



UNIVERSIDAD DEL VALLE
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

**SISTEMA DE SELECCIÓN SEMÁNTICA DE CONOCIMIENTO
PERSONAL APLICABLE EN EL DESARROLLO DE PROYECTOS DE
SOFTWARE
CASO: INGENIERÍA DE SISTEMAS INFORMÁTICOS - UNIVERSIDAD
DEL VALLE**

**PROYECTO DE GRADO PARA OPTAR
AL TÍTULO DE LICENCIATURA EN
INGENIERÍA DE SISTEMAS
INFORMÁTICOS**

POSTULANTE: JOHN GUSTAVO CHOQUE CONDORI
TUTOR: ING. ALEJANDRO ARIAS URÍA

La Paz - Bolivia

2018

DEDICATORIA

A toda mi familia.

AGRADECIMIENTOS

A mis padres, hermana, cuñado y sobrina.

A Miro, Sascha y todo MD Systems.

A mis amigos en Japón.

A Aru, amigos de Univalle y Claudia.

RESUMEN

El presente Proyecto de Grado se realizó en la Universidad Privada del Valle, durante la segunda gestión académica del año 2017, considerando Proyectos de Grado aprobados, de estudiantes de la Carrera de Ingeniería de Sistemas Informáticos, desde la fundación de la carrera en la Universidad.

El propósito del presente proyecto, fue el desarrollo de un sistema para la selección semántica de conocimiento personal que sea aplicable en el desarrollo de proyectos de software.

A fin de permitir el desarrollo de un prototipo, se hizo uso de tecnologías semánticas, bases de datos orientadas a grafos y ontologías. Lo que significó el desarrollo de una herramienta de soporte para la gestión de conocimiento personal que permita mantener una ventaja competitiva en el área de estudios de los usuarios.

El desarrollo del sistema se sustenta en la teoría de la gestión de conocimiento organizacional y personal y el trabajo se enfoca metodológicamente desde la perspectiva descriptiva, en referencia a la gestión de conocimiento personal llevada a cabo por los estudiantes de la Carrera de Ingeniería de Sistemas en el desarrollo de Proyectos de software.

El procesamiento de datos permitió medir la eficacia de la selección de conocimiento personal de los Proyectos de Grado con nota de aprobación. Lo que permitió llegar a las siguientes conclusiones: 1) la determinación satisfactoria de las variables para la gestión de conocimiento personal, mediante el análisis del modelo de Völkel, 2) el diseño del modelo ontológico de la herramienta, mediante el uso de tecnologías semánticas y ontologías, 3) el modelamiento de un método de búsqueda semántica para la selección y extracción de conocimiento y 4) la eficacia de la selección de conocimiento en los Proyectos de Grado que formaron parte de la muestra, obtuvieron un valor que supera la nota mínima de aprobación requerida.

ABSTRACT

The present Degree Project was carried out at del Valle University, during the second academic year of 2017, considering approved Degree Projects of students of the Computer Systems Engineering Career, since the foundation of the degree at the University.

The purpose of the present project was to develop a system for the semantic selection of personal knowledge that is applicable in the development of software projects.

In order to allow the development of a prototype, semantic technologies, graph and ontology oriented databases were used. This meant the development of a support tool for the management of personal knowledge to maintain a competitive advantage in the area of the user studies.

The development of the system is based on the theory of the management of organizational and personal knowledge and the work is approached methodologically from a descriptive perspective, in reference to the management of personal knowledge carried out by students of the Systems Engineering Career in the development of software projects.

The data processing allowed to measure the effectiveness of the personal knowledge selection of the Grade Projects with approval note. This led to the following conclusions: 1) the satisfactory determination of variables for personal knowledge management by analysing the Völkel model, 2) The design of the ontological model of the tool, using semantic technologies and ontologies, 3) The modeling of a semantic search method for the selection and extraction of knowledge and 4) The effectiveness of knowledge selection in the Grade Projects that were part of the sample, obtained a value that exceeds the minimum score of approval required.

ÍNDICE DE CONTENIDO

CAPÍTULO I

MARCO INTRODUCTORIO

| | |
|---|---|
| 1.1. INTRODUCCIÓN..... | 1 |
| 1.2. PLANTEAMIENTO DEL PROBLEMA..... | 2 |
| 1.2.1. IDENTIFICACIÓN DEL PROBLEMA..... | 2 |
| 1.2.2. FORMULACIÓN DEL PROBLEMA..... | 5 |
| 1.3. JUSTIFICACIÓN..... | 5 |
| 1.3.1. JUSTIFICACIÓN TEÓRICA..... | 5 |
| 1.3.2. JUSTIFICACIÓN TÉCNICA..... | 6 |
| 1.3.3. JUSTIFICACIÓN SOCIAL..... | 6 |
| 1.4. OBJETIVOS..... | 7 |
| 1.4.1. OBJETIVO GENERAL..... | 7 |
| 1.4.2. OBJETIVOS ESPECÍFICOS..... | 7 |
| 1.5. ALCANCES Y LÍMITES..... | 7 |
| 1.5.1. ALCANCES..... | 7 |
| 1.5.2. LÍMITES..... | 7 |
| 1.6. METODOLOGÍA..... | 8 |
| 1.6.1. TIPO DE ESTUDIO..... | 8 |
| 1.6.2. MÉTODO DE INVESTIGACIÓN..... | 8 |
| 1.6.3. TÉCNICAS..... | 9 |
| 1.6.4. MUESTRA..... | 9 |

