

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad 1

Desarrollo de una ontología para pruebas de aceptación con usuarios
finales

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

Autor:

Alexis Edgardo Guerrero Herrera

Tutores:

MSc. Alionuska Velázquez Cintra

Ing. Yordanka Fuentes Castillo

La Habana, diciembre de 2021

“Año 63 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Alexis Edgardo Guerrero Herrera, con carné de identidad 97092321405 soy el autor principal del trabajo de diploma “Desarrollo de una ontología para pruebas de aceptación con usuarios finales”, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmamos la presente en La Habana a los ____ días del mes de _____ del año _____.



Alexis Edgardo Guerrero

Herrera

Autor

Ing. Yordanka Fuentes Castillo

Tutora

MSc. Alionuska Velázquez Cintra

Tutora

“A los revolucionarios más jóvenes, especialmente, recomiendo exigencia máxima y disciplina férrea, sin ambición de poder, autosuficiencia, ni vanaglorias. Cuidarse de métodos y mecanismos burocráticos. Usar la ciencia y la computación sin caer en lenguaje tecnicista e ininteligible de élites especializadas. Sed de saber, constancia, ejercicios físicos y también mentales”.

Fidel Castro Ruz

AGRADECIMIENTOS

A todos los que desde un inicio estuvieron presentes o en la distancia, pero que nunca se olvidaron de mí y me desearon de corazón éxitos.

DEDICATORIA

A mis preciosas joyas mamá, abuela, hermano y mi pareja que siempre han estado en cada proceso de mi vida apoyándome, siendo guía y bastón, y me dan el impulso día a día para ser una mejor persona y un excelente profesional.

RESUMEN

En la Ingeniería del Conocimiento, las ontologías están ocupando un lugar importante debido a las bondades que éstas brindan para organizar el conocimiento en dominios específicos. Estas aportan los elementos conceptuales y sus relaciones dentro del campo de análisis de la Ingeniería del Software, que propician el rendimiento y optimización acertada al equipo de trabajo, cuando elabora un producto orientado al usuario. Para asegurar los requisitos del usuario y la calidad esperada son fundamentales las pruebas de software. Dentro de estas, las de aceptación, ayudan a reducir los riesgos en las aplicaciones, identificando los defectos antes de que se ejecuten. En la Universidad de las Ciencias Informáticas, el Centro de Innovación y Desarrollo para Internet se encarga del proceso de desarrollo de portales web e intranets. A partir de una revisión documental a sus expedientes de proyectos, se comprobaron deficiencias causadas por problemas de comunicación entre los involucrados durante las pruebas de aceptación, evidenciándose que no existe una gestión del conocimiento donde se recojan las experiencias y los posibles problemas e insatisfacciones presentadas por los clientes en esta actividad. El presente trabajo describe el desarrollo de una ontología de tipo aplicación para pruebas de aceptación con usuarios finales para lograr una comunicación efectiva entre clientes, usuario final y el equipo de desarrollo. En su desarrollo se empleó la herramienta Protégé Desktop en su versión 5.5.0, y el lenguaje OWL, guiado por la metodología Methontology. Para validar la solución se aplicó consulta de expertos y el Método de evaluación triangular.

Palabras clave: ontología, pruebas de aceptación, usuarios finales, portales web

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Ontología	6
1.1.1 Conceptos relacionados	6
1.1.2 Tipos de ontología	7
1.1.3 Componentes de la ontología	9
1.1.4 Lenguajes para la representación de ontologías	10
1.1.5 Herramientas para la construcción de ontologías	12
1.1.6 Metodología para la construcción de ontologías	14
1.2 Pruebas de aceptación	16
1.2.1 Desarrollo de las pruebas de aceptación en CIDI	18
1.3 Conclusiones del capítulo	21
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN E IMPLEMENTACIÓN	22
2.1. Descripción de la propuesta de solución	22
2.2. Aplicación de la metodología Methontology	24
2.2.1 Especificación	24
2.2.2 Conceptualización	25
2.2.3 Formalización	35
2.2.4 Implementación	40
2.2.5 Instalación y Mantenimiento	42
2.3. Conclusiones del capítulo	43
CAPÍTULO 3: VALIDACIÓN Y PRUEBA DE LA ONTOLOGÍA	44
3.1. Validación de la ontología	44
3.2. Conclusiones del capítulo	48
CONCLUSIONES GENERALES	50
RECOMENDACIONES	51

REFERENCIAS BIBLIOGRÁFICAS.....	52
ANEXOS.....	56
Anexo 1.....	56
Anexo 2.....	57
Anexo 3.....	60
Anexo 4.....	61
Anexo 5.....	63
Anexo 6.....	64

ÍNDICE DE TABLAS

Tabla 1 Documento de Especificación de Requerimientos.....	24
Tabla 2 Glosario de términos.	25
Tabla 3 Diccionario de conceptos	30
Tabla 4 Relación binaria de la ontología Proceso de Pruebas de Aceptación.	31
Tabla 5 Descripción de los atributos de instancias.....	32
Tabla 6 Ejemplo de reglas en la ontología en el proceso de pruebas de aceptación.....	34
Tabla 7 Lista de chequeo de errores comunes en el diseño de ontologías.	45
Tabla 8 Creación de instancias de los casos de prueba.....	47
Tabla 9 Características de los expertos.	63

ÍNDICE DE FIGURAS

Figura 1 Interfaz de Usuario de Ontolingua Server (Gómez Reyes, 2018).	12
Figura 2 Interfaz de usuario de Enterprise Architecture (UML, 2021).	13
Figura 3 Interfaz de usuario de Protégé Desktop v.5.5.0 (Protégé, 2021).	14
Figura 4 Descripción gráfica del Proceso de Pruebas de Aceptación de Software. Fuente: Elaboración propia.	17
Figura 5 Modelo conceptual de procesos en la UCI (López, 2016).	18
Figura 6 Flujo de actividades. Fuente: Elaboración propia.	22
Figura 7 Modelo conceptual del Proceso de Evaluación de Software. Fuente: Elaboración propia.	24
Figura 8 Taxonomía de los conceptos Fuente: Elaboración propia.	29
Figura 9 Diagrama de relaciones binarias de la ontología Proceso de Pruebas de aceptación. Fuente: Elaboración propia.	29
Figura 10 Definición de las clases creadas en Protégé. Fuente: Elaboración propia.	36
Figura 11 Object Properties (Propiedades de las clases creadas en Protégé. Fuente: Elaboración propia.	37
Figura 12 Creación de la Data Properties en Protégé. Fuente: Elaboración propia.	37
Figura 13 Creación de las reglas. Fuente: Elaboración propia.	38
Figura 14 Creación de las instancias con Protégé. Fuente: Elaboración propia.	38
Figura 15 Pestaña OWL Viz del Protégé. Fuente: Elaboración propia.	39
Figura 16 Declaración de las clases. Fuente: Elaboración propia.	40
Figura 17 Declaración de una subclase de otra clase. Fuente: elaboración propia.	40
Figura 18 Declaración de las relaciones. Fuente: Elaboración propia.	40
Figura 19 Creación de los individuos de una clase. Fuente: Elaboración propia.	41
Figura 20 Declaración de los axiomas. Fuente: Elaboración propia.	41
Figura 21 Consulta SPARQL. Fuente: Elaboración propia.	42
Figura 22 Caso de prueba 1. Fuente: Elaboración propia.	47
Figura 23 Caso de prueba 2. Fuente: Elaboración propia.	64
Figura 24 Caso de prueba 3. Fuente: Elaboración propia.	65
Figura 25 Caso de prueba 4. Fuente: Elaboración propia.	65
Figura 26 Caso de prueba 5. Fuente: Elaboración propia.	65

INTRODUCCIÓN

Las nuevas tecnologías surgen de la necesidad imperativa de utilizar y ampliar el área del conocimiento y la información en la sociedad, aplicarlas implica estar atento a las nuevas formas de crearla, intercambiarla y procesarla. Como herramientas para la representación y recuperación del conocimiento y la información existen: taxonomías, sistemas de clasificación, bases de datos léxicas, bases de conocimiento, mapas conceptuales, tesauros y ontologías (Aguilar, 2019).

En la Ingeniería del Conocimiento, las ontologías están ocupando un lugar importante y dando soluciones, en distintas áreas (la inteligencia artificial, la web semántica, la ingeniería de sistemas, la ingeniería de software, la informática biomédica, la bibliotecología y la arquitectura de la información), aprovechando las bondades que éstas brindan, para organizar el conocimiento en dominios específicos y permitiendo enriquecer semánticamente los recursos de la web de manera formal, explícita y consensuada (Silva Manriquez, 2021).

Las ontologías aportan los elementos conceptuales y relaciones entre estos dentro del campo de análisis de la Ingeniería del Software; punto clave que propicia el rendimiento y optimización acertada al equipo de trabajo cuando elabora un producto orientado al usuario. Con el desarrollo del software, se estudian los componentes necesarios para la gestión, mantenimiento y pruebas, siendo el producto final satisfactorio para el cliente (Cárdenas Quintero, 2021).

En la actualidad, el desarrollo de software se ha integrado cada vez más en las actividades de innovación de las empresas (Andersson, 2021). La capacidad de estas para proporcionar de forma recurrente y continua nuevas funciones de aplicación, que no sólo sean innovadoras y adecuadas para su uso; sino también productos de alta calidad, se ha convertido en un factor crítico en esta industria (Mishra & Otaiwi, 2020).

La calidad del software se ha ganado la cimera para optimizar el proceso y la calidad del producto propiciando procesos más institucionalizados. Su evolución está dada por usuarios más exigentes que demandan productos que cumplan con sus requerimientos y necesidades (Socarrás Ramírez, 2018).

Los requisitos del software son la base para la medida de la calidad. Describen las funciones necesarias que deben diseñarse y desarrollarse en un determinado sistema de software y

serven de acuerdo entre las partes interesadas del proyecto, dando paso a la transformación de modelos conceptuales fáciles de entender para evitar problemas de comunicación (Bozyig'ıt, 2021).

Para asegurar los requisitos del usuario y la calidad esperada son fundamentales las pruebas de software. Dentro de ellas se encuentran las pruebas de aceptación, centradas en el comportamiento y capacidad de todo el sistema, realizadas principalmente entre el equipo de trabajo y el cliente. Ayudan a reducir los riesgos en las aplicaciones, y logran de esta manera que se identifiquen los defectos antes de que se ejecuten (Díaz, 2021).

Varios autores coinciden en que dentro del contexto de las pruebas (Ponce, 2010a; Ponce, 2010b), una de las más importantes son las de aceptación (user's tests). Estas son a menudo responsabilidad de los clientes o usuarios de un sistema, a pesar de que también pueden participar otras partes interesadas (Marin et al., 2020).

En la Universidad de las Ciencias Informáticas (UCI), frente al proceso de desarrollo de portales web, soluciones integrales, productos y servicios relacionados con las tecnologías de internet, se encuentra el Centro de Innovación y Desarrollo para Internet (CIDI). CIDI cuenta con un área de calidad de software encargada de todos los procesos para que la calidad del producto que se entregue, al cliente o usuario final, sea de excelencia (CIDI, 2021).

El área de calidad de (CIDI) es responsable de guiar los procesos de pruebas de aceptación de los portales web desarrollados en el centro. A partir de una revisión documental a los expedientes de proyectos desarrollados en el centro, con énfasis en los procesos de aceptación realizados en el último año pudo comprobarse que:

- Los procesos de pruebas de aceptación exceden el tiempo planificado a partir de las solicitudes realizadas por el usuario final, aspecto que tiene mayor incidencia cuando cambia el personal involucrado en el proceso, de manera tal que dificulta la comprensión del producto.
- Varios criterios aportados por el cliente o los usuarios finales ayudan en el refinamiento de versiones posteriores del portal web o del proceso de desarrollo, sin embargo; no existen mecanismos que faciliten esta retroalimentación.
- Algunas funcionalidades son modificadas durante las pruebas de aceptación pues no fueron explicadas o comprendidas con claridad desde el levantamiento de requisitos.

Elemento que genera el trabajo, por lo que el esfuerzo en llegar a un producto final con la calidad requerida es mayor.

- Los usuarios suelen clasificar como no conformidades varias de las incidencias detectadas durante las pruebas de aceptación, aunque finalmente resultan ser solicitudes de cambios o problemas relacionados con el entorno de despliegue. Ello provoca en ocasiones insatisfacciones en clientes y usuarios finales.

Los elementos anteriormente expuestos demuestran problemas de comunicación entre los involucrados surgiendo inconformidad del cliente con la solución, al presentar otras necesidades que no se contemplaron en los requisitos firmados. A pesar de resaltar la importancia de las pruebas de aceptación y su influencia en la mejora de procesos y productos, no existe una gestión del conocimiento donde se recojan las experiencias y los posibles problemas e insatisfacciones presentadas por los clientes en esta actividad.

En entrevista realizada (Ver Anexo 1) al líder de desarrollo de portales web, jefes de proyectos y equipo de desarrollo se constató la necesidad de lograr un mejor entendimiento y una buena comunicación entre los involucrados en las pruebas de aceptación, ayudando a elevar de manera rápida y eficiente la calidad del producto, lo que facilitará la unificación del vocabulario en estas pruebas y servirá de consulta para una correcta descripción y clasificación de los defectos encontrados en el software.

Se evidencia la necesidad de contar con una herramienta que facilite este proceso de comprensión de los términos empleados a lo largos de los procesos de prueba de aceptación. Dada esta situación, el autor plantea como **problema de investigación**: ¿Cómo lograr una comunicación efectiva entre clientes, usuario final y el equipo de desarrollo durante las pruebas de aceptación?

Para dar solución a este problema de investigación, se toma como **objeto de estudio**: el proceso de pruebas de software en CIDI que contiene como **campo de acción**: el proceso de aplicación de las pruebas de aceptación con usuarios finales en CIDI. Trazándose como **objetivo**: desarrollar una ontología para pruebas de aceptación con usuarios finales.

Hipótesis: El desarrollo de una ontología para pruebas de aceptación con usuarios finales, permitirá lograr una comunicación efectiva entre clientes, usuario final y el equipo de desarrollo durante las pruebas de aceptación.

Para desarrollar esta hipótesis se formulan las siguientes **Tareas**:

1. Elaboración del marco teórico-conceptual de la investigación sobre las pruebas de aceptación y herramientas de creación de ontologías en los sistemas informáticos.
2. Identificación del estado actual de las pruebas de aceptación con usuarios finales en CIDI.
3. Generación de los artefactos de análisis y diseño de la aplicación web propuesta.
4. Implementación de la propuesta de solución.
5. Validación de la aplicación web a partir del Método consulta de expertos y el Método de evaluación triangular.

Como **métodos científicos** empleados:

Del nivel teórico:

Histórico – Lógico: se utilizó con el objetivo de verificar teóricamente cómo ha evolucionado la mejora del proceso de pruebas de aceptación. Permitió determinar las principales causas de la del problema en sí.

Analítico – Sintético: al descomponer el problema de investigación se utilizó para extraer los elementos relevantes relacionados con las pruebas de aceptación con usuarios finales a nivel de centro, enfatizando en las buenas prácticas asociadas a esta actividad.

Sistémico – Estructural: proporcionó organizar la información dispersa encontrada sobre los temas investigados y que los elementos que forman parte del proceso sean un todo que funcione de manera armónica.

Hipotético – Deductivo: permitió analizar las principales disposiciones en cuanto al desarrollo de las pruebas de aceptación con usuarios finales y llegar a soluciones anticipadas para el problema definido en función de la calidad del producto, siendo este el factor más afectado e importante en estos días.

Del nivel empírico:

Revisión documental: se utilizó con el objetivo de consultar los procesos de aceptación realizados en el último año demostrando la importancia de las pruebas de aceptación y su influencia en la mejora de procesos.

Entrevista: se entrevistó al líder del desarrollo de portales web y a jefes de proyectos resultando importante para la caracterización de la situación problemática, sirvió de evaluación y verificación para conocer cómo se realizaba el proceso de pruebas de aceptación con usuarios finales en los proyectos del centro y el tiempo promedio de atención a las incidencias.

Encuesta: para obtener el diagnóstico de la situación actual del centro en relación a la mejora del proceso de pruebas de aceptación y para validar los resultados de la investigación.

Estructura de la tesis:

El presente trabajo de investigación queda estructurado con resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas. Para auxiliar al entendimiento de la investigación se incluyen figuras, tablas y anexos.

Capítulo 1: **Fundamentación Teórica:** se presentan los conceptos fundamentales que sustentan la investigación a partir de los análisis de las características relacionadas con las ontologías y el proceso de las pruebas de aceptación con usuarios finales. Se analizan las tendencias actuales y metodologías empleadas en la elaboración de ontologías, además de otros conceptos, las herramientas, metodologías y lenguajes, para poder seleccionar los aspectos a tener en cuenta en la propuesta de solución.

Capítulo 2: **Diseño de la propuesta de solución e implementación:** se presenta la estructura de la propuesta de solución para entender mejor la ontología que se va a desarrollar. Se realiza una descripción de la propuesta de solución entre otros elementos propios de la fase de análisis y diseño, además del despliegue de las actividades de desarrollo y los componentes para el modelado de la ontología, a través de las herramientas mencionadas en el anterior capítulo por medio de la metodología Methontology. Esta propuesta ontológica tiene como objetivo recuperar y organizar todo lo relacionado con el proceso de las pruebas de aceptación en CIDI.

Capítulo 3: **Validación y prueba de la ontología:** se presentan los resultados de la evaluación de la ontología propuesta, dicho modelo posee cuatro fases, que tienen como objetivo comprobar el alcance y objetivo de la ontología desarrollada. La última fase muestra cómo la herramienta Protégé, validando la capacidad de inferencia de la ontología, responde a las preguntas de competencias identificadas. Para valorar la validez teórica y práctica de la ontología propuesta se utilizó el Método consulta de expertos y el Método de evaluación triangular.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se presentan los conceptos fundamentales que sustentan la investigación a partir de los análisis de las características relacionadas con las ontologías y las pruebas de aceptación, además de otros conceptos, las herramientas, metodologías y lenguajes, para poder seleccionar los aspectos a tener en cuenta en la propuesta de solución.

