ivan Gutierrez del Toro
Propósito	Construir una ontología con el objetivo de proporcionar una mejor organización, representación y recuperación del conocimiento existente y generado en las pruebas de usabilidad en la Dirección de la Calidad UCI.
Nivel de Formalidad	Formal
Alcance (preguntas de competencia)	¿Qué Especialista manipula la herramienta1? ¿Qué técnicas de pruebas utiliza un determinado tipo de prueba? ¿Qué Especialista gestiona la documentación 1, 2 y 3? ¿Qué actividad presenta no conformidad? ¿Qué documentación requiere un determinado artefacto de proyecto para ser analizado?
Fuentes de Conocimiento	Expertos de la Dirección de la Calidad de Software de la UCI.
Uso	La ontología se empleará con el fin de facilitar el desarrollo del proceso de prueba de usabilidad, además formalizar todo el conocimiento y descubrir nuevos a partir de los ya existentes. Esto permitirá encontrar inconsistencia y razonar sobre lo formalizado.
Usuarios finales	Administrador de la calidad, Administrador de la Configuración, Asesor de calidad, Coordinador de calidad, Coordinador de Prueba, Equipo de Proyecto, Jefe de proyecto, Probador Los que se deben hacer responsables de su mantenimiento deben ser expertos en el tema, así como los informáticos de la entidad capacitados en el trabajo con las ontologías, los cuales deben llevar a cabo la administración de la misma.

2.2.2 Conceptualización

La actividad número dos correspondiente a la metodología Methontology es la conceptualización la cual se basa en organizar y convertir una percepción informal del dominio en una especificación semi-formal que pueden ser fácilmente comprendidas. Esta actividad define diez 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, se define cada término relevante del dominio incluyendo los conceptos, instancias y relaciones. En la tabla 2 se muestran cuatro ejemplos, en el Anexo 1 se encuentran todos los términos que tiene la ontología.

Tabla 2: Glosario de términos.

No.	Término	Descripción	Tipo
1.	Tipo de Prueba	Clasificación de las pruebas.	Concepto
2.	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
3.	Producto	Un conjunto de programas informáticos, procedimientos y posiblemente documentación y datos asociados.	Concepto
4.	Técnica de evaluación	Métodos y habilidades necesarios para llevar a cabo una actividad específica.	Concepto

Tarea 2: Construir la taxonomía de conceptos.

Una vez que el glosario de términos contiene suficientes términos, el desarrollador de la ontología construye las taxonomías de conceptos que definen su jerarquía. Para construir taxonomías de conceptos, se seleccionan del glosario de términos aquellos términos que son conceptos(Corcho et al. 2005).

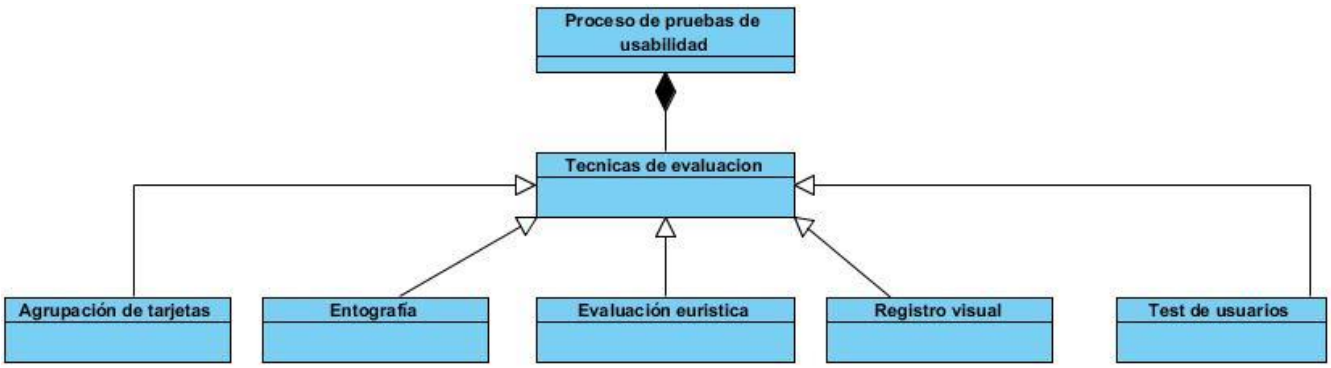

Figura 2: Taxonomía de conceptos. Fuente: Elaboración propia.

Tarea 3: Construir un diagrama de relaciones binarias.

Una vez construida la taxonomía, el siguiente procedimiento será la construcción de diagramas de relaciones binarias. El objetivo de estos diagramas es definir las relaciones existentes entre conceptos de una o más taxonomías. En la figura # se muestra una relación binaria inversa de uno de los conceptos del diagrama, Tipos de prueba y Métodos de evaluación, el resto de relaciones binarias se encuentra en el Anexo 1

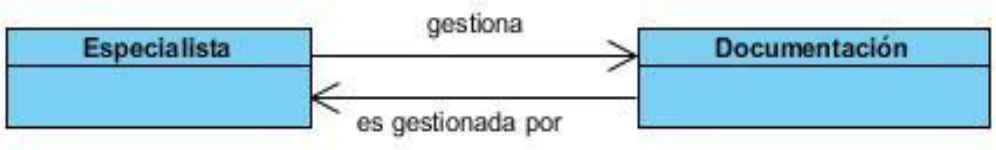

Figura 3: Diagrama de relaciones binarias. Fuente: Elaboración propia.

Tarea 4: Construir el diccionario de conceptos.

Una vez generadas las taxonomías de conceptos y los diagramas de relaciones binarias, se debe especificar cuáles son las propiedades que describen cada concepto de la taxonomía, así como las relaciones identificadas en el diagrama del paso anterior y las instancias de cada uno de los conceptos. El diccionario de conceptos contiene todos los conceptos del dominio, sus relaciones, sus instancias, y sus atributos de clase y de instancia. Las relaciones especificadas para cada concepto son aquellas en las que el concepto es el origen de la misma. Las relaciones, atributos de instancias, y atributos de clases son locales al concepto, lo que significa que sus nombres pueden repetirse en diferentes conceptos. En la tabla 3se muestra un ejemplo en el Anexo 1 se encuentran los conceptos restantes.

Tabla 3: Diccionario de conceptos

Nombre del concepto	**Instancia**	**Atributos de clase**	**Atributos de instancia**	**Relaciones**

Artefacto	Artefacto1		denominación descripción	tiene documentación

Tarea 5: Describir las relaciones binarias en detalle.

Luego de la construcción del diccionario de concepto se procede a definir las relaciones binarias en detalle, con el objetivo de detallar todas las relaciones binarias descritas en los diagramas de relaciones binarias. 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 puede ver un ejemplo, en el Anexo 1 se encuentran las relaciones restantes.

Tabla 4: Relación binaria explicada

Nombre de la relación	**Concepto Origen**	**Cardinal dad**	**Concepto Destino**	**Relación Inversa**
tiene métrica	Producto	1…n	Métrica	es métrica de
utiliza técnica de prueba	Tipo de prueba	1…n	Técnica de prueba	es utilizado por
gestiona documentación	Especialista	1…n	Documentación	es gestionado por
coordina artefacto	Especialista	1…n	Artefacto	es coordinado por

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 atributo de instancia, 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. Se puede apreciar un ejemplo en la tabla 5, en el Anexo 1 se encuentran los atributos restantes.

Tabla 5: Descripción de los atributos de instancias.

Atributo de instancia	Concepto	Tipo de valor	Rango	Cardina-lidad
Nombre	Especialista	string	-	1…1
fecha Inicio	Actividad	string	-	1…1
fecha fin	Actividad	string	-	1…1

Tarea 7: Describir atributos de clases

Esta tarea tiene como objetivo describir todos los atributos de clase que fueron especificados en el diccionario de conceptos. Los cuales representan características genéricas de un concepto, detalladamente, quiere decir que todas las instancias del concepto tendrán el mismo valor para ese atributo. 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. 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.

En la actividad corresponde definir reglas, se le denominan axiomas formales o reglas a expresiones lógicas que son siempre verdaderas y utilizadas normalmente para especificar restricciones en la ontología. 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. En el presente trabajo los conceptos reglas y axiomas

se solapan. Es decir, tienen el mismo significado. Un ejemplo se muestra en la tabla 6, en el Anexo1 se encuentran los axiomas restantes.