CAPÍTULO II

MARCO TEÓRICO

| | |
|---|----|
| 2.1. INGENIERÍA DE SISTEMAS..... | 10 |
| 2.2. INGENIERÍA DE SOFTWARE..... | 11 |
| 2.2.1. METODOLOGÍA ICONIX..... | 12 |
| 2.2.2. UML..... | 13 |
| 2.2.3. ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE..... | 14 |
| 2.2.3.1. Behat..... | 15 |
| 2.3. HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO..... | 15 |
| 2.3.1. DRUPAL..... | 15 |

| | |
|--|----|
| 2.3.1.1. <i>Distribuciones de Drupal</i> | 16 |
| 2.3.2. <i>PHP</i> | 16 |
| 2.3.3. <i>MYSQL</i> | 17 |
| 2.4. <i>CONOCIMIENTO</i> | 17 |
| 2.4.1. <i>CONOCIMIENTO TÁCITO Y CONOCIMIENTO EXPLÍCITO</i> | 19 |
| 2.4.2. <i>TRANSFORMACIÓN DE CONOCIMIENTO</i> | 19 |
| 2.5. <i>GESTIÓN DE CONOCIMIENTO</i> | 21 |
| 2.5.1. <i>GESTIÓN DE CONOCIMIENTO PERSONAL</i> | 22 |
| 2.5.1.1. <i>Modelo de Völkel</i> | 23 |
| 2.6. <i>WEB SEMÁNTICA</i> | 25 |
| 2.6.1. <i>ONTOLOGÍAS</i> | 27 |
| 2.6.1.1. <i>Tipos de ontologías</i> | 28 |
| 2.6.1.2. <i>Ontologías facetadas ligeras</i> | 28 |
| 2.6.1.3. <i>Patrones de diseño ontológicos</i> | 29 |
| 2.6.1.4. <i>Metodología NeOn</i> | 30 |
| 2.6.2. <i>ETIQUETAS SEMÁNTICAS</i> | 32 |
| 2.6.2.1. <i>Servicio web Open Calais</i> | 33 |
| 2.6.3. <i>QUERYGEN</i> | 34 |
| 2.7. <i>BASES DE DATOS ORIENTADAS A GRAFOS</i> | 35 |
| 2.7.1. <i>NEO4J</i> | 36 |
| 2.7.1.1. <i>Grafo de propiedades</i> | 37 |
| 2.8. <i>COMPETENCIAS</i> | 38 |
| 2.9. <i>MEDICIÓN DE EFICACIA DE LEWANDOWSKI</i> | 39 |
| 2.9.1. <i>VALOR-F</i> | 40 |
| 2.10. <i>UNIVERSIDAD DEL VALLE</i> | 40 |
| 2.10.1. <i>MISIÓN</i> | 41 |
| 2.10.2. <i>VISIÓN</i> | 41 |

CAPÍTULO III

INGENIERÍA DEL PROYECTO

| | |
|--|----|
| 3.1. <i>DEFINICIÓN DEL PROCESO DE GESTIÓN DE CONOCIMIENTO PERSONAL</i> | 42 |
| 3.1.1. <i>MODELO DE VÖLKEL</i> | 42 |
| 3.1.1.1. <i>Adquisición de conocimiento</i> | 42 |
| 3.1.1.2. <i>Codificación de conocimiento</i> | 42 |
| 3.1.1.3. <i>Extensión de conocimiento</i> | 43 |

| | |
|--|-----|
| 3.1.1.4. Recuperación de conocimiento..... | 44 |
| 3.1.1.5. Uso de conocimiento..... | 44 |
| 3.1.2. APLICACIÓN DEL MODELO DE VÖLKEL..... | 44 |
| 3.1.3. INDEXACIÓN DE SEÑALES DE CONOCIMIENTO..... | 45 |
| 3.1.4. COMPETENCIAS..... | 47 |
| 3.2. DISEÑO DEL MODELO ONTOLÓGICO..... | 49 |
| 3.2.1. ARQUITECTURA SEMÁNTICA..... | 50 |
| 3.2.1.1. Servicio web Open Calais..... | 50 |
| 3.2.1.2. Neo4j..... | 53 |
| 3.2.1.3. Cypher..... | 55 |
| 3.2.1.4. Ontologías facetadas ligeras..... | 55 |
| 3.2.1.5. Lógica unificadora..... | 58 |
| 3.2.1.6. Pruebas y Encriptación..... | 60 |
| 3.2.1.7. Confianza..... | 61 |
| 3.2.2. PATRÓN DE DISEÑO ONTOLÓGICO..... | 62 |
| 3.3. MÉTODO DE BÚSQUEDA SEMÁNTICA..... | 63 |
| 3.3.1. CONSTRUCCIÓN DE CONSULTAS SEMÁNTICAS..... | 64 |
| 3.3.1.1. Definición de significado de palabras clave..... | 64 |
| 3.3.1.2. Generación de consultas semánticas..... | 66 |
| 3.3.1.3. Acceso a repositorios de información..... | 69 |
| 3.3.2. IMPLEMENTACIÓN DE INTERFACES..... | 71 |
| 3.3.2.1. Fase de requerimientos..... | 71 |
| 3.3.2.2. Fase de análisis y diseño preliminar..... | 75 |
| 3.3.2.3. Fase de diseño detallado..... | 81 |
| 3.3.2.4. Fase de implementación..... | 86 |
| 3.4. EFICACIA DE LA RECUPERACIÓN DE CONOCIMIENTO PERSONAL..... | 93 |
| 3.4.1. DESARROLLO DE ONTOLOGÍAS DE PRUEBA..... | 94 |
| 3.4.1.1. Fase de Inicio..... | 94 |
| 3.4.1.2. Fase de diseño..... | 95 |
| 3.4.1.3. Fase de implementación..... | 97 |
| 3.4.2. MEDICIÓN DE LA EFICACIA DE LA RECUPERACIÓN DE CONOCIMIENTO..... | 98 |
| 3.4.2.1. Selección de queries..... | 101 |
| 3.4.2.2. Obtención de resultados..... | 117 |
| 3.4.2.3. Presentación de resultados..... | 117 |

| | |
|--|-----|
| 3.4.2.4. <i>Valoración de resultados</i> | 117 |
| 3.4.2.5. <i>Análisis de datos</i> | 121 |
| CONCLUSIONES..... | 125 |
| RECOMENDACIONES..... | 127 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 128 |