1.1 Ontología

1.1.1 Conceptos relacionados

La ontología es una antigua disciplina que se define como un esquema específico de categorías que refleja una visión específica del mundo. Desde el punto de vista informático, las ontologías especifican un vocabulario relativo a un cierto dominio. Las ontologías se encargan de definir los términos utilizados para describir y representar un área del conocimiento. Este concepto tiene sus orígenes en la filosofía, no hay una definición aceptada unánimemente para el concepto de ontología, sino que cada una de las definiciones que aparecen en la literatura aporta diferentes y a la vez complementarias visiones del significado de esta palabra (Echeverría, 2011).

La palabra ontología denota la ciencia del ser y las descripciones para la organización, designación y categorización de la existencia (Gruber, 1993). Éstas exhiben características especiales en el manejo de sistemas inteligentes para la representación del conocimiento. Los siguientes autores (Gruber, 1993; Chandrasekaran, 1999; Schreiber, 2001; McGuinness, 2004) destacan las siguientes:

- Las Ontologías proveen un vocabulario común y sin ambigüedades para referirse a los términos en el área aplicada, pudiéndose compartir o reutilizar éstos entre diferentes aplicaciones que hagan uso de la ontología.
- Especifican una taxonomía o herencia de conceptos que establecen una categorización o clasificación de las entidades del dominio. Una buena taxonomía es simple y fácil de recordar, separa sus entidades de forma mutuamente excluyente, y define grupos y subgrupos sin ambigüedad.
- El vocabulario y la taxonomía representan un marco de trabajo conceptual para el análisis, discusión o consulta de información de un dominio.

- Una ontología incluye una completa generalización/especificación de sus clases y subclases, las cuales están formalmente especificadas (incluyendo sus relaciones e instancias) asegurando la consistencia en los procesos deductivos.
- Las ontologías son implementadas en un lenguaje específico de representación ontológica (*ontology representation languages*) de manera que la especificación de sus clases, relaciones entre éstas y sus restricciones dependerán de las características de dicho lenguaje (Fernández, 2015).

Las ontologías aportan un esquema formal de especificaciones explícitas sobre las conceptualizaciones del dominio. Este es el motivo principal por el que las ontologías se consideran tecnologías apropiadas para la interoperabilidad de sistemas heterogéneos, el desarrollo y uso de sistemas inteligentes, la gestión del conocimiento y la comprensión de la información que se difunde, como una forma de reducir la brecha existente entre los usuarios y los expertos de un dominio particular (Galli, 2014).

Al analizar las definiciones dadas por diferentes autores para la investigación de este trabajo de diploma, el autor asume el concepto de ontología como un soporte para el análisis conceptual de métodos, técnicas, diseño, desarrollo en el uso de los sistemas informáticos. La base para una ontología es la conceptualización junto con un vocabulario, para referirse a las entidades de un dominio particular, son muy útiles para estructurar y definir el significado de los términos. En esta perspectiva se analizan dos dimensiones: la visión de los desarrolladores, concerniente a la manera en que una ontología ayuda o se usa para desarrollar un producto informático y la visión del usuario, relativa a la manera en que una ontología facilita la tarea del mismo al interactuar con el producto (Fernández, 2015).

1.1.2 Tipos de ontología

Los autores de diferentes libros a partir de sus estudios no definen una tipología en específico cuando se trata de ontologías, los parámetros para agruparlas varían de un autor a otro aunque en algunos casos coinciden. Uno de los parámetros para clasificar las ontologías se refiere al problema que solucionan. Es aportado por Ikeda, Guarino, Mizoguchi y Vanwelkenhuysen quienes agrupan bajo este enfoque a las ontologías (Echeverría, 2011):

- Contenido: permiten reutilizar el conocimiento de un dominio en otros sistemas y aplicaciones.
- Indización: utilizadas en la recuperación de información.

- Comunicación: facilitan el proceso de comunicación a través de repuestas a preguntas concretas.
- Meta-Ontologías: usadas para representar las ontologías de dominios similares o diferentes.

Según su dependencia y relación con una tarea específica desde un punto de vista, Guarino clasifica las ontologías en (Guarino, 1995):

- Ontologías de alto nivel o genéricas: describen conceptos más generales.
- Ontologías de dominio: describen un vocabulario relacionado con un dominio genérico.
- Ontologías de tareas o técnicas básicas: describen una tarea, actividad o artefacto, por ejemplo componentes, procesos o funciones.
- Ontologías de aplicación: describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

Según Van Heist las ontologías también pueden clasificarse de acuerdo a la cantidad y tipo de estructura de la conceptualización en (Fernández, 2003):

- Ontologías terminológicas, lingüísticas: especifican los términos usados para representar conocimiento en el dominio.
- Ontologías de información: definen la estructura de los registros de la base de datos. Los esquemas de bases de datos serían un ejemplo.
- Ontologías para modelar conocimiento: detallan conceptualizaciones de conocimiento. Estas ontologías tienen una estructura interna mucho más rica que los anteriores tipos de ontologías, y éstas son las ontologías que interesan a los desarrolladores de sistemas basados en conocimiento.

Una clasificación alternativa fue propuesta por Mizoguchi, donde también se proponen tres categorías (Mizoguchi, 1995):

- Ontologías del dominio: Contienen todos los conceptos asociados a un dominio particular.
- Ontologías de tarea: Establecen la forma en la cual se puede usar el conocimiento del dominio para realizar tareas específicas. De esta forma, una aplicación podría realizar

búsquedas de información mientras otra podría gestionar la asignación de bloques libre de memoria.

- Ontologías generales: Contienen descripciones generales sobre objetos, eventos, relaciones temporales, relaciones causales, modelos de comportamiento y funcionalidades.

El análisis realizado resalta la importancia de las ontologías para definir la tipología a usar para el desarrollo de la investigación. Las clasificaciones aportadas por Van Heist, de manera específica las de tipo aplicación, se ajustan a contextos en los que intervienen conceptualizaciones, registros y términos involucrados de diferentes niveles del conocimiento que describen conceptos que dependen del proceso de las pruebas de aceptación en CIDI.

1.1.3 Componentes de la ontología

Una ontología necesita especificaciones en el dominio y para que se valide necesita de dos tipos de componentes: los elementos y las relaciones entre estos, así generalizando los principales componentes de una ontología los autores (Gruber, 1993; Echeverría, 2011; Galli, 2014; Fernández, 2015) en su trabajo los definen como:

- Conceptos: ideas básicas reflejadas en términos que se intentan formalizar, la descripción de una tarea, función, acción y estrategia. Pueden ser clases de objetos, métodos, planes, procesos de razonamiento, etc. Las clases en una ontología se suelen organizar en taxonomías a las que se les pueden aplicar los mecanismos de herencia.
- Individuos o instancias: representaciones de las entidades, conceptos o elementos particulares y concretos que pertenecen a una clase. Por lo general son propiedades que no se recogen en el concepto o nociones que sirven para aclarar la connotación de este en un contexto determinado.
- Relaciones: son las relaciones que establecen el tipo de interacción semántica, generalmente binarias, entre los conceptos. Suelen formar la taxonomía del dominio. Entre las principales relaciones semánticas que pueden ser establecidas se encuentran las relaciones lógicas. El primer argumento de la relación se conoce como el dominio y el segundo como el rango.
- Funciones: constituyen un tipo especial de relación, a través de las cuales, se identifica un elemento mediante el cálculo de una función que considera varios elementos de la

ontología (el elemento n -ésimo de la relación es único para los $n-1$ restantes). Por ejemplo, pueden aparecer funciones como categorizar-clase, asignar-fecha, etc. Este componente es imprescindible cuando las ontologías son usadas para modelar sistemas y procesos.

- Reglas o axiomas: teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Los axiomas formales sirven para modelar sentencias que son siempre ciertas. Se usan para representar conocimientos que no puede ser formalmente definidos por los componentes descritos anteriormente, además, para verificar la consistencia de la propia ontología.

No siempre todos estos elementos deben aparecer en una ontología. En el caso de que se tenga una ontología que posee solo conceptos y relaciones se llama taxonomía u ontología ligera (Bastidas, 2007).

Una acertada concatenación de estos elementos (conceptos, instancias, relaciones, funciones y axiomas) y cumpliendo con los principios del diseño de las ontologías (ver Anexo 2), servirá de apoyo para que los usuarios finales, al buscar la información disponible en la ontología ya creada, sepan utilizarla como recomendador de buenas prácticas y guía, para la correcta realización de las pruebas de aceptación, en los diferentes portales web que se desarrollan en el centro.

1.1.4 Lenguajes para la representación de ontologías

Con el desarrollo del Internet se han creado varios lenguajes enmarcados en las necesidades de la web. Dentro de los principales lenguajes que se usan para el desarrollo de ontologías que explica el autor Stadnicki (2020), se destacan los siguientes:

- SHOE: HTML Ontology Extensions, un proyecto que comenzó en 1995, es un lenguaje de representación del conocimiento que permite anotar las páginas web con semántica simple, fue el primer lenguaje de etiquetado para diseñar Ontologías en la Web. Este lenguaje nació antes de que se ideara la Web Semántica. Las Ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones. Con el tiempo se fueron desarrollando editores, buscadores, APIs; dando paso al abandono del proyecto, fue entonces que se desarrollaron OIL y DAM; existiendo una socialización de este lenguaje en XML (Heflin, 2001).

- OIL: Ontology Inference Layer. Este lenguaje, derivado en parte de SHOE, fue impulsado también por el proyecto de la Unión Europea On-To-Knowledge. Utiliza la sintaxis del lenguaje XML y está definido como una extensión de RDFS (modelo para la descripción de recursos web basado en redes semánticas y con sintaxis XML) (Fernández, 2015). Se basa tanto en la lógica descriptiva (declaración de axiomas) como en los sistemas basados en frames (taxonomías de clases y atributos) (Echeverría, 2011). OIL posee varias capas de sub-lenguajes, entre ellas destaca la capa base que es RDFS, a la que cada una de las subsiguientes añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas (Horrocks, 2001).

La verdadera revolución en el mundo de los lenguajes ontológicos llegó en 2004, cuando OWL (un lenguaje basado en DAML+OIL) se convirtió en un estándar oficial del W3C (World Wide Web Consortium) como lenguaje para la creación de ontologías. OWL ganó rápidamente en popularidad y la reacción general de la comunidad fue positiva, aunque se registraron algunas deficiencias. Así, la siguiente versión del lenguaje, OWL 2 (Hitzler, 2009), se publicó y se proclamó como estándar en 2008, lo que aún no ha cambiado (Stadnicki, 2020).

- OWL: Web Ontology Language o Lenguaje de Ontologías para la Web, es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste. Se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones, donde el contenido solamente necesita ser presentado a los seres humanos (Echeverría, 2011). Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que se precise y a los distintos tipos de aplicaciones existentes (motores de búsqueda, agentes, etc.). Existen 3 sublenguajes de OWL, los cuales van creciendo respecto al nivel de expresión (McGuinness, 2004):

1. OWL Lite: Útil para la creación de jerarquías y restricciones simples, sólo permite valores de cardinalidad 0 y 1, pierde en expresividad.

2. OWL DL: (Description Logic), es el lenguaje más sencillo e indicado para los usuarios que requieren el máximo de expresividad y decibilidad (todos los cálculos acaban en un tiempo finito). Una clase puede ser a la vez subclase de muchas clases, no puede ser una instancia de otra clase.
3. OWL Full: Máximo nivel de expresión y la libertad sintáctica de RDF. Permite expresiones de segundo orden. Por ejemplo, una clase puede ser tratada simultáneamente como una colección de individuos y como un individuo por sí mismo. Es el más completo por lo que se necesita mucho poder computacional para poder hacer inferencias es por eso que se dice que no tiene garantía computacional.

Se propone como lenguaje de representación de ontologías el OWL en su versión DL ya que es un lenguaje diseñado para un procesamiento total de la información, facilitando un mejor mecanismo para que el usuario final, interprete el contenido y sea eficiente a la hora de realizar las pruebas de aceptación.

1.1.5 Herramientas para la construcción de ontologías

Entre las herramientas más destacadas utilizadas para la construcción de las ontologías actualmente, se encuentran (Ayora, 2021):

Ontolingua Server: Esta herramienta permite obtener los mecanismos para crear o reutilizar una ontología ya existente. Inicialmente consistía en un único módulo, el editor de ontologías, aunque progresivamente, con el paso del tiempo, se han ido añadiendo nuevos módulos funcionales, como un editor de ecuaciones, un Webster...(Corcho, 2003). Limita su uso la sencillez y lo primitivo de su diseño comparando esta herramienta con otras en la actualidad, véase la siguiente imagen de la interfaz de usuario de Ontolingua Server.



Figura 1 Interfaz de Usuario de Ontolingua Server (Gómez Reyes, 2018).

Enterprise Architecture: Este software permite el diseño, modelado y construcción no sólo de ontologías, si no de cualquier ciclo de vida de cualquier tipo de proyecto, a través de una interfaz de modelado gráfica sencilla e intuitiva para el usuario que no se encuentra en otras herramientas de diseño de ontologías. La herramienta también posee, entre otros, estándares SysML u OWL, lo que permite la exportación de ontologías en formato.owl directamente. Se puede encontrar en la siguiente dirección <http://www.sparxsystems.com.ar/>.

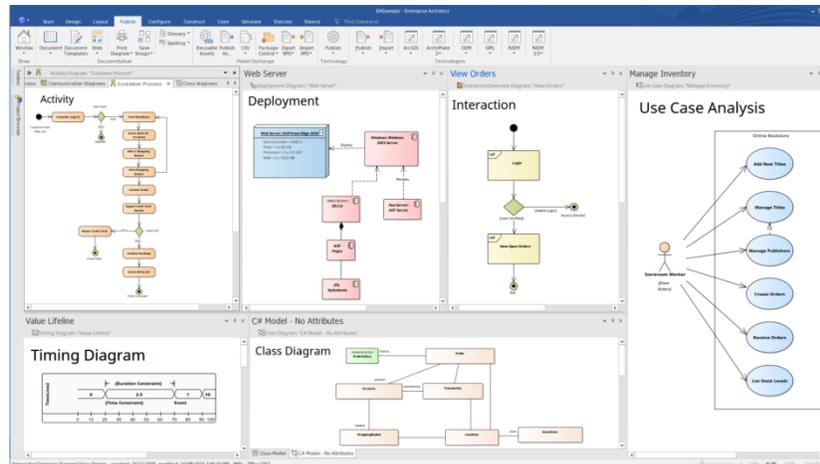


Figura 2 Interfaz de usuario de Enterprise Architecture (UML, 2021).

Protégé Desktop v.5.5.0: es un entorno de edición de ontologías rico en funcionalidades, con soporte completo para el Lenguaje de Ontología OWL, admite la creación y edición de una o más ontologías en un único espacio de trabajo a través de una interfaz de usuario completamente personalizable. Las herramientas de visualización permiten la navegación interactiva de las relaciones ontológicas. El soporte avanzado de explicaciones ayuda a rastrear las inconsistencias. Las operaciones de refactorización disponibles incluyen la fusión de ontologías, el traslado de axiomas entre ontologías, el cambio de nombre de múltiples entidades, etc. Entre sus principales características, cumple con los estándares del W3C, interfaz de usuario personalizable, soporte de visualización, soporte de refactorización de ontologías, interfaz directa con los razonadores, arquitectura altamente conectable. La herramienta puede ser obtenida de forma gratuita y sencilla a través del sitio web oficial <https://protege.stanford.edu/products.php>

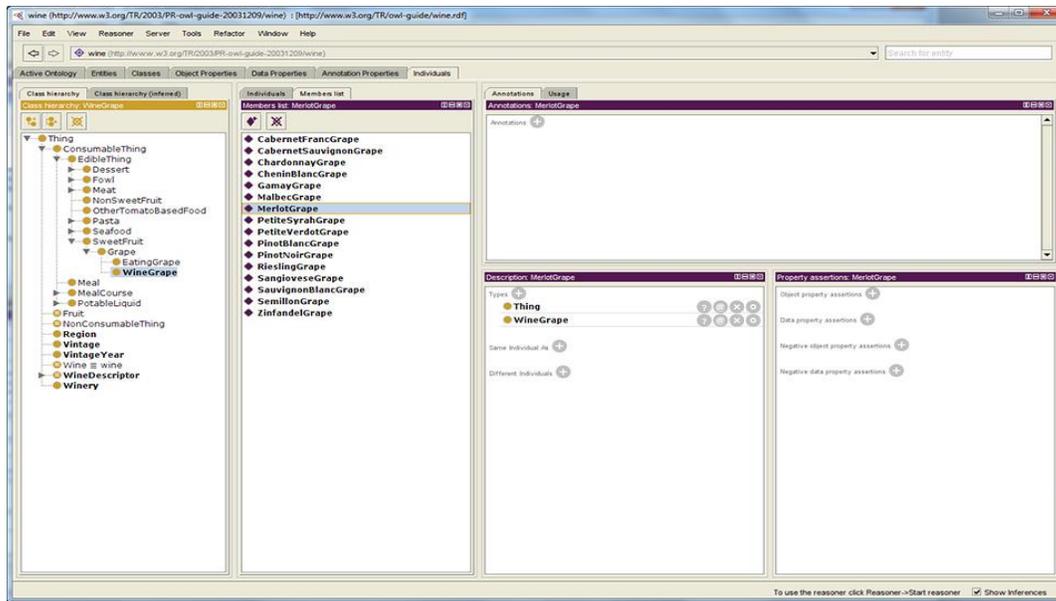


Figura 3 Interfaz de usuario de Protégé Desktop v.5.5.0 (Protégé, 2021).