Tabla 6: Ejemplo de reglas o axiomas.

Nombre de la regla	**Descripción**	**Expresión**	**Conceptos**	**Relaciones**
Especialista Probador	Especialista Probador gestiona alguna Documentación.	Especialista and (gestiona documentación some Caso de prueba)	Probador Caso de prueba	gestiona documentación

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 (instancia), nombre del concepto al que pertenece y valores de los atributos de instancias, si se conocen.

Tabla 7: Descripción de instancias.

Instancia	**Nombre del concepto**	**Atributos de instancia(Data property)**	**Valor de los atributos de instancia**
Doc1	Documentación	denominación	--
		descripción	--

2.2.3 Formalización

En esta actividad se transformó el modelo conceptual elaborado en las actividades anteriores a un modelo formal o semi-computable. Para ello se hizo uso del editor de ontologías Protégé 5.2.

Creación de las clases

Mediante la utilización de la pestaña Entities del Protégé se representaron cada uno de los conceptos identificados en el modelo conceptual. Para representar los conceptos resulta importante especificar el nombre de la clase, la clase padre y las clases disjuntas. En la Figura # se muestra la definición de las clases de la ontología desarrollada.

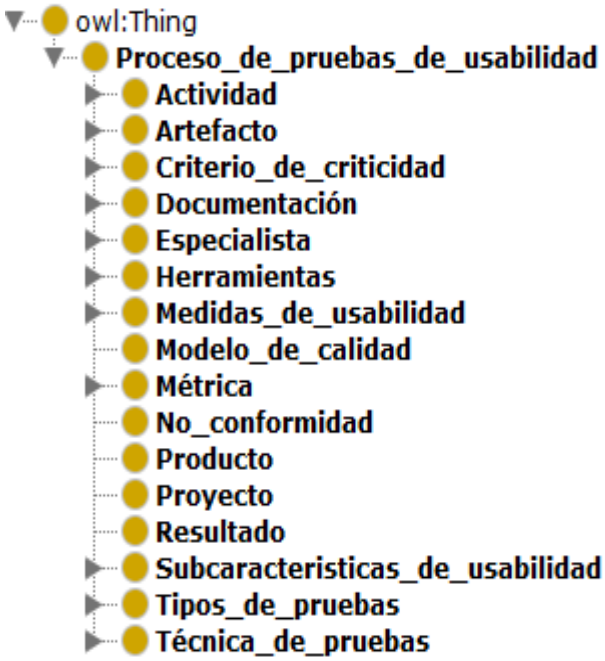

Figura 4: Definición de las clases creadas en Protégé. Fuente: Elaboración Propia

Se identificaron 88 entidades como parte de la ontología desarrollada, es válido aclarar que la flexibilidad del método permite que la misma sea actualizada en caso de que aparezcan nuevas tendencias en el diseño de software.

Creación de las propiedades de las clases

Luego de ser identificadas cada una de las entidades de la ontología y modeladas en la herramienta se establecieron las relaciones semánticas entre cada una de ellas mediante el uso de propiedades de objetos (Object Properties). Fueron representadas un total de 33 relaciones entre las clases.

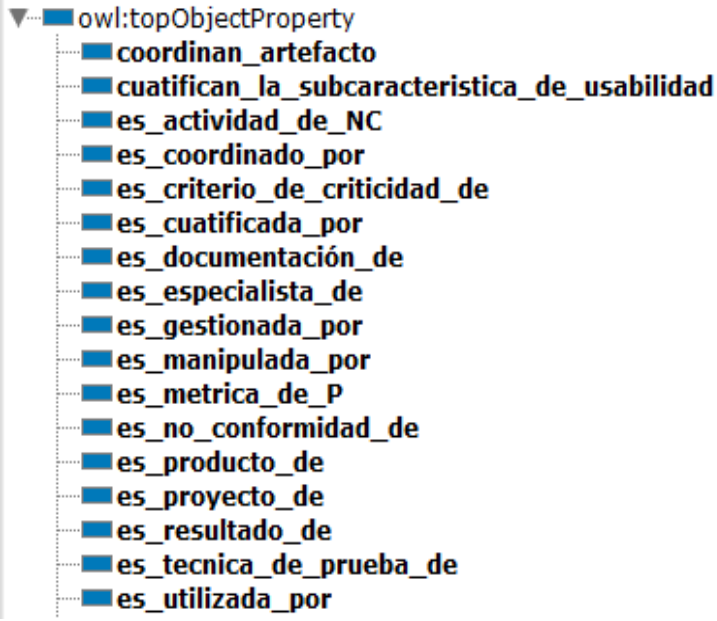

Figura 5: Object Properties Propiedades de las clases creadas en Protégé. Fuente: Elaboración propia.

Definición de atributos

Con el objetivo de describir cada uno de los conceptos fueron representados en la ontología los atributos presentes en el modelo conceptual. Para representar los mismos fue empleada la pestaña Data Properties de la herramienta Protégé. Es importante representar además del nombre, el dominio y el rango a la hora de agregar un nuevo atributo para garantizar la consistencia de los datos de la ontología. En la Figura 6, se muestra una representación de los atributos presentes en la ontología.

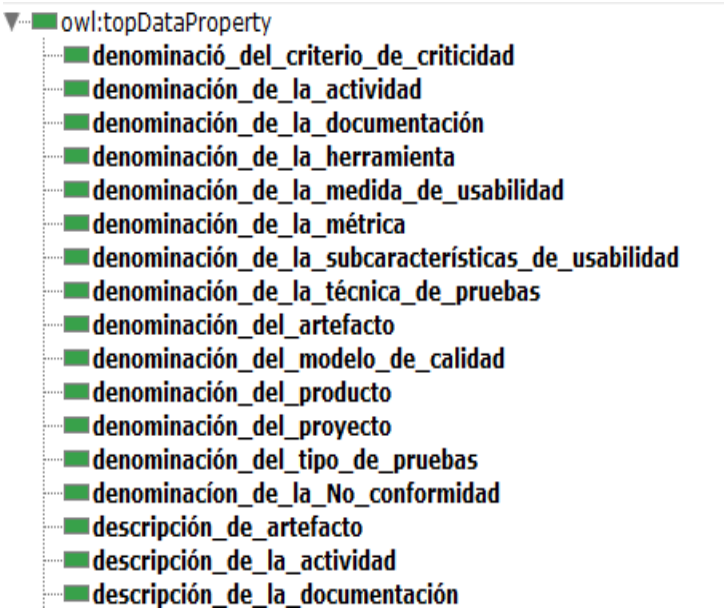

Figura 6: Creación de la Data Properties en Protégé. Fuente: Elaboración propia.

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. 7se puede observar las reglas descritas para la clase Probador.

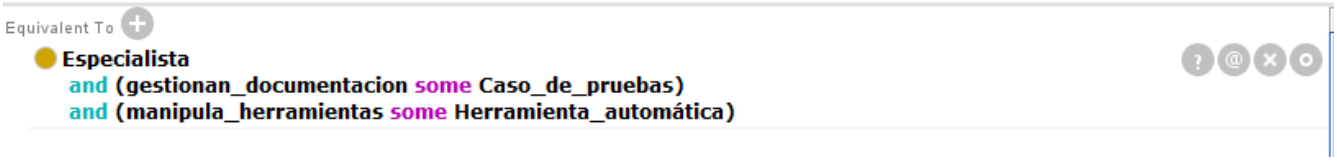

Figura 7: Creación de las reglas. Fuente: Elaboración propia.