ÍNDICE DE TABLAS

| | |
|--|-----|
| Tabla N° 3.1. Matriz de confusión para PG-01..... | 117 |
| Tabla N° 3.2. Matriz de confusión para PG-02..... | 118 |
| Tabla N° 3.3. Matriz de confusión para PG-03..... | 118 |
| Tabla N° 3.4. Matriz de confusión para PG-04..... | 118 |
| Tabla N° 3.5. Matriz de confusión para PG-05..... | 118 |
| Tabla N° 3.6. Matriz de confusión para PG-06..... | 119 |
| Tabla N° 3.7. Matriz de confusión para PG-07..... | 119 |
| Tabla N° 3.8. Matriz de confusión para PG-08..... | 119 |
| Tabla N° 3.9. Matriz de confusión para PG-09..... | 119 |
| Tabla N° 3.10. Matriz de confusión para PG-10..... | 120 |
| Tabla N° 3.11. Matriz de confusión para PG-11..... | 120 |
| Tabla N° 3.12. Matriz de confusión para PG-12..... | 120 |
| Tabla N° 3.13. Matriz de confusión para PG-13..... | 120 |
| Tabla N° 3.14. Matriz de confusión para PG-14..... | 121 |
| Tabla N° 3.15. Matriz de confusión para PG-15..... | 121 |
| Tabla N° 3.16. Resultados de precisión y sensibilidad..... | 122 |
| Tabla N° 3.17. Valor-F para cada Proyecto de grado..... | 123 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura N° 2.1. Diferencia entre datos, información, conocimiento, sabiduría..... | 19 |
| Figura N° 2.2. Modelo SECI..... | 20 |
| Figura N° 2.3. Ciclo de vida de Señales de conocimiento isoladamente..... | 25 |
| Figura N° 2.4. Arquitectura de la web semántica..... | 26 |
| Figura N° 2.5. Proceso de QueryGen..... | 35 |
| Figura N° 2.6. El modelo de grafo de propiedades..... | 38 |
| Figura N° 3.1. Indexación de Señales de conocimiento..... | 47 |
| Figura N° 3.2. Arquitectura semántica resultante..... | 50 |
| Figura N° 3.3. Ejemplo de grafo Neo4j con datos de Drupal..... | 53 |
| Figura N° 3.4. Ejemplo de facetas de entidades..... | 56 |
| Figura N° 3.5. Ejemplo de facetas de temas..... | 56 |
| Figura N° 3.6. Ejemplo de facetas de términos industriales..... | 57 |
| Figura N° 3.7. Ejemplo de facetas de etiquetas sociales..... | 57 |
| Figura N° 3.8. Diagrama de componentes de la capa lógica unificadora..... | 59 |
| Figura N° 3.9. Proceso de encriptación..... | 61 |
| Figura N° 3.10. Patrón de diseño ontológico para Ontologías facetadas ligeras..... | 63 |
| Figura N° 3.11. Método de construcción de consultas..... | 64 |
| Figura N° 3.12. Función de definición de significado..... | 65 |
| Figura N° 3.13. Interfaz de selección de conocimiento..... | 70 |
| Figura N° 3.14. Ejemplo de resultados de consultas..... | 71 |
| Figura N° 3.15. Modelo de dominio..... | 74 |
| Figura N° 3.16. Diagrama de casos de uso del sistema..... | 74 |
| Figura N° 3.17. Diagrama de componentes del sistema..... | 75 |
| Figura N° 3.18. Diagrama de secuencia: Crear Nota..... | 82 |
| Figura N° 3.19. Diagrama de secuencia: Extender contenido..... | 83 |
| Figura N° 3.20. Diagrama de secuencia: Asignar términos taxonómicos..... | 84 |
| Figura N° 3.21. Diagrama de secuencia: Agregar relaciones..... | 85 |
| Figura N° 3.22. Diagrama de secuencia: Recuperar conocimiento..... | 86 |
| Figura N° 3.23. Archivo install del perfil de instalación..... | 87 |
| Figura N° 3.24. Mapa de navegación de la herramienta..... | 88 |
| Figura N° 3.25. Estructura de bloques del sistema..... | 89 |

| | |
|--|----|
| Figura N° 3.26. Tipo de contenido: Nota..... | 90 |
| Figura N° 3.27. Diagrama de planificación de la ontología..... | 95 |
| Figura N° 3.28. Ejemplo de Clases, Individuos, Atributos y relaciones..... | 96 |
| Figura N° 3.29. Transformación de contenidos a ontologías facetadas ligeras..... | 97 |
| Figura N° 3.30. Vista general de una ontología de prueba..... | 98 |