Por contar con una interfaz interactiva, intuitiva y sencilla, sobre todo personalizable se utilizará Protégé Desktop en su versión 5.5.0. Como herramienta de modelado para la construcción y creación de las clases, propiedades, instancias, axiomas de la ontología que se crea con esta investigación.

1.1.6 Metodología para la construcción de ontologías

En relación con las metodologías para el diseño y construcción de ontologías el autor Bravo (2018), establece que una metodología se compone de métodos, técnicas, procesos y actividades. En esta sección se presenta una revisión de las metodologías de diseño de ontologías.

CyC: nace como un proyecto de inteligencia artificial que busca la construcción de una ontología comprensible para habilitar el razonamiento humano. El ciclo de vida propuesto es secuencial. Presenta características como, extracción manual y codificación del conocimiento, extracción semi-automática y codificación del conocimiento, y extracción completamente automática y representación del conocimiento (Echeverría, 2011).

KACTUS: el objetivo de la metodología reutiliza las ontologías existentes y evalúa la factibilidad de reutilizar conocimiento en sistemas complejos. Esta metodología estableció los siguientes procesos generales: desarrollo de la lista de requerimientos (especificación de la aplicación), identificación de los términos generales para generar un modelo preliminar, estructuración y

refinamiento de la ontología, y búsqueda de ontologías pre-existentes para reutilizarlas (Bravo, 2018).

Methontology: es una de las propuestas más completas para el desarrollo ontológico, ya que toma la creación de ontologías como un proyecto informático. Así, además de las actividades propias de la generación de la ontología, esta metodología abarca actividades para la planificación del proyecto, la calidad del resultado y la documentación. Además, permite construir ontologías totalmente nuevas o reutilizar otras ontologías. Proporciona guías sobre cómo llevar a cabo el desarrollo de una ontología mediante las actividades de especificación, conceptualización, formalización, implementación y mantenimiento (Martínez, 2018).

Las actividades de desarrollo identificadas para la metodología Methontology y por las que se realizó la ontología son las siguientes (Carrión, 2018):

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

También existen componentes de modelado de ontologías que deben describirse detalladamente:

- Tarea 1: Construir el glosario de términos.
- Tarea 2: Construir la taxonomía de conceptos.
- Tarea 3: Construir un diagrama de relaciones binarias.
- Tarea 4: Construir el diccionario de conceptos.
- Tarea 5: Definir las relaciones binarias en detalle.
- Tarea 6: Describir atributos de instancias.
- Tarea 7: Describir atributos de clase.
- Tarea 8: Describir constantes.
- Tarea 9: Describir reglas o axiomas.
- Tarea 10: Describir instancias.

3. Formalización: se encargó de la transformación del modelo conceptual generado en la actividad anterior en un modelo formal o semi-computable.
4. Implementación: construyó modelos computables en un lenguaje de ontologías (RDF Schema, OWL).
5. Mantenimiento: permitió la actualización y corrección de la ontología

De todas las metodologías analizadas para la presente investigación, Methontology, es la más completa porque propone entre sus funciones, 5 actividades fundamentales para un correcto desarrollo de las ontologías. Permite configurar el modelo de conocimientos de la aplicación web propuesta para la conceptualización de la ontología creada, además cuenta con una amplia comunidad de usuarios.

1.2 Pruebas de aceptación

El estándar IEEE 1012-1998 (IEEE, 1998) define las pruebas de aceptación como pruebas formales que se llevan a cabo para determinar si un sistema cumple o no los criterios de aceptación y permite al usuario determinar si acepta o no el sistema (López, 2016). La definición del Proceso de Pruebas de Aceptación de Software incluye descripciones gráficas y textuales, las cuales especifican un conjunto de productos de trabajo que constituyen entradas y salidas de cada actividad, al igual que los elementos de control de cada actividad. También se describen los roles involucrados y sus responsabilidades dentro del proceso (Pardo, 2013).

Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software, es necesario que las pruebas de aceptación se escriban antes de generar una sola línea de código (Pressman, 2010). Se centran en el comportamiento y capacidades de todo el sistema o producto en el entorno real. En este nivel es necesario considerar los siguientes elementos: se verifica la adecuación al uso del sistema por parte de usuarios de negocio, las pruebas se realizan utilizando el entorno del cliente, el entorno de cliente puede reproducir nuevos fallos. Entre las formas comunes de pruebas de aceptación se encuentran: pruebas de aceptación de usuarios (UAT), pruebas de aceptación de operación (OAT), pruebas de aceptación contractual o reglamentaria, pruebas alfa y beta (Marin et al., 2020).

Este tipo de pruebas pasa a ser un espacio de entendimiento para usuarios del sistema y aseguradores de la calidad. Es común que, durante su ejecución, se ponga atención a

opiniones y comentarios generales realizados por los usuarios. Para estar seguros de que todo funciona de manera correcta en la aplicación permitir a los usuarios finales probarla es de vital importancia (Velázquez, 2020).

En la siguiente figura se muestran las principales actividades y subprocesos de las pruebas de aceptación, agrupadas en tres subprocesos definidos (Pardo, 2013):

- Planificación de las Pruebas de Aceptación
- Ejecución de las Pruebas de Aceptación
- Cierre de las Pruebas de Aceptación.

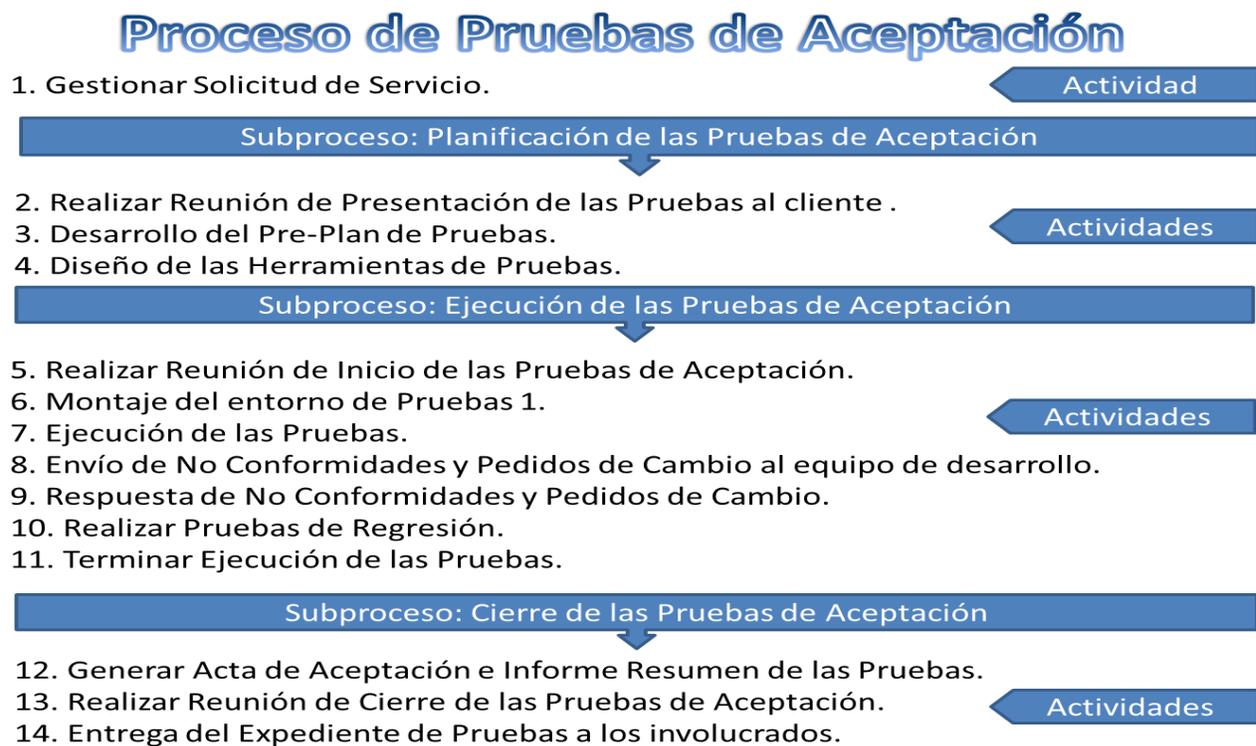


Figura 4 Descripción gráfica del Proceso de Pruebas de Aceptación de Software. Fuente: Elaboración propia.

La interacción con los clientes y usuarios finales tiene lugar desde el inicio del proceso por lo que es importante establecer formas de comunicación comunes a todos los involucrados. Las pruebas de aceptación se clasifican en dos grandes categorías, Alfa y Beta (Valerio, 2021):

- Pruebas alfa: las ejecuta el cliente con el equipo de desarrollo. Las pruebas se realizan mediante el uso regular de la solución, bajo la observación del desarrollador, quien

registra los problemas de uso y errores. Uno de los aspectos más distintivos de las pruebas alfa es que ocurren en entornos controlados.

- Pruebas beta: las realizan los usuarios finales del producto digital en sus puestos de trabajo. En otras palabras, la prueba beta es una aplicación en un entorno real, fuera del control del equipo de desarrollo. El cliente es el responsable de registrar las incidencias e informa al equipo de desarrollo. Normalmente se realiza a través de un proceso estándar de gestión de incidencias o con una solución de manejo de tickets.

Los diferentes criterios coinciden en que las pruebas de aceptación son diseñadas por el equipo de desarrollo y ejecutadas por los clientes y/o usuarios. Tienen como base los requisitos funcionales del sistema, se implementan en la etapa final del software desarrollado, cuando está listo para que los usuarios lo intervengan de una forma real.

1.2.1 Desarrollo de las pruebas de aceptación en CIDI

En la UCI a partir del año 2015 se estandarizaron los procesos. Los modelos, normas y estándares son comparados en base a los criterios: políticas, productos de trabajo, guías, roles, herramientas, recursos, indicadores y actividades, obtenidos a partir del mapa conceptual de los procesos establecidos en la Universidad (López, 2016):

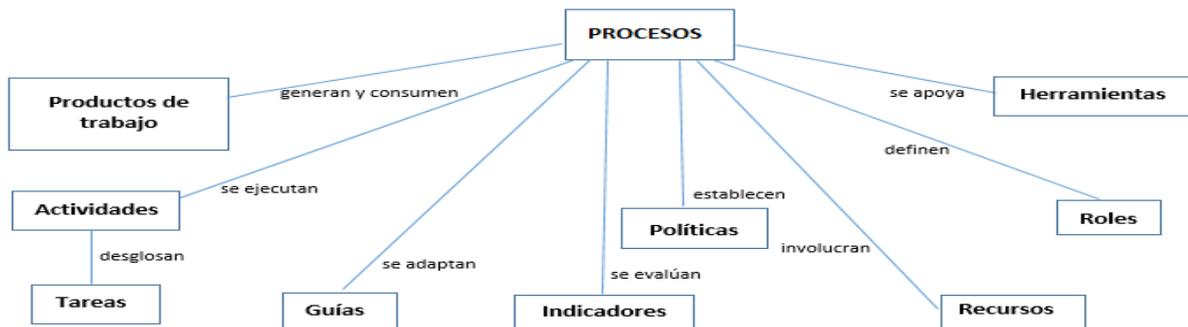


Figura 5 Modelo conceptual de procesos en la UCI (López, 2016).

Para el desarrollo de la investigación se hace necesario realizar un diagnóstico que caracterice la situación actual del proceso de desarrollo de las pruebas de aceptación con usuarios finales en CIDI. Para llevar a cabo el diagnóstico se consultaron a los equipos de desarrollo de los diferentes proyectos a través de una encuesta (ver Anexo 2). Se valoraron los aspectos:

- Momentos del desarrollo de software en que se realizan las pruebas de aceptación.

- Productos obtenidos del desarrollo de software para ejecutar las pruebas de aceptación.
- Involucrados que deben participar en el proceso de pruebas de aceptación.
- Actividades que se realizan como parte del proceso de pruebas de aceptación.
- Relación de comunicación entre los involucrados durante el proceso de aceptación.

De un total de 9 encuestados todos tienen de 6 a 13 años vinculados a la actividad productiva, 4 de ellos tienen el grado científico de máster en ciencias y entre sus roles dentro del centro están: revisor del grupo de revisiones de la dirección de calidad de software, coordinador de pruebas, programador, jefe de proyecto, probador, analista.

Los resultados obtenidos (ver Anexo 4) muestran que:

- Un 100% señala la importancia de una eficaz comunicación mediante un lenguaje común, durante las pruebas de aceptación con la participación del usuario final.
- Para las actividades de pruebas de aceptación 22% plantea que debe ser al finalizar el desarrollo del software, un 100% después que el producto final es liberado, un 11% durante todo el desarrollo, un 22% cada vez que se libera un módulo y un 11% plantean que después de los hitos fundamentales en cada etapa del desarrollo de software: requisitos, diseño, arquitectura.
- En cuanto a que productos obtenidos del proceso de desarrollo de software son sometidos a las pruebas de aceptación, se plantean que un 78% es a la documentación y un 100% a la aplicación.
- Coinciden en que los involucrados que participan en el proceso de pruebas de aceptación son los clientes con un 89%, los especialistas de otros proyectos un 11%, expertos funcionales un 33%, usuarios finales un 78%, los asesores de calidad un 11% y con un 56% coinciden con el equipo de desarrollo.
- Entre las actividades realizadas como parte de las pruebas de aceptación están las de planear las pruebas de aceptación con un 56%, realizar reunión de inicio 44%, preparar el entorno de validación 56%, diseñar prueba 33%, ejecutar validación 67%, solucionar no conformidades 67%, responder solicitudes de cambio y conciliar no conformidades 78%, conciliar respuesta a las solicitudes de cambio 44%, realizar reunión de cierre 78%, análisis de tendencias de las aceptaciones ejecutadas 33%.
- Los términos que se identifican según las experiencias que impactan de manera negativa en el entendimiento entre los involucrados durante el proceso de aceptación son la no conformidad con un 44%, los pedidos de cambio 78%, los casos de pruebas

22%, y con un 11% la no adecuada realización de la aceptación del documento especificación de requisitos.

- En cuanto al impacto de una adecuada comunicación durante las pruebas de aceptación en la calidad del producto final todos coincidieron para un 100%.

Es muy importante lograr un lenguaje común en función de mejorar el entendimiento entre los involucrados durante las pruebas de aceptación, porque en la aceptación de un producto existen personas con diferentes expectativas y experiencias en el dominio de la aplicación, a nivel profesional, como del dominio de las tecnologías. Por tanto, es muy esencial que la comunicación entre todos sea fluida y accesible para minimizar el tiempo, que permita llegar a un consenso sobre la aplicación desarrollada y garantizar la conformidad de todas las partes consensuadas minimizando los errores de los usuarios finales, así se evitan malos entendidos que dan paso a No Conformidades y Peticiones de Cambio.

Se realizaron conjuntamente entrevistas a revisor del grupo de revisiones de la dirección de calidad de software, coordinador de pruebas, programador, jefe de proyecto, probador, analista y director de CIDI, que permitieron identificar escenarios que afectan el proceso de las pruebas de aceptación con usuarios finales. Estas arrojaron las siguientes informaciones:

- Las pruebas de aceptación con la participación de los usuarios finales, que son los que van a interactuar con el producto, junto al cliente y el equipo de trabajo, son muy importantes por la información que aportan.
- De estas pruebas sale el nivel de satisfacción y si se corresponde el sistema final con lo planeado en el inicio del proyecto, además de permitir detectar no conformidades que afecten a los usuarios, sugiriendo cambios cuando ellos mismo inserten casos de prueba y vean que no son de gran utilidad o incluso que se deba agregar alguno, lo cual posibilita su resolución, garantizando que cuando el sistema sea puesto en uso, el mismo usuario pueda verificarlo y aceptarlo, logrando que el producto sea el que realmente necesiten, siendo confiable y a la medida.
- Esto se logra a través de una buena comunicación, permitiendo que todas las partes se puedan poner de acuerdo de forma más eficiente y rápida, ayudando a elevar la calidad, disminuyendo los errores que pueden salir a lo largo de todo el ciclo de desarrollo del software desde su inicio, y de esta manera se disminuye el nivel de vulnerabilidad que corresponde a cada parte, fundamentalmente a las personas que lo utilizan, o sea el usuario final.

- Como alternativas para mejorar la comunicación entre los involucrados se necesitan definir y explicar los términos a utilizar en la reunión de inicio, establecer un vocabulario único, emplear un lenguaje natural para realizar la aceptación, declarando las palabras técnicas a utilizar y explicando su significado, así como la relación que pueda existir entre ellas.
- Pueden ser de mucha ayuda los glosarios de términos, mapas conceptuales, diagramas de procesos y ontologías.
- El cliente forme parte del equipo de desarrollo con más disponibilidad de tiempo y que todos los involucrados estén presentes cuando se vayan a aplicar las pruebas.

Por eso es de vital importancia el cumplimiento del cronograma para la aplicación de las pruebas de aceptación, afectadas generalmente por el no cumplimiento de alguna de las partes, no se realizan pruebas automatizadas a los sistemas, la no comprensión de los términos requisitos funcionales y no funcionales, la inadecuada selección de los especialistas expertos funcionales para realizar el levantamiento y aceptación de los requisitos del sistema, la firma después de una revisión inadecuada, o la no presencia en tiempo para la misma.