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:

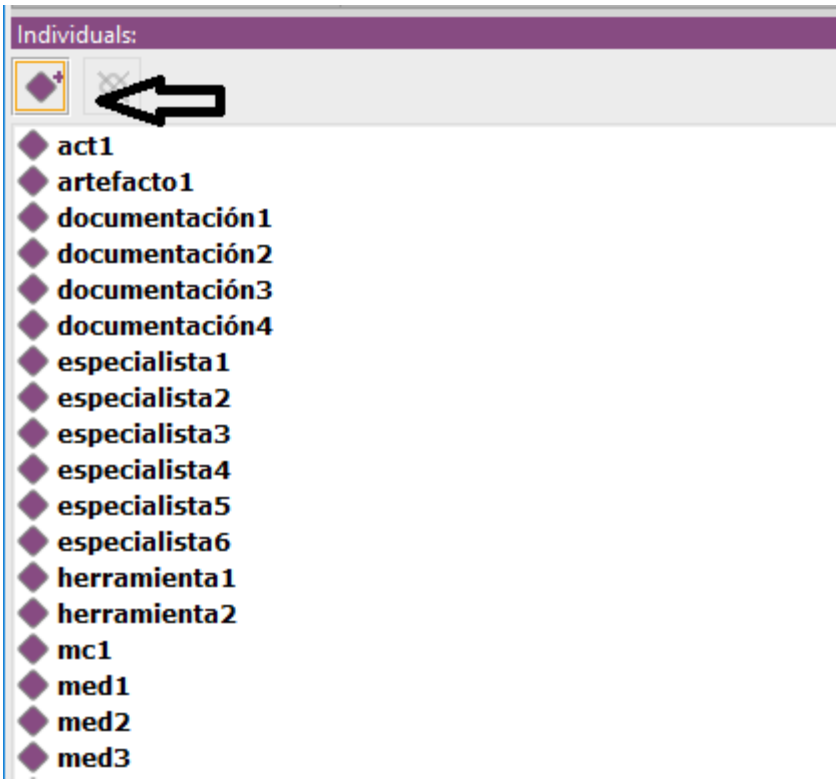

Figura 8: Creación de las instancias con Protégé. Fuente: Elaboración propia.

2.2.4 Implementación

Esta actividad consistió en codificar el modelo generado, en un lenguaje ontológico. Al definir dicha ontología, se genera la codificación en el lenguaje ontológico OWL.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. 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.(Ver tabla 8):

Tabla 8: Declaración de las clases.

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

```
</Declaration>
```

De una forma similar se realiza la declaración de una subclase de otra clase, mostrando que la clase Asesor Tecnológico es subclase de Especialista. (Ver tabla 9):

Tabla 9: Declaración de una subclase de otra clase

```
<SubClassOf>
<Class IRI="#Asesor_Tecnologico"/>
<Class IRI="#Especialista"/>
</SubClassOf>
```

.

La declaración de las relaciones se realiza de la siguiente manera. (Ver tabla 10):

Tabla 10: Declaración de las relaciones.

```
<Declaration>
<ObjectProperty
IRI=”#tiene_especialistas"/>
</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 especialista1, el cual es un individuo de la clase Especialistas (Ver tabla 11).

Tabla 11: Creación de los individuos de las clases.