ÍNDICE DE CUADROS

| | |
|---|-----|
| Cuadro N° 2.1. Tipos de distribuciones de Drupal..... | 16 |
| Cuadro N° 2.2. Beneficios de la gestión de conocimiento personal..... | 23 |
| Cuadro N° 2.3. Diferencias de Neo4j con Bases de datos relacionales..... | 37 |
| Cuadro N° 2.4. Etapas del marco de trabajo de Lewandowski..... | 39 |
| Cuadro N° 3.1. Variables de proceso del modelo de gestión de conocimiento de Völkel..... | 45 |
| Cuadro N° 3.2. Ejemplo de indexación de conocimiento..... | 46 |
| Cuadro N° 3.3. Selección de conocimientos de las competencias..... | 48 |
| Cuadro N° 3.4. Selección de capacidades de las competencias..... | 49 |
| Cuadro N° 3.5. Ejemplo de solicitud a OpenCalais..... | 52 |
| Cuadro N° 3.6. Tipos de nodos y características de almacenamiento..... | 54 |
| Cuadro N° 3.7. Atributos de confianza definidos por Open Calais..... | 62 |
| Cuadro N° 3.8. Selección de facetas asociadas a un término..... | 66 |
| Cuadro N° 3.9. Elementos de consultas generadas..... | 67 |
| Cuadro N° 3.10. Estructura de consultas generadas..... | 68 |
| Cuadro N° 3.11. Ejemplo de generación de consulta..... | 69 |
| Cuadro N° 3.12. Caso de uso: Crear Nota..... | 76 |
| Cuadro N° 3.13. Caso de uso: Extender contenido..... | 77 |
| Cuadro N° 3.14. Caso de uso: Asignar términos taxonómicos..... | 78 |
| Cuadro N° 3.15. Caso de uso: Agregar relaciones..... | 80 |
| Cuadro N° 3.16. Caso de uso: Recuperar conocimiento..... | 81 |
| Cuadro N° 3.17. Plan de pruebas..... | 91 |
| Cuadro N° 3.18. Prueba de aceptación: Crear nota..... | 91 |
| Cuadro N° 3.19. Prueba de aceptación: Extender contenido..... | 92 |
| Cuadro N° 3.20. Prueba de aceptación: Asignar términos taxonómicos..... | 92 |
| Cuadro N° 3.21. Prueba de aceptación: Agregar relaciones..... | 93 |
| Cuadro N° 3.22. Prueba de aceptación: Recuperar conocimiento..... | 93 |
| Cuadro N° 3.23. Listado de Proyectos de Grado..... | 100 |
| Cuadro N° 3.24. Definición de query para PG-01..... | 102 |
| Cuadro N° 3.25. Definición de query para PG-02..... | 103 |
| Cuadro N° 3.26. Definición de query para PG-03..... | 104 |

| | |
|---|-----|
| Cuadro N° 3.27. Definición de query para PG-04..... | 105 |
| Cuadro N° 3.28. Definición de query para PG-05..... | 106 |
| Cuadro N° 3.29. Definición de query para PG-06..... | 107 |
| Cuadro N° 3.30. Definición de query para PG-07..... | 108 |
| Cuadro N° 3.31. Definición de query para PG-08..... | 109 |
| Cuadro N° 3.32. Definición de query para PG-09..... | 110 |
| Cuadro N° 3.33. Definición de query para PG-10..... | 111 |
| Cuadro N° 3.34. Definición de query para PG-11..... | 112 |
| Cuadro N° 3.35. Definición de query para PG-12..... | 113 |
| Cuadro N° 3.36. Definición de query para PG-13..... | 114 |
| Cuadro N° 3.37. Definición de query para PG-14..... | 115 |
| Cuadro N° 3.38. Definición de query para PG-15..... | 116 |

CAPÍTULO I
MARCO INTRODUCTORIO

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

El aprendizaje es parte del desarrollo del ser humano, está presente a lo largo de la vida y es un factor determinante para un futuro desenvolvimiento en la sociedad. La educación tradicional hacía énfasis en la memorización de contenidos y actualmente incluye habilidades como la resolución de problemas a fin de impulsar la transferencia de conocimiento.

Con la llegada del siglo XXI, se ha marcado una revolución tecnológica que ha permitido la digitalización de la información. Sin embargo, tal digitalización ha causado que las fuentes de información se multipliquen al punto que los usuarios son incapaces de recuperar información relevante de manera oportuna.

El conocimiento se ha convertido en un activo que las empresas valoran, marcando un factor diferenciador y competitivo en el mercado. Aquellos individuos que cuenten con habilidades para la gestión de conocimiento podrán mantener relevancia en su ámbito laboral. En consecuencia, si los estudiantes son capaces de gestionar su conocimiento y tienen herramientas a su disposición, podrán crear y formar parte de empresas que tomen en cuenta la importancia de la gestión de conocimiento individual. Usarán todo el conocimiento obtenido para resolver problemas, transferirán el mismo y lo condicionarán para especificar los contextos en el que es útil.

La Universidad Privada del Valle, dentro de la Carrera de Ingeniería de Sistemas Informáticos, cuenta con materias destinadas al desarrollo de proyectos que sirven como herramienta práctica para la aplicación del conocimiento adquirido por los estudiantes. Sin embargo, los docentes encargados de dirigir tales materias han identificado diversas deficiencias en el desarrollo de dichos proyectos, que incluyen problemas con la aplicación de metodologías de investigación, redacción, organización, identificación de problemáticas, codificación, entre otros.

Mediante lo mencionado anteriormente, el presente proyecto propone desarrollar un sistema para la selección semántica de conocimiento personal que permita aplicarlo eficazmente y sirva de herramienta de soporte para los estudiantes de Ingeniería de Sistemas. De esa manera, los estudiantes pueden gestionar su conocimiento personal, recuperarlo y aplicarlo a fin de mantener relevancia en su ámbito laboral.

1.2. PLANTEAMIENTO DEL PROBLEMA

1.2.1. IDENTIFICACIÓN DEL PROBLEMA

En la actual economía basada en conocimiento, donde el mismo juega un papel central en la prosperidad económica, las organizaciones se enfrentan a enormes desafíos para mejorar la productividad y capacidad individual (Jain, 2011). La efectividad y crecimiento de las organizaciones ha llegado a depender del conocimiento, lo que ha llevado a entender que el mismo necesita ser administrado (Kupetiene, 2015). Con la gestión adecuada de conocimiento, las empresas y sus empleados podrán actuar inteligentemente sobre los requerimientos básicos para una competitividad sostenida, éxito y viabilidad (Wiig, 1997).