1.3 Conclusiones del capítulo

- Varios autores coinciden que las ontologías son necesarias para la organización y exposición de la información y el conocimiento, en este capítulo se define el concepto, tipología, estructura, herramienta, lenguaje y metodología para la propuesta de solución.
- El proceso de pruebas de aceptación debe implementarse durante el ciclo de vida del proyecto de software, con el fin de disminuir los defectos en el software que es entregado al cliente y cumplir sus expectativas.
- Se analizó el proceso (subproceso y actividades) de las pruebas de aceptación y se aplicó una encuesta a jefes de proyectos y miembros de equipos de desarrollo en CIDI, logrando un mayor entendimiento para desarrollar la propuesta de solución.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN E IMPLEMENTACIÓN

En este capítulo se presenta la estructura de la propuesta de solución para entender mejor la ontología que se va a desarrollar. Se realiza una descripción de la propuesta de solución entre otros elementos propios de la fase de análisis y diseño, además del despliegue de las actividades de desarrollo y los componentes para el modelado de la ontología, a través de las herramientas mencionadas en el anterior capítulo por medio de la metodología Methontology. Esta propuesta ontológica tiene como objetivo recuperar y organizar todo lo relacionado con el proceso de las pruebas de aceptación en CIDI.

2.1. Descripción de la propuesta de solución

La ontología propuesta, tiene como fin describir y representar el conocimiento y la comprensión de la información que existe en el proceso de pruebas de aceptación con usuarios finales en CIDI, logrando un mayor entendimiento de dicho proceso entre las partes involucradas. En ella se describen los términos, relaciones, conceptos e instancias dentro de dicho proceso, proporcionando a los que realizan las pruebas de aceptación un lenguaje natural y una unificación del vocabulario. Para ello sigue el flujo de actividades representado en la figura (ver figura 6).

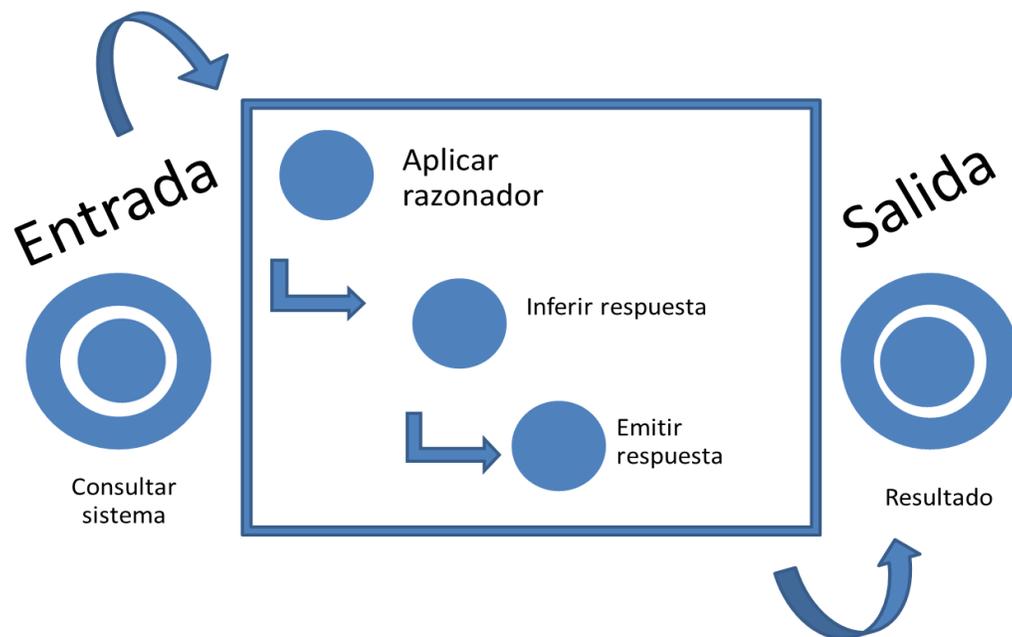


Figura 6 Flujo de actividades. Fuente: Elaboración propia.

A continuación, se muestra una breve descripción con las principales clases y relaciones del proceso de pruebas de aceptación:

Las actividades de control de la calidad se clasifican en actividades de especificación y actividades de validación. El proceso de pruebas de aceptación pertenece a las de validación y tiene tres fases que se aplican con el fin de saber el grado de aceptación del producto en los clientes y usuarios finales. Estas fases contienen evaluación, ejecución y diseño, esta última aplica dos métodos de pruebas, el de caja negra y caja blanca, los cuales tienen tres técnicas de evaluación que son asociadas a los métodos de caja negra, asociadas a los métodos de caja blanca y asociadas a las pruebas de aceptación. En la fase de evaluación se validan los requisitos funcionales y los requisitos no funcionales, mientras que la ejecución revisa estos mismos requisitos, revisa la aplicación, y define los resultados que tienen las peticiones de cambio y las no conformidades. Las fases de diseño y ejecución utilizan herramientas automatizadas y dos técnicas manuales, lista de chequeo y diseño de casos de prueba.

Los usuarios finales, los clientes y los especialistas de proyecto, realizan dichas pruebas de aceptación, los especialistas de proyecto generan un documento con el resultado de esas revisiones y desarrollan el diseño. Los usuarios finales y el cliente participan en el diseño y desarrollan la ejecución.

En la siguiente figura (ver figura 7), confeccionado con la herramienta Visual Paradigm for UML 8.0 Enterprise Edition, se muestra el modelo conceptual, en la misma se representan las clases relaciones y la multiplicidad que se utilizaron para el diseño de la ontología.

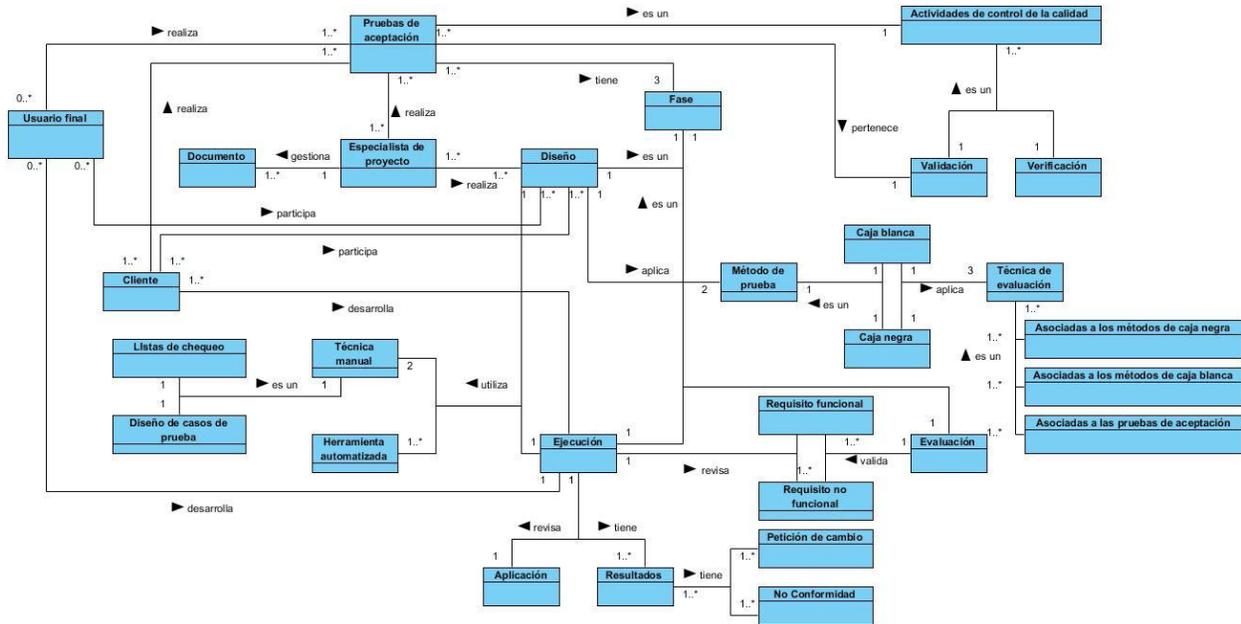


Figura 7 Modelo conceptual del Proceso de Evaluación de Software. Fuente: Elaboración propia.

2.2. Aplicación de la metodología Methontology

2.2.1 Especificación

La primera actividad de la metodología Methontology para hacer la ontología es la especificación, se creó una plantilla (ver tabla 1) 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 (Carrión, 2018).

Tabla 1 Documento de Especificación de Requerimientos.

Dominio	Proceso de pruebas de aceptación en CIDI.
Fecha	1 de septiembre del 2021
Desarrollador(es)	Alexis Edgardo Guerrero Herrera.
Propósito	Desarrollar una propuesta ontológica que tiene como objetivo describir y representar el conocimiento y la comprensión de la información que existe en el proceso de pruebas de aceptación con usuarios finales en CIDI.
Nivel de Formalidad	Formal

Alcance	¿Qué especialista de proyecto gestiona el doc1, doc2, doc3? ¿Qué usuario registra el doc2, doc3? ¿Qué especialista de proyecto realiza el diseño1? ¿Qué usuario revisa la aplicación1? ¿Qué cliente desarrolla la ejecución1? ¿Qué especialista de proyecto utiliza la herramienta automatizadas1? ¿Qué usuario emite la no conformidad1? ¿Qué especialista de proyecto valida los requisitos no funcionales1?
Fuentes de Conocimiento	Jefes de equipo, líderes de proyecto y equipos de desarrollo en CIDI.

Fuente: Elaboración propia.

2.2.2 Conceptualización

Para la conceptualización de la ontología se definen y describen detalladamente diez tareas para el modelado de ontologías que compone esta actividad. Se basan en organizar y convertir una percepción informal del dominio en una especificación semi-formal que pueden ser fácilmente comprendidas.

Tarea 1: Construir el glosario de términos.

La elaboración del glosario de términos es la tarea que define cada término importante del dominio incluyendo las relaciones, instancias y conceptos (ver tabla 2).

Tabla 2 Glosario de términos.

No.	Término	Descripción	Tipo
1.	Prueba de aceptación	Prueba formal que se lleva a cabo para determinar si un sistema cumple o no los criterios de aceptación y permite al usuario determinar si acepta o no el sistema.	Concepto
2.	Usuario final	Persona encargada de realizar las pruebas de aceptación.	Concepto
3.	Cliente	Persona encargada de realizar las pruebas de aceptación y coordinar con el especialista de proyecto.	Concepto
4.	Especialista de proyecto	Persona encarga de coordinar la evaluación en la Dirección de la Calidad de Software.	Concepto
5.	Evaluación	Determinación sistemática del grado en que una entidad cumple con los criterios especificados.	Concepto
6.	Método de prueba	Forma de cómo ejecutar las pruebas.	Concepto
7.	Caja blanca	Se basa en el examen cercano de los detalles de procedimiento del software.	Concepto
8.	Caja negra	Se refiere a las pruebas que se llevan a cabo en la interfaz del software.	Concepto
9.	Técnica de evaluación	Métodos y habilidades necesarios para llevar a cabo una actividad específica.	Concepto
10.	Asociadas a los métodos de caja blanca	Técnica de evaluación para los métodos de caja blanca	Concepto
11.	Asociadas a los métodos de caja negra	Técnica de evaluación para los métodos de caja negra	Concepto

12.	Asociadas a las pruebas de aceptación	Técnica de evaluación para las pruebas de aceptación	Concepto
13.	Fase	Etapas por las que transita el procedimiento de pruebas.	Concepto
14.	Validación	Conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.	Concepto
15.	Verificación	Conjunto de diferentes actividades que demuestran que el software funciona correctamente.	Concepto
16.	Requisitos funcionales	Describen las funciones que lleva a cabo el software, como debe reaccionar ante ciertas entradas y como debe comportarse en situaciones particulares.	Concepto
17.	Requisitos no funcionales	Definen cualidades o atributos globales del sistema, o establecen restricciones sobre el producto desarrollado, el proceso de desarrollo o externas.	Concepto
18.	Ejecución	Realización de una acción para cumplir un proyecto.	Concepto
19.	Documento	Documentos generados en cada una de las actividades del proceso de evaluación.	Concepto
20.	Aplicación	Programa informático diseñado como una herramienta para realizar operaciones o funciones específicas.	Concepto
21.	Diseño	Proceso de planificación para darle una solución al problema.	Concepto
22.	Herramienta automatizada	Instrumento que puede utilizarse durante la evaluación para recolectar datos, simplificar el tiempo y el esfuerzo, y lograr que los resultados sean óptimos y exactos.	Concepto
23.	Técnica manual	Técnica que puede utilizarse durante la evaluación para recolectar datos y llevar a cabo la interpretación de los mismos.	Concepto
24.	Listas de chequeo	Se definen como un listado de preguntas en forma de cuestionario, que sirven para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado.	Concepto
25.	Diseño de casos de prueba	Constituyen una herramienta necesaria para la ejecución de las pruebas funcionales. Se definen como un conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados con el propósito de identificar y comunicar las condiciones que se llevarán a cabo en la prueba.	Concepto
26.	Resultados	Informe de evaluación de la calidad del producto de software.	Concepto
27.	Petición de cambio	Es una modificación que solicita la parte cliente a un requisito o un elemento pactado con anterioridad y se relacionan también a la incorporación de nuevos requisitos o elementos.	Concepto
28.	No conformidad	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
29.	Actividades de control de la calidad	Conjunto de actividades que se desarrollan a lo largo del proceso de desarrollo del software.	Concepto
30.	prueba1	Instancia perteneciente a la clase Prueba de aceptación.	Instancia
31.	usuario1	Instancia perteneciente a la clase Usuario final.	Instancia
32.	cliente1	Instancia perteneciente a la clase Cliente.	Instancia
33.	analista1	Instancia perteneciente a la clase Especialista de proyecto.	Instancia
34.	jefe1	Instancia perteneciente a la clase Especialista de proyecto.	Instancia
35.	administrador_de_calidad1	Instancia perteneciente a la clase Especialista de proyecto.	Instancia
36.	programador1	Instancia perteneciente a la clase Especialista de proyecto.	Instancia
37.	evaluación1	Instancia perteneciente a la clase Evaluación.	Instancia
38.	mp1	Instancia perteneciente a la clase Métodos de prueba.	Instancia
39.	mp2	Instancia perteneciente a la clase Métodos de prueba.	Instancia
40.	te1	Instancia perteneciente a la clase Técnicas de evaluación.	Instancia
41.	te2	Instancia perteneciente a la clase Técnicas de evaluación.	Instancia
42.	te3	Instancia perteneciente a la clase Técnicas de evaluación.	Instancia
43.	ejecución1	Instancia perteneciente a la clase Ejecución.	Instancia
44.	doc1	Instancia perteneciente a la clase Documento.	Instancia
45.	doc2	Instancia perteneciente a la clase Documento.	Instancia
46.	doc3	Instancia perteneciente a la clase Documento.	Instancia

47.	aplicación1	Instancia perteneciente a la clase Aplicación.	Instancia
48.	diseño1	Instancia perteneciente a la clase Diseño.	Instancia
49.	ha1	Instancia perteneciente a la clase Herramienta Automatizada.	Instancia
50.	tm1	Instancia perteneciente a la clase Técnica manual.	Instancia
51.	tm2	Instancia perteneciente a la clase Técnica manual.	Instancia
52.	pc1	Instancia perteneciente a la clase Petición de cambio.	Instancia
53.	nc1	Instancia perteneciente a la clase No conformidad.	Instancia
54.	requisitoF1	Instancia perteneciente a la clase Requisito funcional	Instancia
55.	requisitoNF1	Instancia perteneciente a la clase Requisito no funcional.	Instancia
56.	es un actividades de control de la calidad	Es la relación que existe entre los conceptos de Validación y Actividades de control de la calidad.	Relación
57.	es un actividades de control de la calidadV	Es la relación que existe entre los conceptos de Verificación y Actividades de control de la calidad.	Relación
58.	es un actividades de control de la calidadP	Es la relación que existe entre los conceptos de Pruebas de aceptación y Actividades de control de la calidad.	Relación
59.	pertenece validación	Es la relación que existe entre los conceptos de Pruebas de aceptación y Validación.	Relación
60.	tiene fase	Es la relación que existe entre los conceptos de Pruebas de aceptación y Fase.	Relación
61.	es un fase	Es la relación que existe entre los conceptos de Diseño y Fase.	Relación
62.	es un faseE	Es la relación que existe entre los conceptos de Ejecución y Fase.	Relación
63.	es un faseEV	Es la relación que existe entre los conceptos de Evaluación y Fase.	Relación
64.	aplica métodos de prueba	Es la relación que existe entre los conceptos de Diseño y Métodos de prueba.	Relación
65.	es un métodos de prueba	Es la relación que existe entre los conceptos de Caja blanca y Métodos de prueba.	Relación
66.	es un métodos de pruebaC	Es la relación que existe entre los conceptos de Caja negra y Métodos de prueba.	Relación
67.	aplica técnicas de evaluación	Es la relación que existe entre los conceptos de Caja blanca y Técnicas de evaluación.	Relación
68.	aplica técnicas de evaluaciónC	Es la relación que existe entre los conceptos de Caja negra y Técnicas de evaluación.	Relación
69.	es un técnicas de evaluación	Es la relación que existe entre los conceptos de Asociadas a los métodos de caja blanca y Técnicas de evaluación.	Relación
70.	es un técnicas de evaluaciónA	Es la relación que existe entre los conceptos de Asociadas a los métodos de caja negra y Técnicas de evaluación.	Relación
71.	es un técnicas de evaluaciónP	Es la relación que existe entre los conceptos de Asociadas a las Pruebas de aceptación y Técnicas de evaluación.	Relación
72.	valida requisitos funcionales	Es la relación que existe entre los conceptos de Evaluación y Requisitos funcionales.	Relación
73.	valida requisitos no funcionales	Es la relación que existe entre los conceptos de Evaluación y Requisitos no funcionales.	Relación
74.	revisa requisitos funcionales	Es la relación que existe entre los conceptos de Ejecución y Requisitos no funcionales.	Relación
75.	revisa requisitos no funcionales	Es la relación que existe entre los conceptos de Ejecución y Requisitos no funcionales.	Relación
76.	Tiene resultados	Es la relación que existe entre los conceptos de Ejecución y Resultados.	Relación
77.	tiene petición de cambio	Es la relación que existe entre los conceptos de Resultados y Petición de cambio.	Relación
78.	tiene no conformidad	Es la relación que existe entre los conceptos de Resultados y No conformidad.	Relación
79.	revisa aplicación	Es la relación que existe entre los conceptos de Ejecución y Aplicación.	Relación
80.	utiliza técnicas manuales	Es la relación que existe entre los conceptos de Ejecución y Técnicas manuales.	Relación
81.	utiliza técnicas manualesD	Es la relación que existe entre los conceptos de Diseño y Técnicas manuales.	Relación
82.	Utiliza herramientas	Es la relación que existe entre los conceptos de Ejecución y	Relación

	automatizadas	Herramientas automatizadas.	
83.	Utiliza herramientas automatizadasD	Es la relación que existe entre los conceptos de Diseño y Herramientas automatizadas.	Relación
84.	es un técnicas manuales	Es la relación que existe entre los conceptos de Listas de chequeo y Técnicas manuales.	Relación
85.	es un técnicas manualesD	Es la relación que existe entre los conceptos de Diseños de casos de prueba y Técnicas manuales.	Relación
86.	participa diseño	Es la relación que existe entre los conceptos de Usuario final y Diseño.	Relación
87.	participa diseñoC	Es la relación que existe entre los conceptos de Cliente y Diseño.	Relación
88.	realiza diseño	Es la relación que existe entre los conceptos de Especialista de proyecto y Diseño.	Relación
89.	desarrolla ejecución	Es la relación que existe entre los conceptos de Usuario final y Ejecución.	Relación
90.	desarrolla ejecuciónC	Es la relación que existe entre los conceptos de Cliente y Ejecución.	Relación
91.	gestiona documento	Es la relación que existe entre los conceptos de Especialista de proyecto y Documento.	Relación
92.	realiza pruebas de aceptación	Es la relación que existe entre los conceptos de Usuario final y Pruebas de aceptación.	Relación
93.	realiza pruebas de aceptaciónC	Es la relación que existe entre los conceptos de Cliente y Pruebas de aceptación.	Relación
94.	realiza pruebas de aceptaciónE	Es la relación que existe entre los conceptos de Especialista de proyecto y Pruebas de aceptación.	Relación

Fuente: Elaboración propia.