```
<Declaration>
<NamedIndividual IRI="#especialista1"/>
</Declaration>
```

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 2010). 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, Mariño y Godoy 2012)

Esta actividad fue empleada en todo el proceso de 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 mejora.

2.3 Conclusiones Parciales

Luego de desarrollado el presente capítulo anterior se arribaron a las siguientes conclusiones:

- La ontología de soporte al proceso de pruebas de usabilidad que se obtiene permite representar y reutilizar el conocimiento existente y generado, lo que favorece la gestión del conocimiento en el proceso de pruebas de usabilidad.

- Se pudo caracterizar el proceso de evaluación de producto de software determinando los principales conceptos y sus relaciones.

- Implementar la ontología, con el empleo de la herramienta protege y el lenguaje OWL, permite que pueda ser usada en varias plataformas, la reutilización de la ontología en futuras investigaciones y en organizaciones similares, lo que facilita su uso y mantenimiento.

CAPÍTULO III: Validación y prueba de la ontología

En el presente capítulo se presentan los resultados de la evaluación de la ontología propuesta, 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 identificadas. Se valida la capacidad de inferencia de la ontología.

3.1 Validación de la ontología

Para las ontologías es importante comprobar según la metodología propuesta(Rodríguez y Martínez 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 Martínez 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 corpus.

Según Ramos y otros(Ramos 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 Fase1. Uso correcto del lenguaje

Para evaluar el uso correcto de los lenguajes 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 prueba de usabilidad, logrando obtener una ortografía de óptima calidad.

3.1.2 Fase2. Exactitud de la estructura taxonómica

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

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 y existencia de conceptos repetidos.

3.1.3 Fase3. 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 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:

$$Precisión = \frac{CO\_C}{COnto} \qquad (1)$$

Donde:

- CO_C: Cantidad de términos que se solapan entre la ontologíay 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 88, al igual que la cantidad total de términos de la ontología.

$$Precisión = \frac{88}{88} = 1$$

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:

$$Exhaustividad = \frac{CO\_C}{CCorp} \qquad (2)$$

- $CCorp$: Cantidad total de términos del corpus, el cual su valor es 88.

$$Exhaustividad = \frac{88}{88} = 1$$

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

3.1.4 Fase4. 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 especialista gestiona la documentación1?

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 Especialista.

Clases	Instancias	Propiedades	Valor de las Propiedades
especialista	especialista3	gestiona documentación	Documentación3

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen la tprueba1 debe clasificarse en tipo de prueba funcional

Resultado obtenido: Satisfactorio.

De la misma manera, se aplicarán los casos de pruebas restantes (Ver anexo 3).

Durante el desarrollo de la investigación, el Ingeniero Dayamí Pompa Echavarríase 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:

- La ontología se evaluó en función de la organización y recuperación de la información del proceso de pruebas de usabilidad en la UCI, que demostró la utilización correcta del lenguaje 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 para dar validez de su alcance, capacidad de inferencia y recuperación de conocimiento.

CONCLUSIONES FINALES

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

- El estudio de 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.
- Se desarrolló una ontología siguiendo una metodología sólida lo que contribuyó a garantizar su calidad. La ontología incluye los conceptos más relevantes relacionados con las pruebas de usabilidad, lo que permite el análisis formal de la información registrada.
- La validación de la ontología corroboró que está correctamente diseñada, cumple con las especificaciones como sistema lógico formal y satisface los requisitos para los que fue diseñada.

RECOMENDACIONES

- Trabajar en la integración con otras ontologías que comparten conceptos de interés sobre las pruebas de software
- Poblar la ontología a partir de los datos existentes en la herramienta Gesprod mediante un proceso automatizado

REFERENCIAS BIBLIOGRÁFICAS

ALFONZO, P.L., MARIÑO, S.I. y GODOY, M.V., 2012. Propuesta de aplicación de SCRUM para gestionar el proceso de mantenimiento del software: estudio preliminar. *Técnica Administrativa*, vol. 11, no. 1, pp. 1666-1680.

BARRERA, M., NÚÑEZ, H. y RAMOS, E., 2012. INGENIERÍA ONTOLÓGICA. *Lecturas en Ciencias de la Computación*, ISSN 1316 - 6239.

BRITO RIVEROL, Y., CAPOTE GARCÍA, T. y FEBLES RODRÍGUEZ, J.P., 2015. *Estrategia para estructurar las evaluaciones de un laboratorio de pruebas de software desde la perspectiva de la acreditación*. Master's Thesis. S.l.: Universidad de las Ciencias Informáticas.

CARLOTA BUSTELO RUESTA y RAQUEL AMARILLA IGLESIAS, 2001. Gestión del conocimiento y gestión de la información | revista PH. [en línea], [Consulta: 24 julio 2021]. ISSN 2340-7565. Disponible en: http://www.iaph.es/revistaph/index.php/revistaph/article/view/1153.

CARRIÓN ESTRADET, L., 2018. *Ontología de apoyo al proceso de evaluación de productos de software en la UCI.* La Habana: Universidad de las Ciencias Informáticas.

CASTAÑEDA MARTÍNEZ, A., PARKER LEYVA, C., FERNÁNDEZ PÉREZ, Y. y LÓPEZ RODRÍGUEZ, Y.A., 2018. Ontología de apoyo a las pruebas de software en la UCI. *Revista Cubana de Ciencias Informáticas*, vol. 12, pp. 222-235.