Fundamentalmente, las empresas “tratan de obtener el mejor valor comercial de sus activos existentes de conocimiento” (Wiig, 1997). No obstante, sólo pueden darse cuenta del valor de esos activos, cuando estos pueden ser transferidos efectivamente entre individuos (Gupta, Iyer & Aronson, 2000). De esa manera, los líderes de las organizaciones pueden identificar la ubicación del conocimiento y a quién, cómo y cuándo debe ser transferido.

Las organizaciones pueden obtener buenos resultados con la inclusión de la gestión sistemática de conocimiento como parte de sus prácticas corporativas (Wiig, 1997). Asimismo, se considera que las competencias de los trabajadores de conocimiento son una parte importante del capital intelectual (Ulrich, 1998) y que la gestión de conocimiento afecta a la mayoría de las funciones de la empresa (Wiig, 1997). Sin embargo, un gran porcentaje de trabajadores de conocimiento usan una gran cantidad de su tiempo en la búsqueda de información requerida para completar una tarea (Jefferson, 2016) y son incapaces de encontrar sus propios documentos aproximadamente la mitad del tiempo, causando la reconstrucción de los mismos (Kirsch, 2006).

El conocimiento es parte central de las tareas diarias de los trabajadores de conocimiento y es una fuente de ventaja competitiva no sólo a nivel organizacional, sino también a nivel individual (Razmerita, Kirchner & Sudzina, 2009). Cheong (2011) afirma que el aprendizaje individual está relacionado con el aprendizaje organizacional debido a que todo nuevo conocimiento tiene su inicio en los individuos. De esa manera, “si los individuos se vuelven productivos en la gestión de conocimiento personal, la organización se vuelve productiva” (Jain, 2011).

Las empresas no serán capaces de realizar una gestión efectiva de conocimiento hasta que cada trabajador cuente con las habilidades y herramientas para la gestión de conocimiento personal (Tsui, 2002). De esa forma, se puede equipar a los trabajadores de conocimiento con las habilidades necesarias para la administración de su conocimiento individual a fin de mantener, desarrollar y promocionar sus habilidades que les permitan contar con una ventaja competitiva en su área de trabajo (Pauleen, 2009).

Las características que diferencian a los trabajadores de conocimiento expertos de los novatos¹, son (Pellegrino & Hilton, 2012):

- Los expertos pueden encontrar características y patrones de información que no son diferenciados por novatos.
- Los expertos han adquirido una gran cantidad de conocimiento que está organizado de tal manera que refleja el profundo entendimiento de un determinado tema.
- Los expertos reflejan contextos de aplicabilidad y condicionan su conocimiento.
- Los expertos son capaces de recuperar importantes aspectos de su conocimiento con poco esfuerzo.

Khan, McLeod y Hovy (2004) indican que “un problema clave es que los usuarios pueden verse fácilmente abrumados por la cantidad de información disponible a través de medios electrónicos”. El enorme número de documentos disponibles en la Web hace que encontrar los documentos relevantes sea una tarea difícil. Especialmente debido a la vaguedad de los lenguajes naturales, los conceptos abstractos y las relaciones semánticas. Las ontologías y los metadatos semánticos pueden aportar una solución a estos problemas. (Nagypál, 2005). Así, las ontologías en la gestión de conocimiento se utilizan para definir propiedades y relaciones entre los conceptos. (Belavkin, 2013).

Cuando las conexiones entre conceptos, que los estudiantes crean, forman estructuras de conocimiento organizadas de manera precisa y significativa, los estudiantes son capaces de recuperar y aplicar su conocimiento de manera efectiva y eficiente. Si el conocimiento previo de los estudiantes es robusto y preciso y se activa en el momento adecuado, proporciona una base sólida para construir nuevos conocimientos. (Ambrose, Bridges, DiPietro, Lovett, Normal, 2010).

¹ También referente a estudiantes.

En consecuencia, “la idea de ayudar a los estudiantes a organizar su conocimiento, sugiere que estos, pueden beneficiarse de los modelos de expertos que abordan la resolución de problemas” (Pellegrino & Hilton, 2012). Mientras que las organizaciones proveen oportunidades para el desarrollo y aprendizaje personal, la última responsabilidad se mantiene en los individuos (Pauleen & Gorman, 2011). De esa manera, la gestión de conocimiento personal provee un marco de trabajo para la administración e integración de nueva información, con el fin de enriquecer cada base de conocimiento de manera efectiva. Y así, permitir que cada individuo aplique su propio conocimiento personal en la resolución de problemas (Cheong & Tsui, 2010).

Durante la segunda gestión académica del año 2016, los estudiantes de octavo semestre de la Carrera de Ingeniería de Sistemas Informáticos de la Universidad Privada del Valle, presentaron problemas en el desarrollo de sus proyectos para la materia de Taller de Programación III, enfocada en el desarrollo de Proyectos de Grado, lo que causó que ningún estudiante apruebe la materia.

Entre las falencias que se pudieron identificar, por medio de encuestas (cf. Apéndice 2) realizadas a los docentes de la Carrera de Ingeniería de Sistemas Informáticos, se encuentran:

- La deficiente definición de objetivos.
- El no cumplimiento de objetivos.
- Mala planificación.
- Mala organización.
- Falta de capacidad para relacionar los conceptos aprendidos y olvido de los mismos.

De igual forma, se evidenció que los estudiantes desarrollan proyectos deficientes debido al poco conocimiento sobre la aplicación de metodologías de desarrollo de software y falta de conocimiento sobre metodologías de investigación, esto, junto con la falta de habilidades de organización, estimación y planificación, influye negativamente en el desarrollo de los proyectos de software que los estudiantes realizan.