Tarea 2: Construir la taxonomía de conceptos.

Al finalizar el glosario de términos con una cantidad significativa de elementos, se define una taxonomía que define la jerarquía entre los conceptos. Se debe evaluar que la taxonomía creada no contenga errores. Las taxonomías representan a través de las clases y las instancias propias de esas clases una realidad contextualizada que permitió la organización y recuperación efectiva de los conceptos representados pertenecientes a un determinado dominio. La taxonomía es la ciencia que trata los principios, métodos y fines de la clasificación. Para construir la taxonomía de conceptos, se seleccionaron del glosario de términos aquellos términos que son conceptos, por lo que hay que tener en cuenta que un concepto puede ser subclase de más de un concepto en la taxonomía (Valdivieso, 2020). En las siguientes figuras (Ver figuras 8) se muestran las taxonomías creadas. Al concluir se debe evaluar que la taxonomía que se crea no tenga errores, ejemplo: clasificación de conceptos incompleta, herencias cíclicas, errores de partición, errores semánticos, etc.

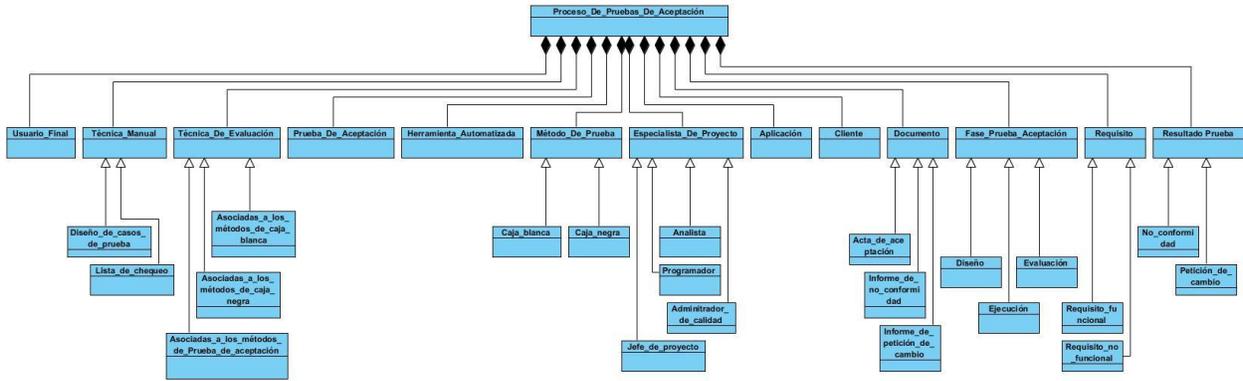


Figura 8 Taxonomía de los conceptos Fuente: Elaboración propia.

Tarea 3: Construir un diagrama de relaciones binarias.

Después de haber construido y evaluado la taxonomía, la actividad de conceptualización plantea construir diagramas de relaciones binarias. El propósito de este diagrama es definir las relaciones existentes entre conceptos de una o más taxonomías de conceptos. En el diagrama de clases mostrado anteriormente todas las relaciones que están son relaciones binarias, pero en la siguiente figura (Ver figura 9) se muestra una relación binaria inversa de uno de los conceptos del diagrama, Métodos de prueba y Diseño.



Figura 9 Diagrama de relaciones binarias de la ontología Proceso de Pruebas de aceptación. Fuente: Elaboración propia.

Tarea 4: Construir el diccionario de conceptos.

Una vez que las taxonomías de conceptos y los diagramas de relaciones binarias han sido generados, el desarrollador de la ontología 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.

Tabla 3 Diccionario de conceptos

Nombre del concepto	Instancia	Atributos de clase	Atributos de instancia	Relaciones
Prueba_De_Aceptación	prueba1		fecha_de_inicio_de_la_prueba_de_aceptación fecha_fin_de_la_prueba_de_aceptación	es_realizado_por
Especialista_De_Proyecto	administrador_d e_calidad1 analista1 programador1 jefe1		descripción_del_especialista responsabilidad_del_especialista nombre_del_especialista	utiliza valida realiza gestiona
Documento	doc1 doc2 doc3		descripción_del_documento denominación_del_documento	es_registrado_por es_gestionado_por
Usuario_Final	usuario1		nombre_del_usuario_final descripción_del_usuario_final	desarrolla emite realiza_prueba_de_aceptación registra revisa utiliza
Cliente	cliente1		nombre_del_cliente descripción_del_cliente	desarrolla emite realiza_prueba_de_aceptación registra revisa utiliza participa
Fase_Prueba_Aceptación	diseño1 evaluación1 ejecución1		denominación_de_la_fase descripción_de_la_fase fecha_de_inicio_de_la_fase fecha_fin_de_la_fase sesión_de_trabajo_de_la_fase responsable_de_la_fase participante_de_la_fase observación_de_la_fase	es_realizado_por es_asistido_por aplica es_desarrollado_por
Método_De_Prueba	mp1 mp2		denominación del Método_De_Prueba descripción del Método_De_Prueba	es_aplicado_por
Técnica_De_Evaluación	te1 te2 te3		denominación de la técnica_de_evaluación descripción de la técnica_de_evaluación	es_aplicado_por
Resultado_Prueba	nc1		descripción del resultado prueba	es_emitido_por

a	pc1		ba número_de_iteración_del_resultado_prueba	
Herramienta_Automatizada	ha1		denominación de la herramienta automatizada descripción de la herramienta automatizada versión de la herramienta automatizada	es_utilizado_por
Técnica_Manual	tm1 tm2		denominación de la técnica manual descripción de la técnica manual	es_utilizado_por
Aplicación	aplicación1		denominación de la aplicación descripción de la aplicación	es_revisado_por
Requisito	requisito1 requisitonf1		descripción del requisito número de iteración del requisito	es_revisado_por es_validado_por

Fuente: Elaboración propia.

Tarea 5: Describir las relaciones binarias en detalle.

Después de haber construido el diccionario de concepto, se procedió a definir las relaciones binarias en detalle. El objetivo de esta tarea es detallar todas las relaciones binarias descritas en el diagrama de relaciones binarias e incluidas en el diccionario de conceptos. Para cada relación binaria, se especifica su nombre, los nombres de sus conceptos origen y destino, su cardinalidad y su relación inversa, si existe. En la siguiente tabla (ver tabla 4) se definen las relaciones binarias de la ontología.

Tabla 4 Relación binaria de la ontología Proceso de Pruebas de Aceptación.

Nombre de la relación	Concepto Origen	Cardinalidad	Concepto Destino	Relación Inversa
aplica	Diseño	1...n	Método_De_Prueba Técnica_De_Evaluación	es_aplicado_por
desarrolla	Cliente Usuario_Final	1	Ejecución	es_desarrollado_por
emite	Cliente Usuario_Final	1...n	Resultado_Prueba	es_emitido_por
gestiona	Jefe_de_proyecto	3	Documento	es_gestionado_por
participa	Cliente	1	Diseño	es_asistido_por
realiza	Analista	1	Diseño	es_realizado_por
realiza_prueba_de_aceptación	Cliente Usuario_Final	1...n	Prueba_De_Aceptación	es_realizada_por
registra	Cliente Usuario_Final	1...n	Resultado_Prueba	es_registrado_por
revisa	Cliente Usuario_Final	1...n	Requisito Aplicación	es_revisado_por
tiene_asociada_a_caja_blanca	Técnica_De_Evaluación	1...n	Asociadas_a_los_métodos_de_caja	es_técnica_de_evaluación

			blanca	
tiene_asociada_a_caja_negra	Técnica_De_Evaluación	1...n	Asociadas_a_los_métodos_de_caja_negra	es_tecnica_de_evaluación
tiene_asociado_a_prueba_de_aceptacion	Técnica_De_Evaluación	1...n	Asociadas_a_las_Pruebas_de_aceptación	es_tecnica_de_evaluación
tiene_caja_blanca	Método_De_Prueba	1...n	Caja_blanca	es_metodo_de_prueba
tiene_caja_negra	Método_De_Prueba	1...n	Caja_negra	es_metodo_de_prueba
tiene_diseño	Fase_Prueba_Aceptación	1	Diseño	es_fase
tiene_ejecución	Fase_Prueba_Aceptación	1	Ejecución	es_fase
tiene_evaluación	Fase_Prueba_Aceptación	1	Evaluación	es_fase
tiene_diseño_de_caso_de_prueba	Técnica_Manual	1...n	Diseño_de_caso_de_prueba	es_técnica_manual
tiene_lista_de_chequeo	Técnica_Manual	1...n	Lista_de_chequeo	es_técnica_manual
tiene_fase	Prueba_De_Aceptación	3	Fase_Prueba_Aceptación	es_fase
tiene_no_conformidad	Resultado_Prueba	1...n	No_conformidad	es_resultado
tiene_petición_de_cambio	Resultado_Prueba	1...n	Petición_de_cambio	es_resultado
utiliza	Cliente Usuario_Final Especialista_De_Proyecto	1...n	Herramienta_automatizada Técnica_Manual	es_utilizado_por
valida	Analista	1...n	Requisito	es_validado_por

Fuente: Elaboración propia.

Tarea 6: Describir atributos de instancias.

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

Tabla 5 Descripción de los atributos de instancias.

Atributo de instancia	Concepto	Tipo de valor	Rango	Cardinalidad
Descripción	Especialista_De_Proyecto Documento Usuario_Final Cliente Diseño Evaluación Método_De_Prueba	string	-	1...1

	Ejecución Técnica_de_Evaluación Resultados No_conformidad Aplicación Herramienta_Automatizada Petición_de_cambio Técnica_Manual			
Nombre	Especialista_De_Proyecto Usuario_Final Cliente	string	-	1...1
Versión	Herramienta_Automatizada	string	-	1...1
Denominación	Documento Diseño Evaluación Método_De_Prueba Ejecución Técnica_de_Evaluación Herramienta_Automatizada Técnica_Manual Aplicación	string	-	1...1
Responsabilidad	Especialista_De_Proyecto	string	-	1...1
Número_de_iteración	No_conformidad Petición_de_cambio	int	-	1...1
Fecha_Inicio	Prueba_De_Aceptación Diseño Evaluación Ejecución	string	-	1...1
Fecha_Fin	Prueba_De_Aceptación Diseño Evaluación Ejecución	string	-	1...1
Sesiones_de_trabajo Responsable Participante Observación	Diseño Evaluación Ejecución	string	-	1...1

Fuente: Elaboración propia.

Tarea 7: Describir atributos de clases.

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

Tarea 8: Describir constantes.

El objetivo principal de esta actividad es describir cada una de las constantes identificadas en el glosario de términos, donde solamente fueron identificados términos de tipo concepto y relaciones. Para cada constante se especificó el nombre, tipo de valor, valor y unidad de medida (para constantes numéricas). En esta ontología no se hace uso de constantes, debido a que el conocimiento almacenado en la misma, está estructurado a través de los conceptos identificados y sus relaciones, los cuales no hacen uso de constantes.

Tarea 9: Describir reglas o axiomas.

Ahora se definen las 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. Los axiomas se muestran en la tabla (ver tabla 6).

Tabla 6 Ejemplo de reglas en la ontología en el proceso de pruebas de aceptación.

Nombre de la regla	Descripción	Expresión	Conceptos	Relaciones
Acciones del usuario final	Usuario final realiza las pruebas de aceptación, registra los resultados en el documento, revisa los requisitos, utiliza técnicas manuales y herramientas automatizadas, desarrolla la ejecución y revisa la aplicación.	and (realiza_prueba_de_aceptación some Prueba_De_Aceptación) and (registra some Documento) and (revisa some Requisito) and (utiliza some Herramienta_Automatizada) and (utiliza some Tecnica_Manual) and (desarrolla only Ejecución) and (revisa only Aplicación)	Prueba_De_Aceptación Documento Requisito Herramienta_Automatizada Tecnica_Manual Ejecucion Aplicacion	desarrolla emite realiza_prueba_de_aceptación registra revisa utiliza
Acciones del cliente	Cliente realiza las pruebas de aceptación, registra los resultados en el documento, revisa los requisitos, utiliza técnicas manuales y herramientas automatizadas,	and (emite some Resultado_Prueba) and (registra some Documento) and (revisa some Requisito) and (utiliza some Herramienta_Automatizada) and (utiliza some Tecnica_Manual) and (desarrolla only Ejecucion) and (revisa only Aplicacion) and (participa min 1 Diseño)	Prueba_De_Aceptación Documento Requisito Herramienta_Automatizada Tecnica_Manual Ejecucion Aplicación Diseño	desarrolla emite realiza_prueba_de_aceptación registra revisa utiliza participa

	desarrolla la ejecución, participa en el diseño y revisa la aplicación.	and (realiza prueba de aceptación min 1 Prueba_De_Aceptación)		
Acciones del diseño	Diseño aplica métodos de prueba y técnicas de evaluación.	and (aplica some Método_De_Prueba) and (aplica some Técnica_De_Evaluación)	Método_De_Prueba Técnica_De_Evaluación	aplica
Gestión documental y manejo de herramienta del especialista Jefe de proyecto	Especialista Jefe de proyecto gestiona algún documento y utiliza herramientas automatizadas.	and (utiliza some Herramienta_Automatizada) and (utiliza some Técnica_Manual) and (gestiona exactly 1 Acta_de_aceptación) and (gestiona exactly 1 Informe_de_no_conformidad) and (gestiona exactly 1 Informe_de_petición_de_cambio)	Herramienta_Automatizada Técnica_Manual Acta_de_aceptación Informe_de_no_conformidad Informe_de_petición_de_cambio	gestiona utiliza
Acciones del especialista Analista	Especialista Analista utiliza herramientas automatizadas y técnicas manuales, valida requisitos y realiza el diseño de las pruebas de aceptación	and (utiliza some Herramienta_Automatizada) and (utiliza some Técnica_Manual) and (valida some Requisito) and (realiza min 1 Diseño)	Herramienta_Automatizada Técnica_Manual Requisito Diseño	Utiliza Valida realiza

Fuente: Elaboración propia.

Tarea 10: Describir instancias.

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

Al finalizar la actividad de conceptualización la ontología cuenta con 34 clases, 40 relaciones, 55 atributos y 26 individuos.

2.2.3 Formalización

A partir de esta actividad, se transformó el modelo conceptual que se construyó en las actividades anteriores a un modelo formal o semi-computable. Es decir, llevar los conceptos y las relaciones, atributos e instancias a dicho modelo. Para ello se hace uso del editor de ontologías Protégé 5.5. De la ontología desarrollada, es válido especificar que la flexibilidad del

método permite que la misma sea actualizada en caso de que aparezcan nuevas tendencias en CIDI y su manera de realizar las pruebas aceptación con usuarios finales.

Creación de las clases

Las clases son creadas según la jerarquía definida en la taxonomía de conceptos, partiendo desde la clase padre “Thing”, que es una palabra clave que hace referencia a la W3C¹. Cada clase presenta: anotaciones, clases equivalentes, sub-clases, instancias y restricciones. En la figura (ver figura 10), se puede apreciar la creación de las clases.