CASTTRILLON, P. y EUCARIO, J., 2018. La gestión del conocimiento en la planificación y desarrollo de proyectos informáticos. *105* [en línea], [Consulta: 23 julio 2021]. ISSN 2227.1899. Disponible en: https://repositorio.uci.cu/jspui/handle/123456789/9446.

CHECA ROJAS, D. y ROJAS ALVARADO, O., 2014. ONTOLOGÍA PARA LOS SISTEMAS HOLÓNICOS DE MANUFACTURA BASADOS EN LA UNIDAD DE PRODUCCIÓN. *Revista Colombiana de Tecnologías de Avanzada*, vol. 1, no. 23. ISSN 1692-7257.

CORCHO, O., FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. y LÓPEZ-CIMA, A., 2005. Construcción de ontologías legales con la metodología METHONTOLOGY y la herramienta WebODE. *Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*, pp. 142-157.

CORRALES, Y.V., ABASCAL, A.V. y REAL, V.P., 2017. Procedimiento para pruebas de usabilidad en las aplicaciones informáticas basado en la ISO/IEC 25000 Procedure for usability testing in computer applications based on ISO/IEC 25000. , pp. 13-23. ISSN 2306-2495.

CORRALES, Y.V., ABASCAL, A.V. y REAL, V.P., [sin fecha]. Procedimiento para pruebas de usabilidad en las aplicaciones informáticas basado en la ISO/IEC 25000 Procedure for usability testing in computer applications based on ISO/IEC 25000. ,

DÍAZ DÍAZ, Y., 2009. *Estudio de las funcionalidades de Protégé para el soporte de ontologías de dominio* [en línea]. Thesis. S.l.: Universidad Central “Marta Abreu” de Las Villas. [Consulta: 28 julio 2021]. Disponible en: http://dspace.uclv.edu.cu:8089/xmlui/handle/123456789/9467.

ECHEVERRÍA PÉREZ, D. y FERNÁNDEZ PÉREZ, Y., 2013. *Desarrollo de una ontología de apoyo al procedimiento del Departamento de Pruebas de Software.* [en línea]. La Habana: Universidad de las Ciencias Informáticas. Disponible en: https://repositorio.uci.cu/jspui/handle/ident/7999.

EXPERIENCIAS EN LA IMPLANTACIÓN DE UN SISTEMA DE GESTIÓN DE LA CALIDAD PARA EL PROCESO DE PRODUCCIÓN DE SOFTWARE. [en línea], 2004. [Consulta: 15 julio 2021]. Disponible en: https://docplayer.es/2845760-Experiencias-en-la-implantacion-de-un-sistema-de-gestion-de-la-calidad-para-el-proceso-de-produccion-de-software.html.

GRUBER, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907-928. DOI https://doi.org/10.1006/ijhc.1995.1081 Get.

GUARINO, N., 1998. *Formal ontology in information systems: Proceedings of the first international conference (FOIS’98), June 6-8, Trento, Italy*. S.l.: IOS press.

HORRIDGE, M., BRANDT, S., PARSIA, B. y RECTOR, A.L., 2014. A domain specific ontology authoring environment for a clinical documentation system. *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*. S.l.: IEEE, pp. 329-334.

IAN SOMMERVILLE, 2005. *Ingeniería del software*. 7a edición. Madrid (España): s.n. ISBN 84-7829-074-5. Informática

IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. *IEEE Std 610*, 1991. pp. 1-217. DOI 10.1109/IEEESTD.1991.106963.

IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, 1990. pp. 1-84. DOI 10.1109/IEEESTD.1990.101064.

JIMÉNEZ, A.G., 2003. Organización y gestión del conocimiento en la comunicación. *El profesional de la información*, vol. 12, no. 4.

JOSÉ MANUEL SÁNCHEZ PEÑO, 2015. *Pruebas de Software. Fundamentos y Tecnicas.* S.l.: Universidad Politecnica de Madrid.

JULIÁN ANDRÉS MERA PAZ, 2015. La Importancia del proceso de pruebas de Calidad de Software en la Formación de los Ingenieros de Sistemas. [en línea]. [Consulta: 17 julio 2021]. Disponible

en: https://www.ucc.edu.co/prensa/2015/Paginas/la-importancia-del-proceso-de-pruebas-de-calidad-de-software-en-la-formacion-de-los-ingenieros-de-sistemas.aspx.

JULIÁN ANDRÉS MERA PAZ, 2016. Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria* [en línea], vol. 12, no. 20. [Consulta: 21 julio 2021]. Disponible en: https://repository.ucc.edu.co/handle/20.500.12494/962.

KOTSIANTIS, S.B., KANELLOPOULOS, D., KARIOTI, V. y TAMPAKAS, V., 2009. An ontology-based portal for credit risk analysis. *2009 2nd IEEE international conference on computer science and information technology*. S.l.: IEEE, pp. 165-169.

LÓPEZ, D.S., PÉREZ, J.F.R. y FERIA, L.M.G., 2017. La integración en la gestión de proyectos: diagnóstico y buenas prácticas a implementar en la UCI. *Serie Científica de la Universidad de las Ciencias Informáticas* [en línea], vol. 10, no. 3. [Consulta: 15 julio 2021]. ISSN 2306-2495. Disponible en: https://publicaciones.uci.cu/index.php/serie/article/view/109.

LÓPEZ-HUERTAS, M.J., 2008. Some current research questions in the field of knowledge organization. *KO KNOWLEDGE ORGANIZATION*, vol. 35, no. 2-3, pp. 113-136.

LUNA GONZÁLEZ, M.E., 2015. Organización del conocimiento en la red digital. *Investigación bibliotecológica*, vol. 29, no. 67, pp. 77-89. ISSN 0187-358X.

LUNA, J.A.G., BONILLA, M.L. y TORRES, I.D., 2012. Metodologías y métodos para la construcción de ontologías. *Scientia et Technica*, vol. 2, no. 50, pp. 133-140. ISSN 2344-7214. DOI 10.22517/23447214.6693.

MANJÓN, J.V.G., 2010. *Gestin de La Innovacin Empresarial*. S.l.: Netbiblo. ISBN 978-84-9745-477-3.

MARIO PÉREZ-MONTORO GUTIÉRREZ, 2008. *Gestión del conocimiento en las organizaciones*. España: Ediciones Trea, s. L. ISBN 978-84-9704-376-2.

MARTÍNEZ VALDIVIESO, A.V., 2020. *Ontología para el proceso de pruebas de liberación de la Universidad de las Ciencias Informáticas.* La Habana: Universidad de las Ciencias Informáticas.

MASCHERONI, M., GREINER, C., PETRIS, R., DAPOZO, G., y ESTAYNO M.(, 2012. Calidad de software e Ingeniería de Usabilidad. ,

MONTOYA, E.S., 2010. Acercamiento ontológico a la gestión del conocimiento en el mantenimiento del software. , pp. 10.

MORA, R.P. y ARIAS, B.L.I., 2018. La organización del conocimiento como proceso: la movilización del conocimiento. *Liinc em Revista* [en línea], vol. 14, no. 2. [Consulta: 25 julio 2021]. ISSN 1808-3536. DOI 10.18617/liinc.v14i2.4308. Disponible en: http://revista.ibict.br/liinc/article/view/4308.