Tomando en cuenta que en el desarrollo de Proyectos de Grado se debe aplicar todo el conocimiento adquirido durante la Carrera, permitir que los estudiantes cuenten con una base de conocimiento generada a lo largo de sus estudios influiría, a futuro, en que los mismos sean capaces de aplicar su propio conocimiento y se mantengan relevantes dentro de su área de

estudios.

Las encuestas realizadas además, revelaron que los docentes apoyan la idea de desarrollar un sistema para la gestión de conocimiento personal. Los estudiantes podrán recolectar información de las materias cursadas, asociar conceptos aprendidos, recuperar el conocimiento y aplicarlo en el desarrollo de proyectos de software.

Por otra parte, el soporte de los sistemas informáticos en el proceso de aprendizaje conlleva enfoques dirigidos a instituciones, docentes y grupos estudiantiles. Tal es el caso de los “Sistemas de administración de aprendizaje” (LMS) los cuáles, se centran en permitir que los docentes administren la distribución de material, tareas, comunicación y otros aspectos instructivos de sus cursos. (Shawar & Al-Sadi, 2010). De manera similar, el objetivo de los “Sistemas de gestión de conocimiento” (KMS) es apoyar con la construcción, el compartir y la aplicación del conocimiento en las organizaciones” (Alavi, M., & Leidner, D., 1999). Así pues, si bien existen diferentes alternativas para soportar el proceso educativo, éstas se enfocan principalmente en la comunicación docente-alumno y la construcción de conocimiento grupal.

1.2.2. FORMULACIÓN DEL PROBLEMA

¿Cómo se puede brindar a los estudiantes de la Carrera de Ingeniería de Sistemas, una herramienta de apoyo en la resolución de problemas para proyectos de software?

1.3. JUSTIFICACIÓN

1.3.1. JUSTIFICACIÓN TEÓRICA

La creación de un sistema de gestión de conocimiento personal explora diferentes ámbitos de la Ingeniería de Sistemas Informáticos, entre los cuales se incluyen: la ingeniería de software, con la aplicación de una metodología de desarrollo en el proceso de desarrollo de la herramienta; la metodología de la investigación, con los diferentes enfoques para realizar la investigación, revisión bibliográfica y estudio; la programación, requerida para el desarrollo de la herramienta, módulos de la misma y la interacción de todos los elementos que la conforman; el aseguramiento de la calidad, referente a las pruebas que el sistema requiere para asegurar su correcto funcionamiento; la web semántica, para la clasificación de la información en el sistema y su

posterior selección; y las bases de datos orientadas a grafos, para el almacenamiento estructurado del conocimiento ingresado en el sistema.

1.3.2. JUSTIFICACIÓN TÉCNICA

El desarrollo de un sistema para la gestión y recuperación de conocimiento personal, requiere el entendimiento de la gestión de contenidos y las herramientas que la permiten. La creación de ontologías a partir de notas debe realizarse mediante la comunicación de Bases de datos orientadas a grafos, como es el caso de Neo4j, y un entorno *open source* multiplataforma, con el uso de módulos escritos en el lenguaje PHP para Drupal. La selección semántica de conocimiento personal debe hacerse mediante la creación de consultas en el lenguaje Cypher, nativo de la Base de datos Neo4j, y la asignación de etiquetas semánticas con el servicio Open Calais hace uso del *API* público que el servicio ofrece, incluyendo los métodos de asignación de etiquetas y análisis semántico. El aseguramiento de calidad se realiza mediante la implementación de pruebas con el *framework* Behat y el *API* ofrecido por el núcleo de Drupal. El manejo de dependencias requiere el conocimiento de la herramienta Composer a fin de ofrecer al usuario una instalación simplificada del sistema.

1.3.3. JUSTIFICACIÓN SOCIAL

Debido a que el conocimiento es un activo importante en la sociedad actual, permitir que los estudiantes puedan desarrollar habilidades para la gestión de conocimiento influirá en su capacidad para funcionar como miembros de la comunidad. Aquellos estudiantes que obtengan tales habilidades durante su desarrollo académico podrán generar estrategias para una eficiente realización de labores y podrán mantener una ventaja competitiva en su área de estudios.

El sistema propuesto, tiene como fin servir de apoyo a los estudiantes en la recuperación y aplicación de su conocimiento. La herramienta permite la administración de conocimiento duradero y promueve la vigencia de los estudiantes en su área académica.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un sistema para la selección y recuperación semántica de conocimiento personal que sea aplicable en el desarrollo de proyectos de software para los estudiantes de Ingeniería de Sistemas.

1.4.2. OBJETIVOS ESPECÍFICOS

El presente proyecto tiene los siguientes objetivos específicos:

- Determinar las variables necesarias para el proceso de gestión de conocimiento personal.
- Diseñar el modelo ontológico para la gestión de conocimiento personal.
- Modelar el método de búsqueda semántica para la selección y extracción de conocimiento.
- Medir la eficacia de la recuperación del conocimiento personal del sistema propuesto.

1.5. ALCANCES Y LÍMITES

1.5.1. ALCANCES

Los alcances del presente proyecto son:

- Se realizó el modelado de búsquedas semánticas para la selección de conocimientos asociados a etiquetas semánticas.
- Los datos utilizados en la pruebas se obtuvieron de una muestra de los Proyectos de Grado defendidos y aprobados de los estudiantes de la Carrera de Ingeniería de Sistemas Informáticos de la Universidad Privada del Valle, desde la fundación de la carrera.
- El prototipo se desarrolló orientado a plataforma web.
- El prototipo fue validado en la Universidad Privada del Valle sub sede La Paz.
- La selección de conocimiento fue modelada mediante palabras clave.