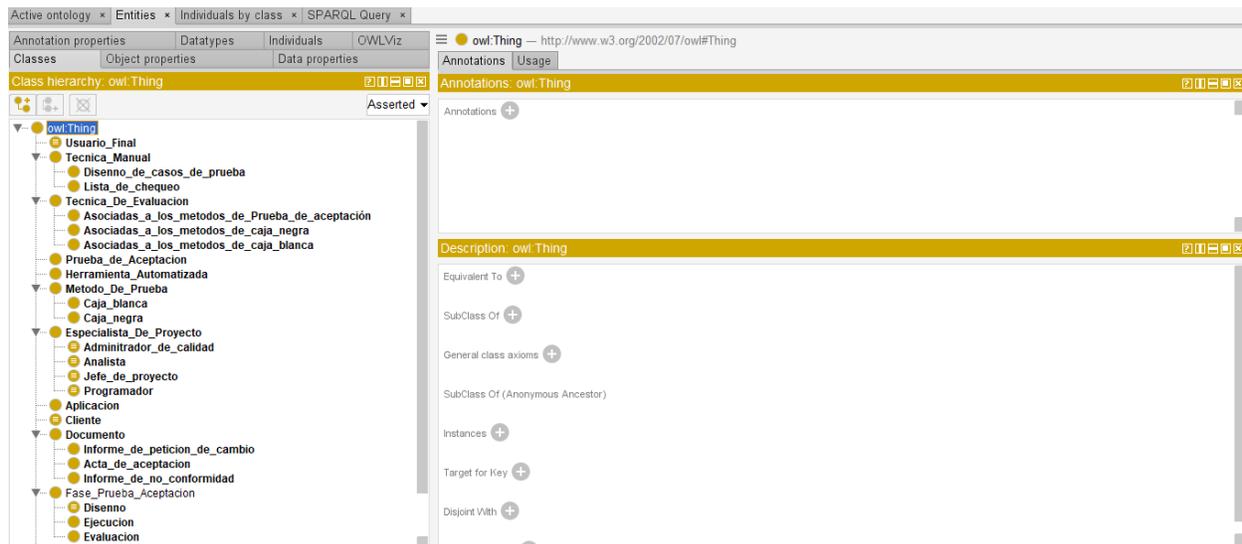


Figura 10 Definición de las clases creadas en Protégé. Fuente: Elaboración propia.

Creación de las propiedades de las clases

Cuando las entidades de la ontología estén identificadas y modeladas en la herramienta se establecieron las relaciones semánticas entre cada una de ellas mediante el uso de propiedades de objetos (Object Properties). Las propiedades son los elementos ontológicos que permiten definir las relaciones entre los conceptos del dominio de la ontología. Estas describen las cualidades internas de los conceptos y representan las propiedades diferentes de los objetos.

¹World Wide Web Consortium, es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

Existen dos tipos principales de propiedades: los “Object Properties” y “Data type Properties”. Los Object Properties sirven para definir propiedades que conectan una clase con otra, mientras que los Data type Properties se utilizan para definir propiedades que conectan una clase con un tipo de dato. Mientras se añaden las Object Properties deben definirse las propiedades inversas de estas, también se especifica el dominio y rango, o sea las clases que se relacionan. En la figura (ver figura 11) se pueden apreciar las propiedades de las clases y en la figura (ver figura 12) la creación de las propiedades.

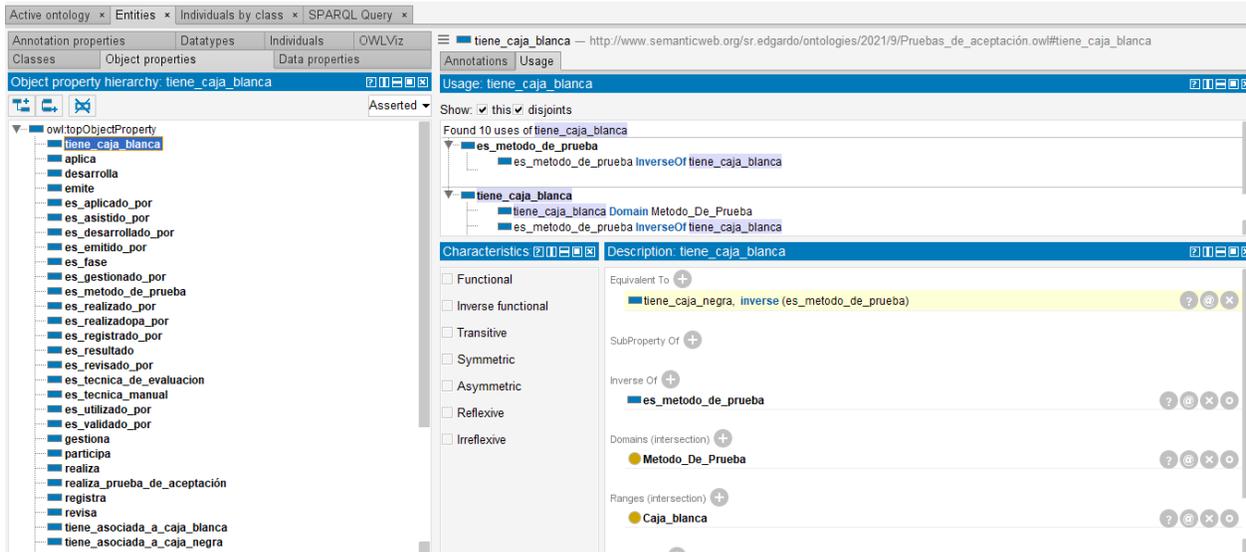


Figura 11 Object Properties (Propiedades de las clases creadas en Protégé. Fuente: Elaboración propia.

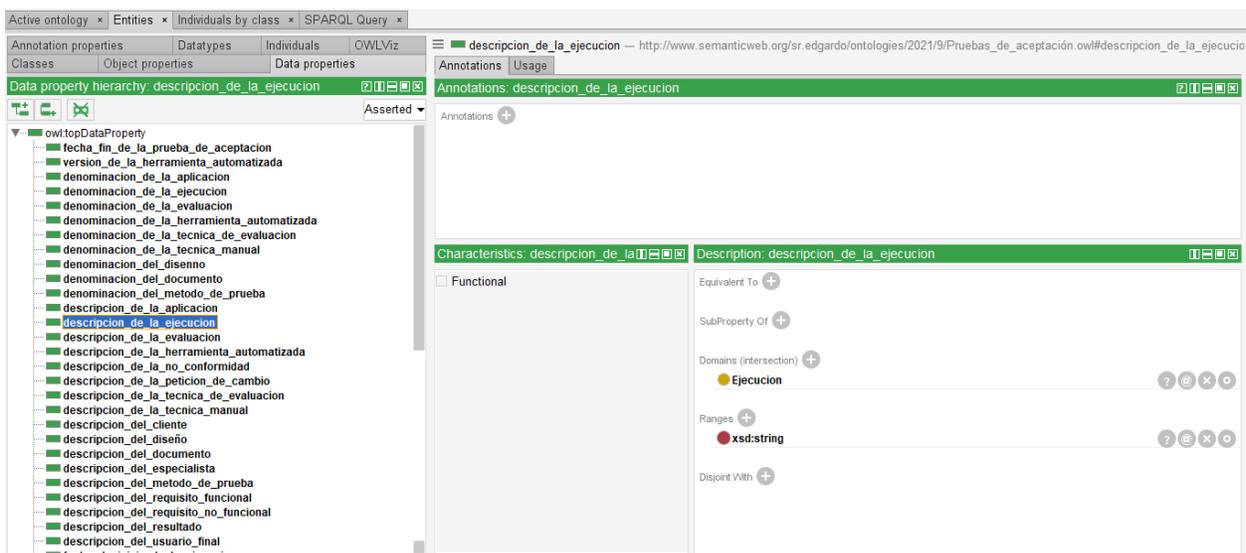


Figura 12 Creación de la Data Properties en Protégé. Fuente: Elaboración propia.

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 (ver figura 13) se puede observar las reglas descritas para la clase Jefe de proyecto.

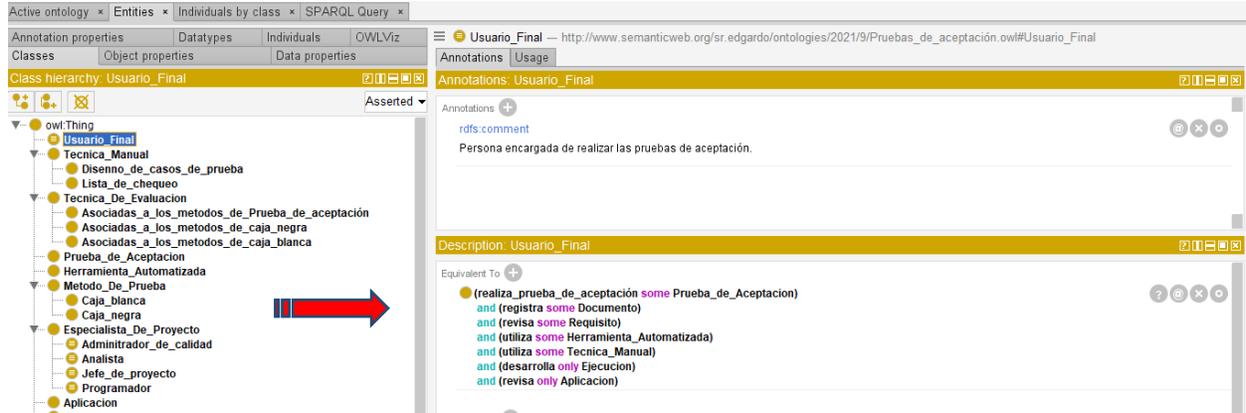


Figura 13 Creación de las reglas. Fuente: Elaboración propia.

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 (ver figura 14).

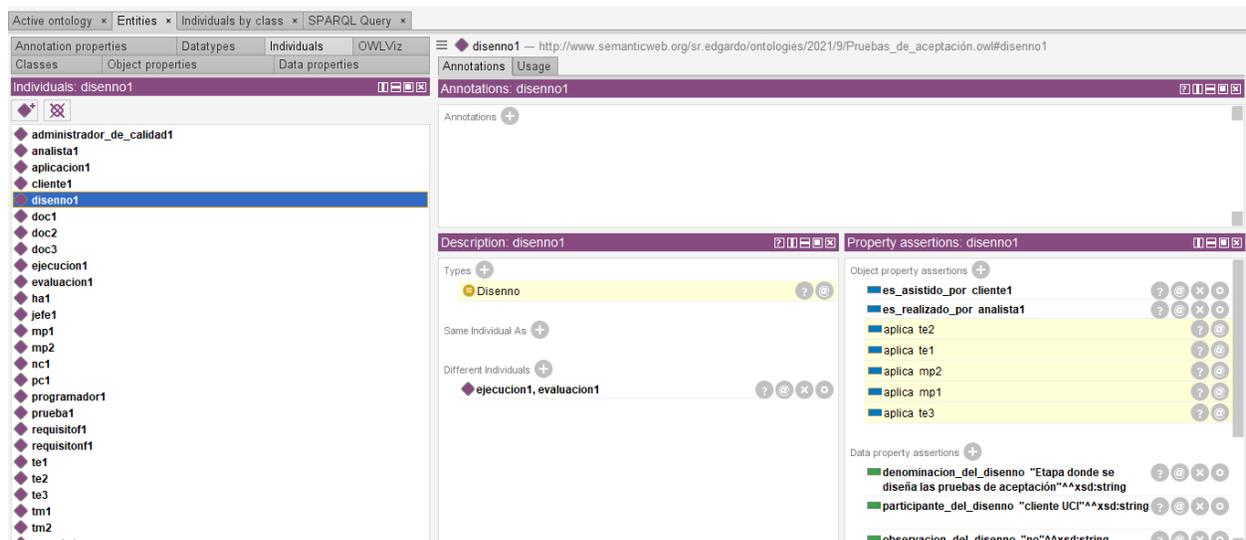


Figura 14 Creación de las instancias con Protégé. Fuente: Elaboración propia.

Obtención del grafo de la ontología propuesta

OWL Viz: es un plugin para Protégé que permite visualizar con grafos los conceptos y las relaciones que tienen creadas, para adicionar este plugin se debe tener instalado el GraphViz. En la figura (ver figura 15) se muestra la gráfica de cómo quedaría la ontología para pruebas de aceptación con usuarios finales

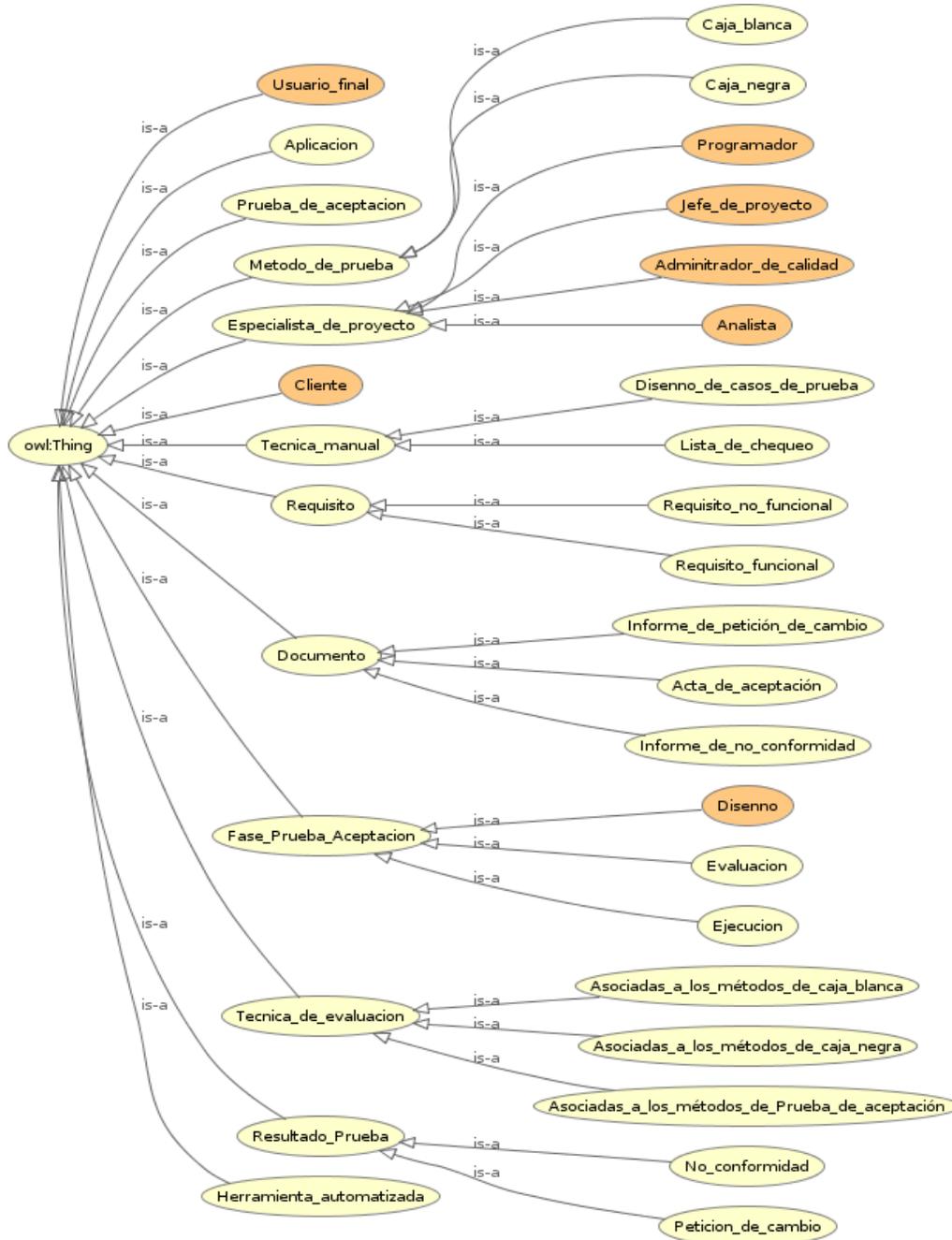


Figura 15 Pestaña OWL Viz del Protégé. Fuente: Elaboración propia.

2.2.4 Implementación

Esta actividad consistió en codificar el modelo generado en un lenguaje ontológico; para esto se seleccionó la herramienta de edición Protégé, versión 5.5, a través de la cual, al definir dicha ontología, se genera la codificación en el lenguaje ontológico OWL. Este lenguaje se construye sobre RDF. El Lenguaje de Ontologías Web (OWL) provee un lenguaje para definir ontologías. OWL trata de facilitar un lenguaje para ser usado con el fin de describir las clases, la relación entre las clases y las propiedades de las clases. 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 (Valdivieso, 2020).

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

Un ejemplo (ver figura 16) de cómo se hace la declaración de las clases, donde se declara la clase Prueba_De_Aceptación.

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

Figura 16 Declaración de las clases. Fuente: Elaboración propia.

La declaración de una subclase de otra clase, se realiza de la siguiente manera (ver figura 17), mostrando que la clase, Informe_de_petición_de_cambio es subclase de la clase Documento.

```
<SubClassOf>
  <Class IRI="#Informe_de_peticion_de_cambio"/>
  <Class IRI="#Documento"/>
</SubClassOf>
```

Figura 17 Declaración de una subclase de otra clase. Fuente: elaboración propia.

La declaración de las relaciones (ver figura 18) se realiza de la siguiente manera, donde se describe la relación que existe en la propuesta ontológica, realiza_prueba_de_aceptación.

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

Figura 18 Declaración de las relaciones. Fuente: Elaboración propia.

Se puede observar en este otro fragmento de código la creación de los individuos de las clases, en este caso la creación del individuo ha1 (ver figura 19), el cual es un individuo de la clase Herramienta Automatizada.

```
<Declaration>
  <NamedIndividual IRI="#ha1"/>
</Declaration>
<Declaration>
```

Figura 19 Creación de los individuos de una clase. Fuente: Elaboración propia.