Norma ISO/IEC 25000. [en línea], 2015. [Consulta: 17 julio 2021]. Disponible en: https://revistas.udistrital.edu.co/index.php/tia/article/view/8373.

OBERLE, D., VOLZ, R., STAAB, S. y MOTIK, B., 2004. An extensible ontology software environment. *Handbook on ontologies*. S.l.: Springer, pp. 299-319.

PÉREZ, D. y DRESSLER, M., 2007. Tecnologías de la información para la gestión del conocimiento. [en línea], [Consulta: 2 julio 2021]. ISSN 1697-9818. Disponible en: https://upcommons.upc.edu/handle/2099/2945.

Proceso productivo de la UCI evaluado con CMMI nivel 2 | Universidad de las Ciencias Informáticas. [en línea], 2015. [Consulta: 15 julio 2021]. Disponible en: https://www.uci.cu/proceso-productivo-de-la-uci-evaluado-con-cmmi-nivel-2.

PROENZA, R.S.G. y MARTÍNEZ, A.G., 2012. Metodología para el diseño y desarrollo de ontologías en el campo de la educación. *Revista Cubana de Ciencias Informáticas*, vol. 6, no. 2, pp. 1-11.

RAMOS, E., 2009. *Esquema para evaluar ontologías Ramos, Esmeralda; Núñez, Haydemar; Casañas, Roberto*. S.l.: s.n.

RAMOS, E. y NÚÑEZ, H., 2007. ONTOLOGÍAS: componentes, metodologías, lenguajes, herramientas y aplicaciones. [en línea]. Caracas: Universidad Central de Venezuela. Lecturas en Ciencias de la Computación. Disponible en: https://docplayer.es/54553669-Ontologias-componentes-metodologias-lenguajes-herramientas-y-aplicaciones.html.

REUCO, R., 2008. *Marco de Trabajo para el desarrollo de aplicaciones con ontologías de dominio*. Santa Clara: Universidad Central “Marta Abreu” de Las Villas.

RODRÍGUEZ, Y.A.L. y MARTÍNEZ, C.N.S., 2017. *Método para la integración de ontologías en sistemas relacionales para la evaluación de créditos*. Maestria en Informatica Aplicada. La Habana: Universidad de las Ciencias Informáticas.

ROGER S. PRESSMAN, 2010. *Ingeniería del software. Un Enfoque Practico*. Séptima Edición. S.l.: s.n. ISBN 978-607-15-0314-5.

ROJAS GRASS, O.Y., 2018. *Método para la descripción y validación de procesos de negocio de gestión empresarial basado en ontología*. La Habana: Universidad de las Ciencias Informáticas.

SÁNCHEZ, A.S., 2017. Sistemas para la organización del conocimiento: definición y evolución histórica. [en línea], vol. 7. ISSN 1658-142. DOI http://dx.doi.org/10.15517/eci.v7i2.26878. Disponible en: http://revistaebci.ucr.ac.cr.

SILEGA, N., TERESA-LOUREIRO, T., NOGUERA, M. y PEDRO-FEBLES, J., 2017. Framework basado en ontología para la descripción y validación de procesos de negocio. *Ingeniería Industrial*, vol. 38, no. 3, pp. 276-288.

VARGAS, G.A.T. y SANTAMARÍA, B.J., 2008. Los sistemas de organización del conocimiento y el manejo de contenidos digitales. *Biblioteca Universitaria*, vol. 11, no. 1, pp. 3-12. ISSN 0187-750X, 2594-0074.

ZHAO, H., ZHANG, S. y ZHAO, J., 2012. Research of using protege to build ontology. *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*. S.l.: IEEE, pp. 697-700.

ANEXOS

Anexo 1: Tablas y figuras generadas de los pasos de la metodología Methontology.

Tabla 13: Glosario de Términos

No.	Término	Descripción	Tipo
1.	Tipo de Prueba	Clasificación de las Pruebas.	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.	Criterio de criticidad	Elementos o bases para determinar el estado a un proyecto al presentar un número considerable de No Conformidades.	Concepto
5.	Sub-característica de usabilidad	El concepto Modelo de calidad tiene Sub-características de usabilidad.	Concepto
6.	Medida de usabilidad	Medida interna de la usabilidad del software, medida externa de la usabilidad del software o medida de la calidad en el uso del software.	Concepto
7.	Proceso de prueba de usabilidad	El proceso de prueba comprende la planificación de pruebas, control, análisis y diseño de pruebas, implementación de pruebas y ejecución, evaluación de criterios de salida e informes, y prueba actividades de cierre.	Concepto
8.	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
9.	Técnica de prueba	Formas de cómo ejecutar las pruebas.	Concepto
10.	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
11.	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
12.	Especialista	Persona encarga de coordinar la evaluación en la Dirección de la Calidad de Software.	Concepto

13.	Herramienta	Instrumento que puede utilizarse durante la prueba para recolectar datos, llevar a cabo la interpretación de los mismos o automatizar parte de la prueba.	Concepto
14.	Resultado	Informe de evaluación de la calidad del producto de software	Concepto
15.	Actividad	Etapas por las que transita el procedimiento de pruebas.	Concepto
16.	Documentación	Documentos generados en cada una de las actividades del proceso de pruebas.	Concepto
17.	Métrica	Métricas de calidad de software es un conjunto de medidas utilizadas para estimar la calidad de un proyecto a desarrollar, entre otros conceptos, y que permiten comparar o planificar estas aplicaciones.	Concepto
18.	act1	Instancia perteneciente a la clase Actividad	Instancia
19.	artefacto1	Instancia perteneciente a la clase Artefacto	Instancia
20.	cc1	Instancia perteneciente a la clase Características de calidad	Instancia
21.	criterioC1	Instancia perteneciente a la clase Criterio de criticidad	Instancia
22.	casodeprueba	Instancia perteneciente a la clase Documentación	Instancia
23.	Rol1	Instancia perteneciente a la clase Especialista	Instancia
24.	herramienta1	Instancia perteneciente a la clase Herramienta	Instancia
25.	medidaC1	Instancia perteneciente a la clase Medida de calidad	Instancia
26.	modeloC1	Instancia perteneciente a la clase Modelo de calidad	Instancia
27.	métrica1	Instancia perteneciente a la clase Métrica	Instancia
28.	nivel1	Instancia perteneciente a la clase Nivel de prueba	Instancia