1.5.2. LÍMITES

Los límites del presente proyecto son:

- Se desarrolló únicamente un prototipo del sistema.

- Se utilizaron datos simulados para la retroalimentación del sistema.
- Los datos de prueba fueron generados a partir de la información disponible.
- No se utilizó lenguaje natural para la selección de conocimiento.
- Las actualizaciones de seguridad y corrección de errores del gestor de contenidos dependen del núcleo de Drupal.
- No se midió la influencia de la motivación de los estudiantes en el uso del sistema.

1.6. METODOLOGÍA

1.6.1. TIPO DE ESTUDIO

En el presente proyecto se realizó un estudio descriptivo, puesto que se analizó el efecto de la gestión de conocimiento personal en el trabajo realizado por personas que forman parte de empresas que toman al conocimiento como activo principal. Además, se pudo definir las falencias presentadas por los estudiantes en el desarrollo de proyectos de software. Así mismo, se pudo evidenciar y verificar la eficacia de la selección semántica de conocimiento mediante la implementación de una herramienta y el uso de Proyectos de Grado como información de muestra.

Por otra parte, se realizó un estudio documental, mediante la búsqueda de información referente a la gestión de conocimiento personal y el análisis de la misma en su influencia sobre el desempeño organizacional e individual. También, se analizó la información referente a la representación de conocimiento mediante el uso de ontologías y el uso de la web semántica para la selección de conocimiento.

1.6.2. MÉTODO DE INVESTIGACIÓN

El proyecto fue realizado con el método deductivo, a fin de inferir conclusiones en base a premisas planteadas. Para llegar a una deducción final, se realizaron verificaciones con datos de prueba. Mediante el sistema y las pruebas realizadas al mismo, se pudo verificar la eficacia de la aplicación de conocimiento en problemas de desarrollo de software de los estudiantes de Ingeniería de Sistemas Informáticos de la Universidad Privada del Valle.

1.6.3. TÉCNICAS

En el presente proyecto se utilizaron las siguientes técnicas:

- Encuesta: Se usó con el objetivo de identificar los problemas presentados por los estudiantes en la aplicación de conocimiento en proyectos de software, en base a la percepción de los docentes de la carrera de Ingeniería de Sistemas encargados de la revisión de los Proyectos de Grado. Su aplicación sirvió de soporte en la identificación del problema en el caso de estudio.
- Revisión documental: Permitió la recolección de métodos, técnicas y teoría referente a la gestión de conocimiento personal y la selección semántica de conocimiento, a fin de contar con una base teórica en el desarrollo de la herramienta. Además, permitió recolectar la información necesaria para la realización de pruebas.

1.6.4. MUESTRA

Resultado del muestreo simple aleatorio de una población de 19 Proyectos de Grado, que constituyen la cantidad de Proyectos defendidos y aprobados por estudiantes de la Carrera de Ingeniería de Sistemas Informáticos de la Universidad del Valle, sub sede La Paz, disponibles para su revisión en los registros de la Universidad.

La fórmula aplicada para la obtención de la muestra fue:

$$n = \frac{Z^2 pq N}{NE^2 + Z^2 pq}$$

Donde, considerando los valores:

$$Z = 1.645; p = 0.5; q = 0.5; N = 19; E = 0.1$$

$$n = 14.83 \Rightarrow 15$$

Se obtuvo una muestra de 15 Proyectos de Grado que fueron usados para la realización de pruebas con el software desarrollado en el proceso de selección de conocimiento personal. Se usaron los datos de los Proyectos a fin de crear ontologías de prueba.

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II

MARCO TEÓRICO

2.1. INGENIERÍA DE SISTEMAS

La Ingeniería de Sistemas se define como “la aplicación efectiva de métodos científicos y de ingeniería para transformar una necesidad operativa en una configuración determinada del sistema mediante un proceso vertical (*top-down*) iterativo de establecimiento de requisitos, selección del concepto, análisis y asignación funcional, síntesis, optimización del diseño, prueba y evaluación” (Blanchard, Fabrycky & Verma, 1994).

El Consejo Internacional de Ingeniería de Sistemas (INCOSE) define a la Ingeniería de Sistemas como “un enfoque interdisciplinario y un medio para permitir la realización de sistemas exitosos. Se centra en definir las necesidades del cliente y la funcionalidad requerida al principio del ciclo de desarrollo, documentar los requisitos y luego proceder con la síntesis del diseño y validación del sistema mientras se considera el problema completo.” (Walden, Roedler, Forsberg, Hamelin, & Shortell, 2015, p. 5).

Además, la Ingeniería de Sistemas “busca un diseño seguro y equilibrado ante intereses opuestos y múltiples limitaciones, a veces conflictivas. El ingeniero de sistemas debe desarrollar la habilidad e instinto para identificar y enfocar esfuerzos en evaluaciones para optimizar el diseño general y no favorecer un sistema/subsistema a expensas de otro.” (Kapurch, 2010).

Thayer (Thayer, 2002) lista cinco funciones que incluye la Ingeniería de Sistemas:

- Definición del problema: Determina las necesidades y límites mediante los requerimientos.
- Análisis de soluciones: Determina un conjunto de posibles soluciones para seleccionar la opción más óptima.
- Planificación del proceso: Determina las tareas a ser realizadas, el esfuerzo requerido para el desarrollo del producto y los riesgos probables.
- Control del proceso: Determina los métodos para el control del proyecto, midiendo el progreso del mismo.
- Evaluación del producto: Determina la calidad del producto entregado mediante pruebas, demostración, análisis, examinación e inspección.

2.2. INGENIERÍA DE SOFTWARE

La IEEE define a la Ingeniería de Software como “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software” (Radatz, Geraci & Katki, 1990).