La declaración de los axiomas se realiza de la siguiente manera, donde se describe la regla que existe en la ontología en la clase Analista (ver figura 20):

```
<EquivalentClasses>
  <Class IRI="#Analista"/>
  <ObjectIntersectionOf>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI="#utiliza"/>
      <Class IRI="#Herramienta_Automatizada"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI="#utiliza"/>
      <Class IRI="#Tecnica_Manual"/>
    </ObjectSomeValuesFrom>
    <ObjectSomeValuesFrom>
      <ObjectProperty IRI="#valida"/>
      <Class IRI="#Requisito"/>
    </ObjectSomeValuesFrom>
    <ObjectMinCardinality cardinality="1">
      <ObjectProperty IRI="#realiza"/>
      <Class IRI="#Disenno"/>
    </ObjectMinCardinality>
  </ObjectIntersectionOf>
</EquivalentClasses>
```

Figura 20 Declaración de los axiomas. Fuente: Elaboración propia.

SPARQL: Una de las ventajas que tiene el editor Protégé es que permite realizar ontologías con extensión OWL y RDF, debido a su entorno gráfico basado en plugin. Como ejemplo de estos plugin tenemos el lenguaje SPARQL. A continuación se muestra como se codificaron los datos para dar respuesta a una pregunta de competencia realizando una consulta SPARQL (ver figura 21).

Pregunta de competencia: ¿Qué documento gestiona el jefe1?

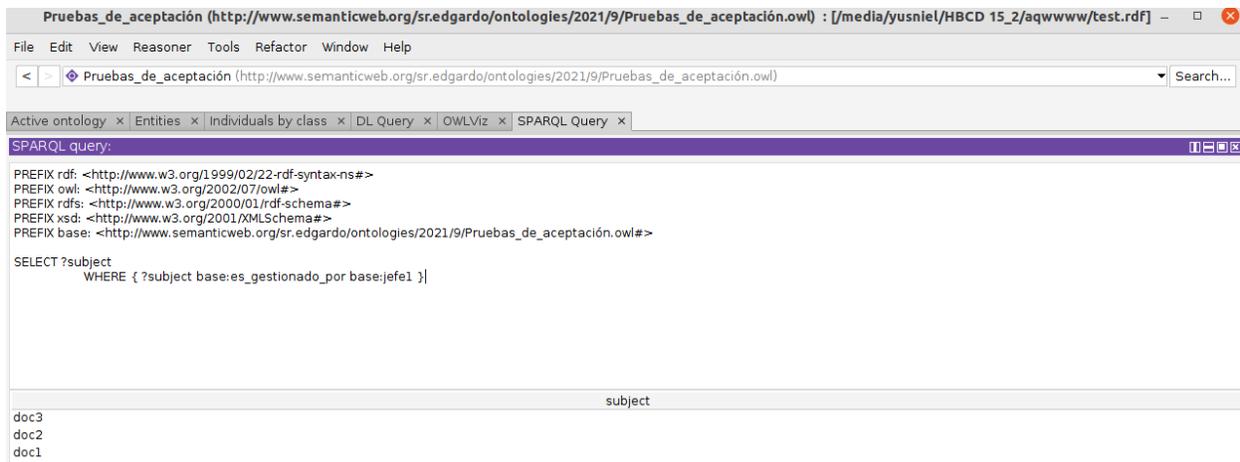


Figura 21 Consulta SPARQL. Fuente: Elaboración propia.

2.2.5 Instalación y Mantenimiento

Un programa portable o portátil es el que nos permite ejecutarse en cualquier ordenador sin necesidad de instalarlo. Estos programas vienen por defecto con todo lo necesario para poder ejecutarse en cualquier ordenador (librerías, configuraciones, datos de usuario, etc.). No necesitamos instalar nada sino que basta con tenerlo a mano y llevarlos encima, por ejemplo, en un USB para ejecutarlos en cualquier ordenador sin problemas. Además, como los datos se guardan dentro de la carpeta del programa, los podremos llevar siempre con nosotros y empezar a utilizarlo automáticamente (*Programas portables*, 2021).

Protege es un software de código abierto, el software de código abierto no necesita ser descifrado para su descarga en su sitio oficial. Sin embargo para su correcto funcionamiento como software portable, dado que Protege se basa en JAVA, se requiere al menos un JRE para instalar y configurar el entorno JAVA.

El mantenimiento del software es una de las fases del ciclo de vida de desarrollo de sistemas y no solo engloba la corrección de defectos. Un 80 % del esfuerzo de mantenimiento es invertido en mejoras y nuevas funcionalidades. El propósito del mantenimiento es preservar el valor del software sobre el tiempo alargando su vida útil y retrasando el costo económico que supondría una migración total hacia una nueva aplicación. Para ello, a lo largo de la vida útil de una aplicación, esta puede necesitar modificaciones por distintas razones (Martos, 2019):

- Para detectar problemas que puedan surgir en el futuro (mantenimiento preventivo).
- Corregir defectos encontrados y que originan un comportamiento distinto al esperado (mantenimiento correctivo).

- Cambiar el entorno de uso de la aplicación (mantenimiento adaptativo).
- Agregar nuevas funcionalidades o características (mantenimiento perfectivo).

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 del capítulo

Las conclusiones que se pueden extraer del capítulo son:

- La ontología para el proceso de pruebas de aceptación con usuarios finales en CIDI que se obtiene, permite representar y reutilizar el conocimiento existente y generarlo, lo que favorece la gestión del conocimiento en el proceso de pruebas de aceptación.
- La realización de las actividades de especificación, conceptualización y formalización de la metodología Methontology permitió lograr una ontología que cuenta con 34 clases, 40 relaciones, 55 atributos y 26 individuos.
- 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 3: VALIDACIÓN Y PRUEBA DE LA ONTOLOGÍA

En el presente capítulo se presentan los resultados de la validación de la ontología propuesta, dicho modelo posee cuatro fases, que tienen como objetivo comprobar el alcance y objetivo de la ontología desarrollada. La última fase muestra cómo la herramienta Protégé, validando la capacidad de inferencia de la ontología, responde a las preguntas de competencias identificadas. Para valorar la validez teórica y práctica de la ontología propuesta se utilizó el Método de validación consulta de expertos y el Método de evaluación triangular.

3.1. Validación de la ontología

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 (Guerrero-Proenza, 2012).

Según Ramos y otros (Ramos et al., 2009) el modelo para validar ontologías de dominio consta de cuatro fases. Resulta importante aplicarlas a lo largo del ciclo de vida. Ellas son:

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

Fase 1: Uso correcto del lenguaje

Para la evaluación del lenguaje y su correcto uso, es necesario que este sea sólido, siendo capaz para que cualquier expresión se derive del conocimiento codificado y completo de tal manera que la expresión que esté lógicamente implícita en la base de conocimiento pueda ser derivada. Se comprobó que este cumpla con los estándares para el desarrollo ontológico. En esta investigación se utilizó el lenguaje OWL el cual cumple con dichos estándares. Este es un lenguaje consistente y completo, permitiendo la aplicación de métodos de razonamiento sobre la ontología de manera satisfactoria.

Es importante resaltar que la escritura está libre de inconsistencias sintácticas y de errores. 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 pruebas de aceptación, logrando obtener una ortografía de óptima calidad.

Fase 2: Exactitud de la estructura taxonómica

En el Método de evaluación triangular se consideran tanto aspectos internos como aspectos relacionados con la retroalimentación del usuario. En el mismo se integran buenas prácticas de otros métodos existentes y de la Ingeniería de Software. Constituye una ampliación a la actividad de validación del Método para la integración de ontologías en un sistema para la evaluación de créditos (Rodríguez et al., 2018).

Para llevar a cabo la validación de esta fase se empleará el Método de evaluación triangular que propone utilizar casos de prueba, razonadores y una lista de chequeo con los errores más comunes detectados en el diseño de las ontologías propuesto por Rodríguez y otros (2018) pues en este se consideran tanto aspectos internos como aspectos relacionados con la retroalimentación del usuario.

Verificar el diseño

De vital importancia resuelta la evaluación del diseño de la ontología desde etapas tempranas de su elaboración y puede ser actualizada con nuevos errores que sean detectados, permitiendo así su evolución.

Al aplicar en 2 iteraciones la lista de chequeo propuesta por el Método de evaluación triangular (ver tabla 7), se detectaron errores de tipo II, III, V, VII en la primera iteración y errores de tipo VI, IX, X en la segunda iteración. Aunque la ontología propuesta en la presente investigación carece de características que conllevan a los errores XI y XII, todos los errores fueron resueltos y se evaluó como correcto el diseño de la misma.

Tabla 7 Lista de chequeo de errores comunes en el diseño de ontologías.

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

Fuente:(Rodríguez et al., 2018)

Fase 3. Validez del vocabulario

Para validar el vocabulario se chequea que los términos codificados en la ontología existan y sean significativos en otras fuentes de conocimiento independientes, como por ejemplo, el conocimiento contenido en el corpus². del dominio (Valdivieso, 2020).

Las actividades realizadas en esta fase fueron las siguientes:

- Análisis del corpus del dominio identificando los términos significativos a partir de los documentos.
- se evaluó el vocabulario considerando medidas de calidad de resultados usadas en escenarios de recuperación de información tales como la precisión y la exhaustividad.

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

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

Dónde:

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

En este caso la cantidad de términos que se solapan entre la ontología y el corpus es 34, al igual que la cantidad total de términos de la ontología.

$$\text{Precisión} = \frac{34}{34} = 1$$

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:

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

- $CCorp$: Cantidad total de términos del corpus, el cual su valor es 34.

$$\text{Exhaustividad} = \frac{34}{34} = 1$$

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

² Conjunto más extenso y ordenado posible de datos o textos científicos, literarios, que pueden servir de base a una investigación.

Fase 4. Adecuación a requisitos

Verificar las preguntas de competencias permite evaluar el uso de la ontología a partir de los objetivos que persigue. En este caso se aplican casos de prueba con la estructura: pregunta de competencia, escenario de prueba, resultado esperado y resultado obtenido.

En todos los casos de prueba se alcanzaron resultados satisfactorios. A continuación se muestra el caso de prueba 1.

Pregunta de competencia: ¿Qué especialista de proyecto gestiona el doc1, doc2 y doc3?

Escenario: En la herramienta Protege se crearon las instancias de la ontología (ver tabla 8).

Tabla 8 Creación de instancias de los casos de prueba.

Clases	Instancias	Propiedades	Valor de las propiedades
Especialista_De_Proyecto	jefe1	gestiona	doc1 doc2 doc3
Usuario	usuario1	registra	doc2 doc3
Especialista_De_Proyecto	analista1	realiza	diseño1
Usuario_Final	usuario1	revisa	aplicación1
Cliente	cliente1	desarrolla	ejecución1
Especialista_De_Proyecto	analista1 administrador_de_calidad1 programador1 jefe1	utiliza	ha1
Usuario_Final	usuario1	emite	nc1
Especialista_De_Proyecto	analista1	valida	requisitonf1

Fuente: Elaboración propia

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen el jefe1 gestiona doc1, doc2 y doc3 (ver figura 22).

Resultado obtenido: Satisfactorio.



Figura 22 Caso de prueba 1. Fuente: Elaboración propia.

Por otra parte, fueron realizados los otros 7 casos de pruebas (ver anexo 6) para comprobar la ontología del proceso de pruebas de aceptación con usuarios finales, evidenciando el cumplimiento de los objetivos iniciales de esta.

Durante el desarrollo de la investigación, la MSc. Alionuska Velázquez Cintra y la Ing. Yordanka Fuentes Castillo se mantuvieron 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.

Consulta a expertos

Para valorar la validez teórica de la ontología propuesta a partir de la caracterización del proceso de pruebas de aceptación se realizó una encuesta a 5 expertos.

La selección de los expertos se hizo a partir de analizar la síntesis curricular de cada uno a partir de la ficha de experto (Anexo 5). Los criterios de selección fueron:

- Experiencia en el desarrollo de pruebas de aceptación (mayor peso).
- Trabajo como miembro de equipo de desarrollo.
- Trayectoria científica en temáticas afines a las pruebas de aceptación o el desarrollo de ontologías.

Los resultados indican lo siguiente:

- El 100 por ciento de los encuestados plantean la utilidad de la propuesta y hacen referencia a su uso en etapas de formación del personal o para el entendimiento del proceso de pruebas de aceptación.
- Todos los expertos coinciden en que la propuesta se ajusta al escenario de la UCI y del Centro en particular.
- Coincidieron en calificar como adecuadas las relaciones que se establecen entre los elementos que intervienen en las pruebas de aceptación.
- Todos consideran que la propuesta permite lograr una comunicación efectiva entre clientes, usuario final y el equipo de desarrollo durante las pruebas de aceptación.

Los expertos recomendaron la identificación de otras reglas que permitieran conocer otros elementos de interés a lo largo del proceso de pruebas de aceptación.

3.2. Conclusiones del capítulo

Al finalizar el capítulo se arribaron a las siguientes conclusiones:

- La ontología se validó en función de la recuperación de la información y organización del proceso de pruebas de aceptación con usuarios finales en CIDI, demostrando que la estructura taxonómica no contiene inconsistencias ni redundancias y el manejo correcto del lenguaje.
- Se realizó el proceso de validación de forma iterativa, a través del razonador se chequearon las propiedades lógico-formales de manera práctica y sencilla. 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 GENERALES

El presente Trabajo de Diploma concluye con el desarrollo de una Ontología para pruebas de aceptación con usuarios finales, para lo cual:

- El análisis y la fundamentación teórica de los principales conceptos asociados a la investigación, permitió comprender el alcance de la misma.
- Para el desarrollo de la ontología se empleó la metodología Methontology, OWL como lenguaje para la representación y Protégé como herramienta para la creación, lo que permitió que la ontología pueda ser usada en varias plataformas, y reutilizarla en futuras investigaciones.
- A través de revisiones documentales, entrevistas y encuestas a jefes de proyectos y equipo de desarrollo en CIDI se constató la necesidad de la creación de una ontología capaz de recoger todo lo referente a esta área del conocimiento y se logró unificar el vocabulario empleado para que sea óptima la comunicación entre las partes.
- La metodología Methontology resultó de gran utilidad para la construcción de la ontología. La misma permitió analizar, diseñar, estructurar e implementar el conocimiento adquirido mediante técnicas, tablas y diagramas que pueden ser entendidos por los expertos del dominio.
- El método de evaluación triangular y la valoración de los criterios emitida por los expertos permitieron validar la propuesta, demostrando que la ontología desarrollada se elaboró correctamente y que se cumplieron los requisitos para los cuales fue creada.

RECOMENDACIONES

- Aplicar la ontología desarrollada de apoyo a los procesos de pruebas de aceptación y formación del personal en el CIDI.
- Seguir incorporando conceptos y relaciones de manera tal que siga creciendo la ontología.
- Extender el uso de la ontología a todos los centros de desarrollo de software de la universidad.

REFERENCIAS BIBLIOGRÁFICAS

Aguilar Gordón, F. del R., Chamba Zarango, A. P., Aguilar Gordón, F. del R., & Chamba Zarango, A. P. (2019). Reflexiones sobre la filosofía de la tecnología en los procesos educativos. *Conrado*, 15(70), 109-119.

Andersson, M., Kusetogullari, A., & Wernberg, J. (2021). Software development and innovation: Exploring the software shift in innovation in Swedish firms. *Technological Forecasting and Social Change*, 167, 120695. <https://doi.org/10.1016/j.techfore.2021.120695>

Ayora, M. J. M. (2021). *Diseño e implementación de una ontología para la gestión de la interoperabilidad en datos de producto entre sistemas ERP y PLM*. <https://idus.us.es/bitstream/handle/11441/109236/M1978%20Garrocho%20L%c3%b3pez%2c%20Francisco.pdf?sequence=1&isAllowed=y>

Bastidas, A. Q. (2007). *DEFINICION DE UNA ONTOLOGIA PARA LA GUIA DE CONOCIMIENTO SWEBOK*.

Bozyigit, F. (2021). *Linking software requirements and conceptual models: A systematic literature review*. <https://pdf.sciencedirectassets.com>

Bravo, M. C. (2018). *INGENIERÍA ONTOLÓGICA APLICADA EN EL DISEÑO DE UN SISTEMA DE ONTOLOGÍAS PARA LA GESTIÓN DE HORARIOS*. *Pistas Educativas*, No. 128.

Cárdenas Quintero, B. G. (2021). *Modelo basado en ingeniería ontológica para la formalización semántica de la información en el dominio de la gerencia de proyectos*. <https://repository.ean.edu.co/handle/10882/10365>

Carrión, L. (2018). *Ontología de apoyo al proceso de evaluación de productos de software en la UCI*.

Chandrasekaran, B. (1999). "What are ontologies, and why do we need them? (Vol. 14). in *IEEE Intelligent Systems and their Applications*.

CIDI. (2021). <https://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-innovacion-y-desarrollo-para-internet>

Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46(1), 41-64. [https://doi.org/10.1016/S0169-023X\(02\)00195-7](https://doi.org/10.1016/S0169-023X(02)00195-7)

Díaz, L. (2021). *Vista de Proceso de pruebas de software para un modelo de calidad en Cuba*. <https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/2914/3617>

Echeverría, D. (2011). *Desarrollo de una ontología de apoyo al procedimiento del Departamento de Pruebas de Software*. https://repositorio.uci.cu/bitstream/ident/7999/3/TM_04836_11.pdf

Fernández, A. (2015). *Modelo Ontológico de recuperación de información para la toma de decisiones en Gestión de Proyectos*.

Fernández, J. T. (2003). *Un Entorno de Integración de Ontologías para el Desarrollo de Sistemas de Gestión de Conocimiento*. <https://tdx.cat/bitstream/handle/10803/10921/FernandezBreis.pdf?sequence=1&isAllowed=y>

Galli, M. R. (2014). *Construcción basada en ontologías del Léxico Extendido del Lenguaje*. XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIO)-XV Simposio Argentino de Ingeniería de Software (Buenos Aires, 2014). <http://sedici.unlp.edu.ar/handle/10915/41751>

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199-220. <https://doi.org/10.1006/knac.1993.1008>

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, 43(5-6), 625-640. <https://doi.org/10.1006/ijhc.1995.1066>

Guerrero-Proenza, R. S. (2012). Metodología para el diseño y desarrollo de ontologías en el campo de la educación (Methodology for the Design and Development of Ontologies in the Field of Education). *Revista Cubana de Ciencias Informáticas*, 6.