29.	nc1	Instancia perteneciente a la clase No Conformidad	Instancia
30.	producto1	Instancia perteneciente a la clase Producto	Instancia
31.	proyecto1	Instancia perteneciente a la clase Proyecto	Instancia
32.	resultado1	Instancia perteneciente a la clase Resultado	Instancia
33.	subCart1	Instancia perteneciente a la clase Sub-característica de calidad	Instancia
34.	rendimiento	Instancia perteneciente a la clase tipo de prueba	Instancia
35.	usabilidad	Instancia perteneciente a la clase tipo de prueba	Instancia
36.	seguridad	Instancia perteneciente a la clase tipo de prueba	Instancia
37.	t_caja_blanca	Instancia perteneciente a la clase Técnica de prueba	Instancia
38.	t_caja_negra	Instancia perteneciente a la clase Técnica de prueba	Instancia
39.	Tiene No conformidad	Es la relación que existe entre los conceptos (No Conformidad y Actividad) Y (No conformidad y Criterio de criticidad)	Relación
40.	Utiliza técnica de evaluación	Es la relación que existe entre los conceptos Tipos de pruebas y Técnicas de evaluación.	Relación
41.	Coordina artefacto	Es la relación que existe entre los conceptos Especialistas y Artefacto.	Relación

42.	Gestiona documentación	Es la relación que existe entre los conceptos Especialistas y Documentación.	Relación
43.	Manipula herramienta	Es la relación que existe entre los conceptos Especialistas y Herramienta.	Relación
44.	Cuantifica subcaracterísticas de usabilidad	Es la relación que existe entre los conceptos Medida de usabilidad y Subcaracterísticas de usabilidad.	Relación
45.	Tiene métricaM	Es la relación que existe entre los conceptos Medida de usabilidad y Métrica.	Relación
46.	Tiene métricaP	Es la relación que existe entre los conceptos Producto y Métrica.	Relación
47.	Tiene Actividad	Es la relación que existe entre los conceptos Pruebas de usabilidad y Actividad.	Relación
48.	Tieneproyecto	Es la relación que existe entre los conceptos Actividad y Proyecto.	Relación
49.	Tiene Especialistas	Es la relación que existe entre los conceptos Proyecto y Especialistas.	Relación
50.	Tiene producto	Es la relación que existe entre los conceptos Proyecto y Producto.	Relación
51.	Tiene modelo de calidad	Es la relación que existe entre los conceptos Pruebas de usabilidad y Modelo de calidad.	Relación
52.	Tiene resultado	Es la relación que existe entre los conceptos Pruebas de usabilidad y Resultado.	Relación
53.	Tienetipo de prueba	Es la relación que existe entre los conceptos Pruebas de usabilidad y Tipo de prueba.	Relación
54.	Tiene documentación	Es la relación que existe entre los conceptos Artefacto y Documentación.	Relación
55.	Tiene métricaP	Es la relación que existe entre los conceptos Producto y Métrica.	Relación
56.	Utiliza herramienta	Es la relación que existe entre los conceptos Técnica de evaluación y Herramienta	Relación
57.	Tiene subcaracterísticas de usabilidad	Es la relación que existe entre los conceptos Modelo de calidad y Subcaracterísticas de usabilidad.	Relación
58.	Se basa en modelo de calidad	Es la relación que existe entre los conceptos Pruebas de usabilidad y Modelo de calidad	Relación

Taxonomía de conceptos de la ontología Proceso de Pruebas de usabilidad.

Taxonomía 2: Taxonomía para los conceptos Modelo de calidad, Resultado, Proyecto, Producto y No Conformidad.

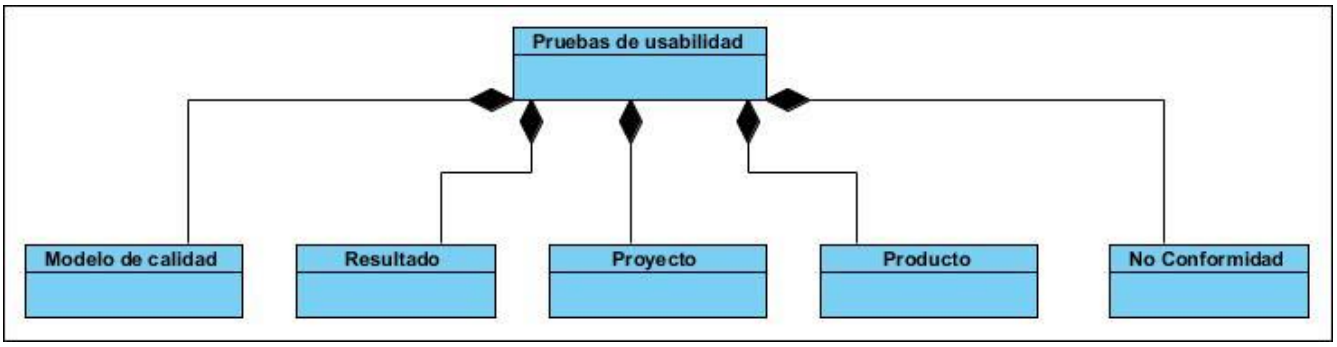

Figura 9: Taxonomía 2. Fuente: Elaboración propia

Taxonomía 3: Taxonomía para los conceptosHerramientas y Criterio de criticidad.

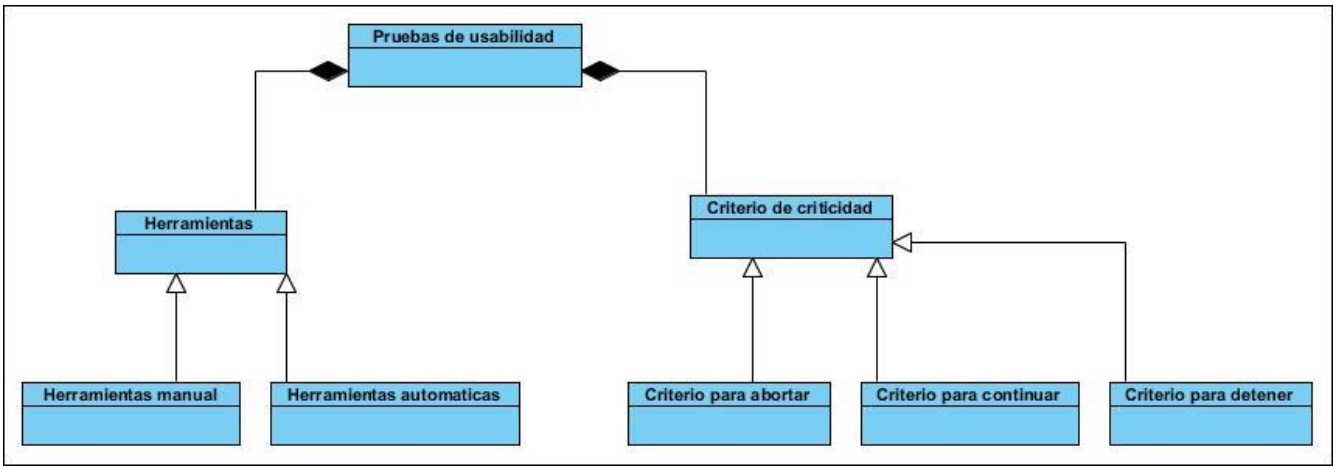

Figura 10: Taxonomía 3. Fuente: Elaboración propia.

Taxonomía 4: Taxonomía para los conceptos Documentación y Artefacto.

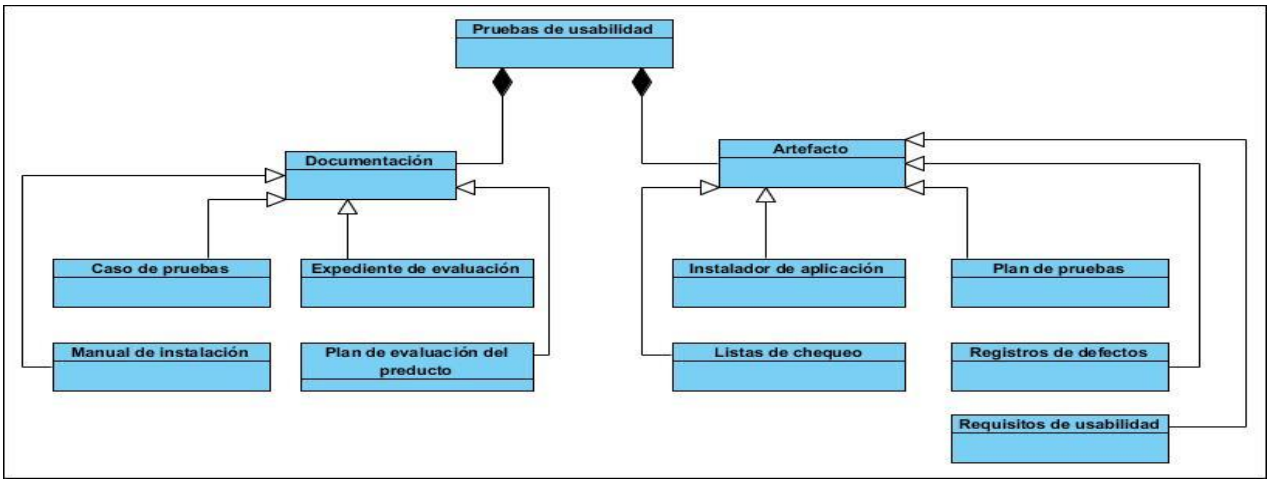

Figura 11: Taxonomía 4. Fuente: Elaboración propia.

Taxonomía 5: Taxonomía para los conceptos de Especialistas y Tipos de pruebas.

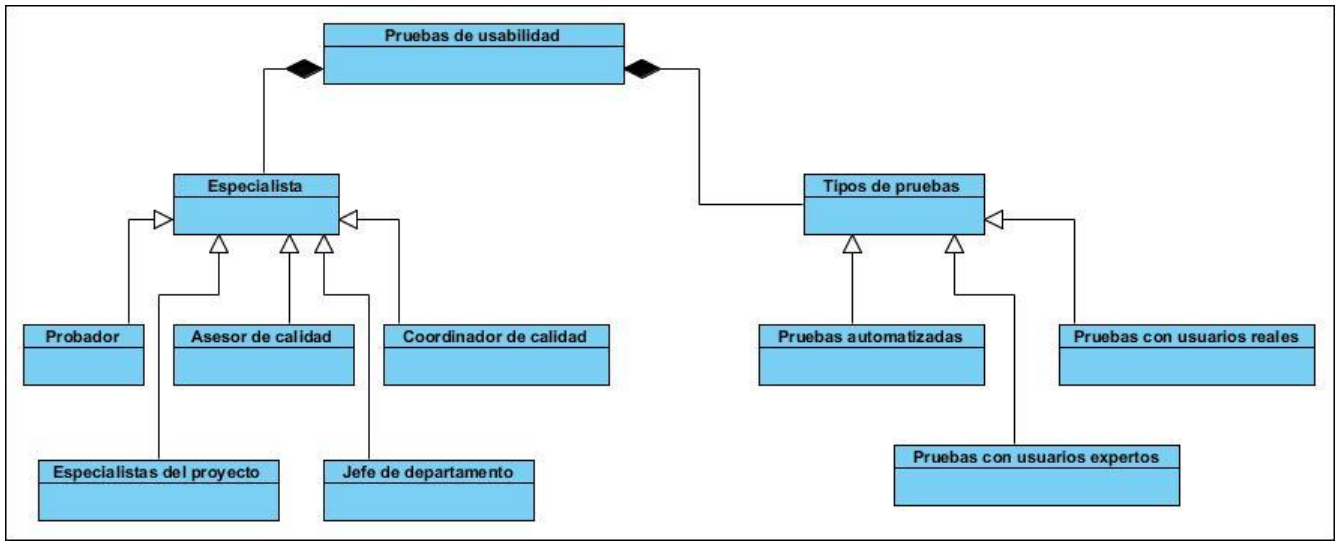

Figura 12: Taxonomía 5. Fuente: Elaboración propia.

Taxonomía 6: *Taxonomía para el concepto de Métrica.*

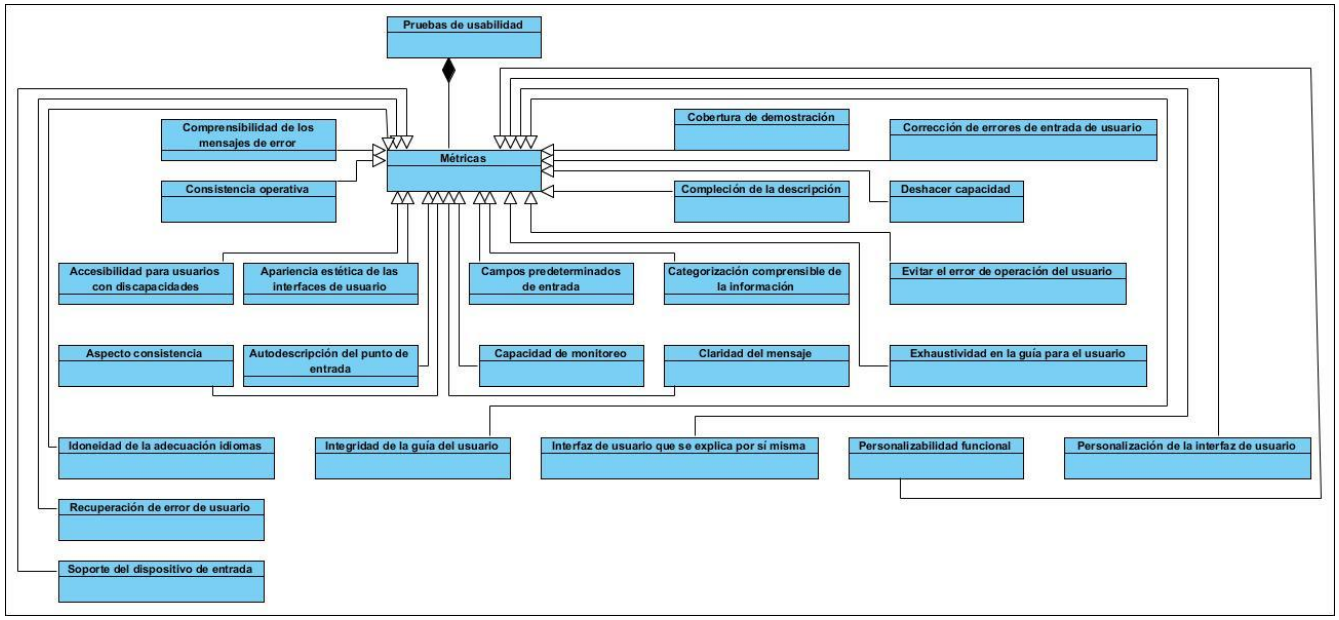

Figura 13: Taxonomía 6. Fuente: Elaboración propia.

Taxonomía 7: *Taxonomía para el concepto de Actividad y Medidas de usabilidad.*

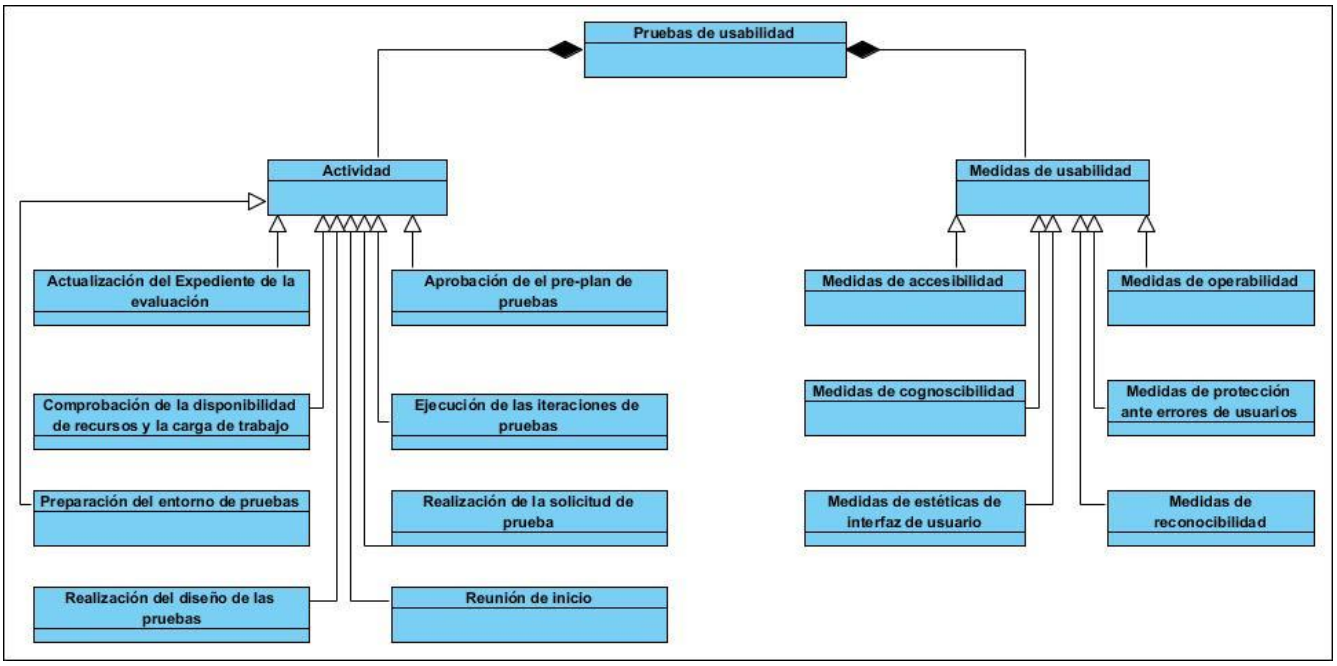

Figura 14: Taxonomía 7. Fuente: Elaboración propia.

Taxonomía 8: *Taxonomía para el concepto de Actividad y Medidas de usabilidad.*

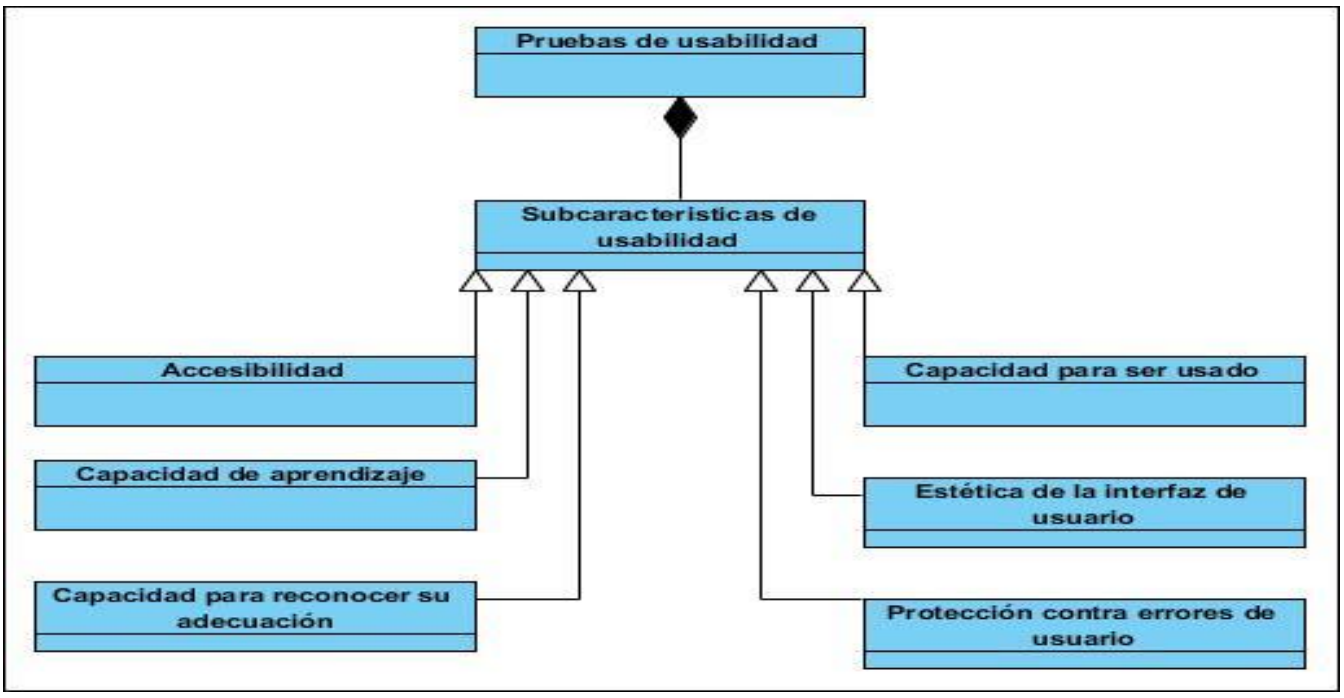

Figura 15: Taxonomía 8. Fuente: Elaboración propia.

Tabla 14: Diccionario de conceptos

Nombre del concepto	**Instancia**	**Atributos de clase**	**Atributos de instancia(Data property)**	**Relaciones(object property)**
Prueba de usabilidad	prueba		fecha Inicio fecha Fin	tiene actividad tiene modelo de calidad tiene resultado tiene tipo de prueba
Artefacto	artefacto		denominación descripción documentación	tiene documentación
Subcaracterísticas de usabilidad	subCart1 subCart2 subCart3 subCart4 subCart5 subCart6		denominación descripción	---
Métrica	Métrica1		denominación descripción	---
Producto	producto1		denominación descripción	tiene métricaP
Modelo de calidad	modeloC1		denominación descripción	tiene característica de calidad
Criterio de criticidad	criterioC1		denominación descripción	Tiene no conformidad
No Conformidad (NC)	nc1		descripción número de iteración	----
Medida de calidad	medidaC1		denominación descripción	cuantifica característica de calidad cuantifica sub-característica de calidad tiene metricaM
Herramienta	herramienta1 herramienta2		denominación versión descripción	---
Rol	espec1		nombre	manipula herramienta

	espec2 espec3 espec4 espec5 espec6 espec7 espec8		responsabilidad descripción	coordina artefacto gestiona documentación
Característica de calidad	cc1 cc2 cc3 cc4 cc5		denominación consecuencia descripción	tiene sub-característica de calidad tiene tipo de prueba
Documentación	documentación1 documentación3 documentación2 documentación4		denominación descripción	---