Heflin, J. (2001). *SHOE: A Prototype Language for the Semantic Web*. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.458.4234&rep=rep1&type=pdf>

Hitzler, P. (2009). *OWL 2 Web Ontology Language Primer*.

- Horrocks. (2001). *The Ontology Inference Layer OIL*. https://d1wqtxts1xzle7.cloudfront.net/30706203/10.1.1.22.2713.pdf?1361987441=&response-content-disposition=inline%3B+filename%3DThe_ontology_inference_layer_OIL.pdf
- IEEE. (1998). *IEEE Standard for Software Verification and Validation*. IEEE Std. 1012-1998. <https://people.eecs.ku.edu/~hossein/Teaching/Stds/1012.pdf>
- López, I. (2016). *PROCESO DE VALIDACIÓN PARA PROYECTOS DE DESARROLLO DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*.
- Marin, A., Trujillo Casañola, Y., Buedo Hidalgo, D., Marin Diaz, A., Trujillo Casañola, Y., & Buedo Hidalgo, D. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. *Revista Cubana de Ciencias Informáticas*, 14(3), 83-104.
- Martínez. (2018). *Ontología de apoyo a las pruebas de software en la UCI*.
- Martos, S. (2019). *Martos—Visualización y análisis de métricas de desarrollo y mantenimiento en productos software*.
- McGuinness, D. L. (2004). *OWL Web Ontology Language O...* <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. *Computer Science Review*, 38, 100308. <https://doi.org/10.1016/j.cosrev.2020.100308>
- Mizoguchi, R. (1995). *Task_ontology_for_reuse_of_problem_solving_knowled.pdf*. https://www.researchgate.net/publication/243763523_Task_ontology_for_reuse_of_problem_solving_knowledge/link/00b4953b34da1787e0000000/download
- Pardo, J. M. (2013). *Proceso de Pruebas de Aceptación de Software*.
- Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Practico*. 810.
- Programas portables*. (2021). <https://www.softzone.es/programas/utilidades/programas-portables-imprescindibles/>
- Protégé*. (2021). <https://protege.stanford.edu/products.php>
- Ramos, E., Núñez, H., & Casañas, R. (2009). Esquema para evaluar ontologías únicas para un dominio de conocimiento. *Enlace*, 6(1), 57-71.

Rodríguez, Y. A., Hidalgo Delgado, Y., & Silega Martínez, N. (2018). *Un método práctico para la evaluación de ontologías*.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N. R., Van de Velde, W., & Wielinga, B. J. (2001). *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press. <https://doi.org/10.7551/mitpress/4073.001.0001>

Silva Manriquez, J. E. (2021, junio). *Creación de contenedores Docker sobre construcción de APIs REST a partir de ontologías y grafos de conocimientos* [Info:eu-repo/semantics/bachelorThesis]. E.T.S. de Ingenieros Informáticos (UPM). <https://oa.upm.es/68543/>

Socarrás Ramírez, I. de la C. (2018). *Proceso de Mejora de Procesos Organizacionales para Proyectos de Desarrollo de Software*. Universidad de las Ciencias Informáticas. <https://repositorio.uci.cu/bitstream/123456789/7876/1/Tesis%20Final%20IsmaraySISl.pdf>

Stadnicki, A. (2020). Towards a Modern Ontology Development Environment. *Procedia Computer Science*, 176, 753-762. <https://doi.org/10.1016/j.procs.2020.09.070>

UML. (2021). <http://www.sparxsystems.com.ar/>

Valdivieso, A. (2020). *Ontología para el proceso de pruebas de liberación de la Universidad de las Ciencias Informáticas*.

Valerio, Y. (2021, febrero 12). Ingeniero de pruebas UAT | Perfiles profesionales en TI. *Freelancer Blog*. <https://www.freelancermap.com/blog/es/que-hace-un-ingeniero-de-pruebas-de-aceptacion-uat/>

Velázquez, Y. (2020). *Herramienta para el diseño y gestión de casos de pruebas funcionales*.

ANEXOS

Anexo 1. Guía de entrevista para obtener información sobre el comportamiento de las pruebas de aceptación con la participación del cliente, con énfasis en el componente comunicacional.

Con esta entrevista se pretende evaluar sobre la necesidad de crear una ontología para pruebas de aceptación con usuarios finales.

Nombre:

Especialidad:

Grado científico:

¿Qué cargo ocupa?

¿Qué tiempo lleva ejerciendo esa responsabilidad?

¿En cuántos procesos de pruebas de aceptación ha participado?

Menos de 5

Entre 5 y 10

Más de 10

Las pruebas de aceptación

1. ¿Considera importante el desarrollo de pruebas de aceptación con la participación de los usuarios finales? ¿Por qué?
2. ¿Cree usted que conseguir una adecuada comunicación entre el usuario final, el cliente y el equipo de trabajo, contribuye a elevar los resultados obtenidos durante las pruebas de aceptación? ¿Por qué?
3. ¿Cuál de las partes involucradas, según su criterio, se ve más afectada con una mala comunicación?
4. ¿Considera que este elemento impacta en que las pruebas de aceptación exceden el tiempo planificado?
5. ¿Puede señalar algunos términos que impacten de manera negativa en el entendimiento entre los involucrados durante el proceso de aceptación?

- 6. ¿Qué alternativas utilizaría para mejorar la comunicación entre los involucrados en las pruebas de aceptación?
- 7. ¿Podiera lograrse una unificación del vocabulario en las pruebas de aceptación? ¿De qué manera?

Anexo 2. Encuesta para diagnóstico del proceso de pruebas de aceptación en CIDI.

El siguiente cuestionario pretende recopilar información sobre el desarrollo de las pruebas de aceptación con usuarios finales en CIDI.

Rol: _____

Grado científico: _____

Años de experiencia vinculada (a) a la actividad productiva: ____

¿En cuántos procesos de pruebas de aceptación ha participado?

____ menos de 5 _____ entre 5 y 10 _____ más de 10

1. En una escala ascendente del 1 al 10, marcando con un (x), señale la importancia que le otorga a la comunicación mediante un lenguaje común durante las pruebas de aceptación con la participación del usuario final:

1	2	3	4	5	6	7	8	9	10

2. En qué momentos del desarrollo de software se realizan las actividades de pruebas de aceptación, marque con una X:

__Al finalizar el desarrollo de cualquier artefacto

__Al finalizar la implementación del producto final

__Luego de revisado cualquier artefacto

__Después que el producto final es liberado

- Al inicio del desarrollo
- Durante todo el desarrollo
- Cada vez que se implementa un módulo
- Cada vez que se libera un módulo
- Otros

En caso de marcar Otros especifique: _____

3. Marque con una X qué tipo de productos, obtenidos del proceso de desarrollo de software, son sometidos a un proceso de pruebas de aceptación en su centro:

- Documentación
- Aplicación

4. Marque con una X los involucrados que participan en el proceso de pruebas de aceptación en su centro:

- Cliente
- Especialistas de otros proyectos
- Expertos Funcionales
- Especialistas de otros centros
- Usuarios finales
- Asesor de calidad
- Otros

En caso de marcar Otros especifique: _____

5. Señale con una X las actividades que sean realizadas como parte de las pruebas de aceptación en su centro.

- Planear las pruebas de aceptación

- Realizar reunión de inicio
- Preparar el entorno de validación
- Diseñar prueba
- Ejecutar validación
- Solucionar No Conformidades
- Responder Solicitudes de Cambio Conciliar No Conformidades
- Conciliar respuesta a las Solicitudes de Cambio
- Realizar Reunión de Cierre
- Análisis de Tendencias de las aceptaciones ejecutadas
- Otras

En caso de marcar Otros especifique: _____

6. Identifique los términos que, según su experiencia, impactan de manera negativa en el entendimiento entre los involucrados durante el proceso de aceptación

- No conformidad
- Pedidos de cambio
- Casos de pruebas
- Roles y responsabilidades

Otros _____

En caso de marcar Otros especifique: _____

7. En una escala ascendente del 1 al 10, marcando con un (x), señale el impacto de una adecuada comunicación durante las pruebas de aceptación en la calidad del producto final:

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

--	--	--	--	--	--	--	--	--	--

8. ¿Qué opinión usted tiene sobre la importancia de lograr un lenguaje común en función de mejorar el entendimiento entre los involucrados durante las pruebas de aceptación?

Anexo 3. Principios para el diseño de ontologías.

Para diseñar una Ontología, es necesario considerar algunas de las características deseables que éstas deberían exhibir. Los principios de diseño a considerar son los siguientes:

- ✓ Claridad y Objetividad: Definir los conceptos en forma clara y objetiva utilizando lenguaje natural para evitar ambigüedades.
- ✓ Coherencia: Garantizar que todas las inferencias derivadas sean consistentes con los axiomas.
- ✓ Completitud: Los conceptos deben ser expresados en términos necesarios y suficientes.
- ✓ Estandarización: Siempre que sea posible, los nombres asignados a los términos deberán seguir un estándar, definiendo y respetando reglas para la formación de los mismos.
- ✓ Máxima extensibilidad monótona: Deberá ser posible incluir en la Ontología especializaciones o generalizaciones, sin requerir una revisión de las definiciones existentes.
- ✓ Principio de distinción ontológica: Las clases de la Ontología con diferente criterio de identidad, deberán ser disjuntas.
- ✓ Diversificación de las jerarquías: Para que la Ontología se vea favorecida con los mecanismos de herencia múltiple, es conveniente usar tantos criterios de clasificación como sea posible.
- ✓ Minimización de la distancia semántica: Conceptos similares deberán ser agrupados y representados utilizando las mismas primitivas.
- ✓ Mínimo compromiso ontológico: Una Ontología debería imponer las menores exigencias posibles sobre el dominio que modela, es decir, se deben construir sólo los axiomas necesarios para representar el mundo a ser modelado.
- ✓ Modularidad: Al especificar una Ontología se hacen definiciones de diferentes elementos como clases, relaciones y axiomas; tales definiciones se pueden agrupar en

teorías que reúnen los objetos de una Ontología más relacionados entre sí. Se puede lograr una organización altamente modular con máxima cohesión en cada módulo y mínima interacción, considerando que cada teoría es un módulo en la organización de la Ontología. La modularidad permite flexibilidad y posibilidad de rehusar algunos módulos de la Ontología.

- ✓ Mínima dependencia con respecto a la codificación: Una Ontología debería permitir que los agentes que compartan los conocimientos, puedan ser implementados en diferentes sistemas y estilos de representación. Un diseño ontológico ideal debería cumplir con todos estos criterios, pero no siempre es posible.

Anexo 4. Datos recopilados en la encuesta para diagnóstico del proceso de pruebas de aceptación en CIDI.

	1	2	3	4	5	6	7	8	9	
Años de experiencia	13	13	7	6	12	7	7	7	6	%
¿Cuántos procesos de pruebas de aceptación ha participado?										
menos de 5	1								1	22%
entre 5 y 10			1			1		1		33%
más de 10		1		1	1		1			44%
Pregunta 1										
1										0%
2										0%
3										0%
4										0%
5										0%
6										0%
7										0%
8										0%
9										0%
10	1	1	1	1	1	1	1	1	1	100%
Pregunta 2										
Al finalizar el desarrollo de cualquier artefacto			1		1					22%
Al finalizar la implementación del producto final										0%
Luego de revisado cualquier artefacto										0%
Después que el producto final es liberado	1	1	1	1	1	1	1	1	1	100%
Al inicio del desarrollo										0%
Durante todo el desarrollo	1									11%
Cada vez que se implementa un módulo										0%
Cada vez que se libera un módulo					1		1			22%
Otros		1								11%
Pregunta 3										

Documentación	1	1	1			1	1	1	1		78%
Aplicación	1	1	1	1		1	1	1	1	1	100%
Pregunta 4											
Cliente	1	1		1		1	1	1	1	1	89%
Especialistas de otros proyectos			1								11%
Expertos Funcionales	1					1				1	33%
Especialistas de otros centros											0%
Usuarios finales	1			1		1	1	1	1	1	78%
Asesor de calidad			1								11%
Otros	1	1					1	1		1	56%
Pregunta 5											
Planear las pruebas de aceptación	1			1		1		1		1	56%
Realizar reunión de inicio	1					1		1		1	44%
Preparar el entorno de validación	1					1	1	1	1		56%
Diseñar prueba	1							1	1		33%
Ejecutar validación	1	1		1		1	1	1			67%
Solucionar No Conformidades	1	1	1	1					1	1	67%
Responder Solicitudes de Cambio Conciliar No Conformidades	1	1		1		1		1	1	1	78%
Conciliar respuesta a las Solicitudes de Cambio	1	1							1	1	44%
Realizar Reunión de Cierre	1	1	1	1		1		1		1	78%
Análisis de Tendencias de las aceptaciones ejecutadas	1	1				1					33%
Otras											0%
Pregunta 6											
No conformidad			1		1				1	1	44%
Pedidos de cambio			1	1	1			1	1	1	78%
Casos de pruebas			1			1					22%
Roles y responsabilidades											0%
Otros	1										11%
Pregunta 7											
	1										0%
	2										0%
	3										0%
	4										0%
	5										0%
	6										0%
	7										0%
	8										0%
	9										0%
	10	1	1	1	1	1	1	1	1	1	100%

Anexo 5. Ficha de experto para resumen curricular y Características de los expertos.

1. Nombre y Apellidos:

2. Datos de contacto:

3. Teléfono:

4. Email:

5. Datos laborales:

Organización a la que pertenece:

Cargo actual:

6. Años de experiencia en la industria de software:

7. Años de experiencia en actividades relacionadas con el proceso de pruebas de aceptación:

8. Seleccione de las siguientes funciones, cuál o cuáles desempeña o ha desempeñado:

Administrador de calidad _

Analista líder de proyectos de desarrollo de software _

Analista de pruebas de aceptación _

Jefe de proyectos de desarrollo de software _

Probador _

Programador_

9. Mencione las experiencias que posee en el desarrollo de ontologías y los resultados científicos relacionados con estas temáticas.

10. Mencione los conocimientos que posee en pruebas de aceptación y los resultados científicos relacionados con estas temáticas.

Tabla 9 Características de los expertos.

No. Experto	Años (experiencia)	Categoría científica	Cargo actual	Procesos de Pruebas de aceptación	
				Cantidad	Roles

1.	12	MSc.	Directora de calidad	más de 10	coordinadora
2.	15	Dr.C	Pdta UIC	más de 10	coordinadora y cliente
3.	9	MSc.	Administrador calidad proyectos	más de 10	coordinadora, cliente, y equipo de desarrollo
4.	6	MSc.	Metodóloga	entre 5 y 10	coordinadora
5.	5	ninguna	Líder de desarrollo CIDI	más de 10	equipo de desarrollo
Promedio	9.4				

Fuente: Elaboración Propia.

Anexo 6. Casos de prueba de la ontología Proceso de Pruebas de Aceptación.

Caso de prueba: 2

Pregunta de competencia: ¿Qué usuario registra el doc2, doc3?

Escenario: En la herramienta Protege se crearon las instancias de la ontología (ver tabla 8).

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen el usuario1 registra doc2 y doc3 (ver figura 23).

Resultado obtenido: Satisfactorio.

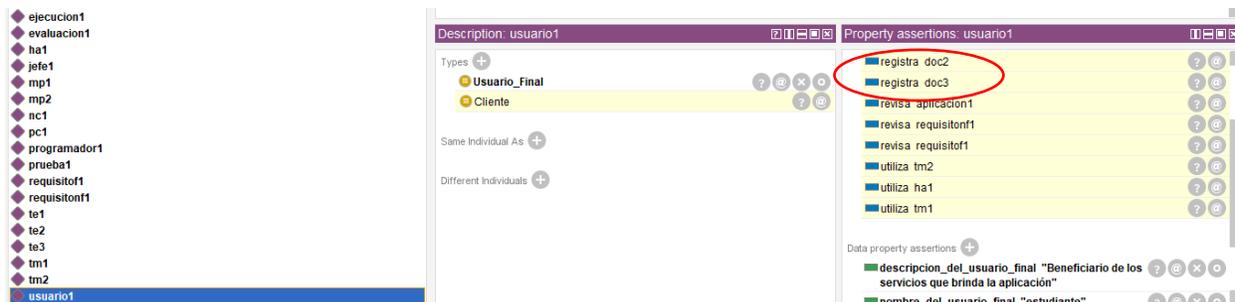


Figura 23 Caso de prueba 2. Fuente: Elaboración propia.

Caso de prueba: 3

Pregunta de competencia: ¿Qué especialista de proyecto realiza el diseño1?

Escenario: En la herramienta Protege se crearon las instancias de la ontología (ver tabla 8).

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen el analista1 realiza diseño1 (ver figura 24).

Resultado obtenido: Satisfactorio.



Figura 24 Caso de prueba 3. Fuente: Elaboración propia.

Caso de prueba: 4

Pregunta de competencia: ¿Qué usuario revisa la aplicación1?

Escenario: En la herramienta Protege se crearon las instancias de la ontología (ver tabla 8).

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen el usuario1 revisa la aplicación1 (ver figura 25).

Resultado obtenido: Satisfactorio.



Figura 25 Caso de prueba 4. Fuente: Elaboración propia.

Caso de prueba: 5

Pregunta de competencia: ¿Qué especialista de proyecto valida los requisitos no funcionales1?

Escenario: En la herramienta Protege se crearon las instancias de la ontología (ver tabla 8).

Resultado esperado: Al aplicar un razonador como se muestra en la siguiente imagen el analista1 valida el requisitonf1 (ver figura 26).

Resultado obtenido: Satisfactorio.



Figura 26 Caso de prueba 5. Fuente: Elaboración propia.