Nivel de prueba	nivel1 nivel2 nivel3		denominación descripción	--
Proyecto	proyecto1		denominación descripción	tiene producto tiene rol tiene actividad
Actividad	act1		denominación descripción fecha Inicio fecha Fin sesiones de trabajo responsable participante observación	tiene no conformidad
Técnica de prueba	t_caja_blanca t_caja_negra		denominación descripción	utiliza herramienta
Resultado	resultado1		descripción	---
Tipo de prueba	funcionales no_funcionales		denominación descripción	Utiliza técnica de prueba Tiene nivel de prueba

Tabla 15: Relaciones binarias en detalle.

Nombre de la relación	Concepto Origen	Cardinalidad	Concepto Destino	Relación Inversa
Tiene no conformidad	Criterio de criticidad	1…n	No conformidad	--
utiliza técnica de evaluación	Tipos de pruebas	1…n	Técnica de evaluación	es técnica de evaluación de
coordina artefacto	Especialistas	1…n	Artefacto	es coordinado por
gestiona documentación	Especialistas	1…n	Documentación	es gestionada por
manipula herramienta	Especialista	1…n	Herramienta	es manipulada por
cuantifica subcaracterísticas de usabilidad	Medida de usabilidad	1…n	Subcaracterísticas de usabilidad	es cuantificada por
tiene métricaM	Medida de usabilidad	1…n	Métrica	es métrica de
tiene proyecto	Actividad	1…n	Proyecto	es proyecto de
tiene especialista	Proyecto	1…n	Especialista	es especialista de
tiene producto	Proyecto	1…n	Producto	es producto de
tiene actividad	Prueba de usabilidad	1…1	Actividad	--
tiene modelo de calidad	Prueba de usabilidad	1…1	Modelo de calidad	--
tiene resultado	Prueba de usabilidad	1…1	Resultado	--
Tiene tipo de prueba	Prueba de usabilidad	1...n	Tipo de prueba	--
tiene documentación	Artefacto	1…n	Documentación	es documentación de
tiene métricaP	Producto	1…n	Métrica	es métrica de p
tiene subcaracterísticas de usabilidad	Modelo de calidad	1…n	Subcaracterísticas de usabilidad	--
utiliza herramienta	Técnica de evaluación	1…n	Herramienta	es utilizada por

Tabla 16: Atributos de instancia en detalle

Atributo de instancia	**Concepto**	**Tipo de valor**	**Rango**	**Cardinalidad**
Descripción	Actividad Artefacto Documentación Criterio de criticidad Especialista Herramienta Medida de usabilidad Modelo de calidad Métrica No conformidad Producto Proyecto Resultado Subcaracterísticas de usabilidad Técnica de evaluación Tipo de prueba	string	-	1…1
Nombre	Especialista	string	-	1…1
Denominación	Actividad Artefacto Documentación Criterio de criticidad Herramienta Medida de usabilidad Modelo de calidad Métrica No conformidad Producto Proyecto	string	-	1…1

	Subcaracterísticas de usabilidad Técnica de evaluación Tipo de prueba			
Responsable Fecha Inicio Fecha Fin Resultado Observación	Actividad	string	-	1…1

Tabla 17: Ejemplo de reglas en la ontología Proceso de prueba de usabilidad.

Nombre de la regla	**Descripción**	**Expresión**	**Conceptos**	**Relaciones**
Especialista Probador	Especialista Probador gestiona alguna Documentación.	Especialista and (gestiona documentación some Caso de prueba)	Probador Caso de prueba	gestiona documentación
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 sola una Documentación y manipula solo una Herramienta.	Especialista and (gestiona documentación only Expediente de la evaluación) and (manipula herramienta only Herramienta automática)	Coordinador de la evaluación Expediente de la evaluación Herramienta automática	gestiona documentación manipula herramienta
Especialista del equipo de desarrollo	Especialista del equipo de desarrollo coordina solo un Artefacto.	Especialista and (coordina artefacto only Instalador de aplicación) and (gestiona documentación only Plan de evaluación del producto.	Especialista del equipo de desarrollo Instalador de aplicación	coordina artefacto gestiona documentación

Tabla 18: Describir atributos de clases.

Nombre del concepto	**Instancia**	**Atributos de instancia(Data property)**	**Valor de los atributos de instancia**
Prueba de usabilidad	prueba	fecha Inicio	--
		fecha Fin	
Artefacto	Artefacto1	denominación	
		descripción	
Subcaracterística de usabilidad	subcart1 subcart2 subcart3 subcart4 subcart5 subcart6	denominación	
		descripción	
Métrica	Métrica1	denominación	
		descripción	
Producto	producto1	denominación	
		descripción	
Modelo de calidad	mc1	denominación	
		descripción	
Criterio de criticidad	cc1	denominación	
		descripción	
No Conformidad (NC)	NC1	descripción	
		denominación	
Medida de usabilidad	med1 med2 med3	denominación	
		descripción	

	med4 med5 med6		
Herramienta	herramienta1 herramienta2	denominación	
		descripción	
Especialista	especialista1 especialista2 especialista3 especialista4 especialista5 especialista6	nombre	
		descripción	
Documentación	documentación1 documentación3 documentación2 documentación4	denominación	
		descripción	
Proyecto	proyecto1	denominación	
		descripción	
Actividad	act1	denominación	
		descripción	
		fecha Inicio	
		fecha Fin	
		responsable	
Resultado	resultado1	descripción	
Tipo de prueba	TP1 TP2 TP3	denominación	
		descripción	

Anexo 3: Casos de prueba de la Ontología para el proceso de pruebas de usabilidad de la Universidad de las Ciencias Informáticas.

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 19)

Tabla 19: Ejemplo de instancia de la clase Especialista.

Clases	Instancias	Propiedades	Valor de las Propiedades
Eapecialista	Especialista6	manipula herramienta	herramienta1

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

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

Caso de Prueba 3

Pregunta de competencia: ¿Qué medida de usabilidad cuantifica la característica de usabilidad 1?

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

Clases	Instancias	Propiedades	Valor de las Propiedades
Medidas de usabilidad	Med1	Cuantifica la Subcaracterística de usabilidad	subcaract1

Resultado esperado: Al aplicar un razonador el Med1 debe clasificarse Medida de Accesibilidad.