Sommerville indica que la ingeniería de software es “un enfoque sistemático para la producción de software que considera los costos, calendarios y problemas de confiabilidad, como también las necesidades de los clientes y productores de software.” (2011, p. 10). Además, el autor señala que el enfoque sistemático de la ingeniería de software es llamado “proceso de software”, una secuencia de actividades que conducen al desarrollo de un producto de software. Según el autor, existen cuatro actividades fundamentales que son comunes en todos los proceso de software (2011, p. 12):

- Especificación de software, los clientes e ingenieros definen el software que será desarrollado y las restricciones del mismo.
- Desarrollo de software, el sistema es desarrollado y programado.
- Validación de software, se verifica que el software cumple los requerimientos del cliente.
- Evolución del software, el software es modificado para reflejar los cambios en los requerimientos del mercado y de los clientes.

Van Vliet presenta una lista de las características de la Ingeniería de Software (2007, p. 6):

- La Ingeniería de Software se ocupa del desarrollo de grandes programas, creando una diferenciación entre el desarrollo de pequeños y grandes proyectos de software.
- El tema principal implica el dominio de la complejidad, separando el problema principal en partes donde la complejidad sea manejable.
- El software evoluciona, tomando en cuenta los gastos a futuro en la actualización del proyecto.
- El software debe apoyar efectivamente a sus usuarios, con funcionalidades que se adecuen a las tareas del usuario.
- La ingeniería de software es una acción de equilibrio, que considera los cambios a los requerimientos en el desarrollo de un proyecto.

2.2.1. METODOLOGÍA ICONIX

ICONIX es una metodología de desarrollo de software que “está entre la complejidad de *Rational Unified Processes* (RUP) y la simplicidad y pragmatismo de *Extreme Programming* (XP), sin eliminar las tareas de análisis y de diseño que XP no contempla. ICONIX es un proceso simplificado en comparación con otros procesos más tradicionales, que unifican un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto.” (Amavizca, García, Jiménez, Duarte & Vásquez, 2014).

Rosenberg y Scott (2001, p. 21) denotan tres características que definen el enfoque de la metodología:

- Ofrece un uso simplificado de *UML*: centrándose en un subconjunto de los diferentes diagramas de *UML*.
- Ofrece un alto nivel de trazabilidad: la metodología impulsa la permanencia de interés en las necesidades del usuario.
- Es iterativo e incremental: se realizan múltiples iteraciones mientras se realiza el modelado del dominio y se identifican los casos de uso.

Rosenberg & Stephens (2007, p. 3) describen el proceso de la metodología:

Fase de requerimientos:

- Requerimientos funcionales: Se define lo que el sistema será capaz de hacer.
- Modelo de dominio: Se debe entender el espacio del problema en términos no ambiguos.
- Requerimientos de comportamiento: Se define cómo interactúan el usuario y el sistema. Se recomienda iniciar con el desarrollo de prototipos de interfaces y la identificación de los casos de uso que serán implementados.

Hito 1: Evaluación de requerimientos, verificar que los casos de uso concuerdan con las expectativas del cliente.

Luego, para cada grupo de casos de uso identificados, separados en iteraciones, se realiza lo siguiente.

Fase de análisis y diseño preliminar

- Análisis de robustez: Se desarrolla un diagrama de robustez.

- Se actualiza el modelo de dominio mientras se realiza el diagrama de robustez. Aquí se identifican las clases faltantes, se corrigen las ambigüedades y se añaden atributos a los objetos del dominio.
- Nombrar los controladores de software para permitir la implementación del caso de uso.
- Mejorar los casos de uso preliminares.

Hito 2: Revisión de los diseños preliminares.

Fase de diseño detallado

- Diagrama de secuencias: Realizar un diagrama de secuencias para mostrar en detalle cómo se implementará el caso de uso seleccionado.
- Actualizar el modelo de dominio y agregar operaciones a los objetos del dominio.

Hito 3: Revisión del diseño crítico.

Fase de implementación

- Pruebas de código y unitarias: Escribir el código y las pruebas unitarias.
- Pruebas de integración y escenario: Basar las pruebas de integración con los casos de uso.
- Realizar una revisión del código y actualización del modelo de dominio para continuar con la siguiente iteración del desarrollo.

2.2.2. UML

UML (Unified Modeling Language) es “un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria.” (Grau, & Segura, 2001).

Eriksson, Penker, Lyons & Fado (2003) señalan que UML “provee mecanismos estándar de la industria para visualizar, especificar, construir y documentar sistemas de software”. Indican también, que UML “busca definir conceptos, para establecer una notación para la comunicación de los mismos y fortalecer las reglas gramaticales relacionadas para construir modelos de software”. El objetivo es entonces, según los autores, “tener la flexibilidad aplicable a todo el desarrollo de software al proveer definiciones suficientemente coherentes para la automatización”.

Booch, Rumbaugh, Jacobson, Martínez, & Molina (1999) listan los diagramas principales

definidos por el lenguaje:

- Diagrama de clases: describe el diseño estructural del sistema mediante clases y sus relaciones.
- Diagrama de objetos: representa las instancias de los elementos presentes en el diagrama de clases, expresando la parte estática de una interacción.
- Diagrama de interacción: representa un conjunto de objetos y sus relaciones, además de los mensajes enviados entre ellos.
- Diagrama de secuencia: denota el orden temporal de los mensajes intercambiados en un escenario concreto.
- Diagrama de comunicación: describe la interacción de un objeto mediante su distribución en el diagrama.
- Diagrama de casos de uso: representa las acciones que cada tipo de usuario realiza.
- Diagrama de estados: señala los comportamientos de los sistemas dirigidos por eventos.
- Diagrama de actividades: representa el orden con el que las actividades son realizadas.

2.2.3. ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE

El aseguramiento de calidad de software es un proceso “para revisar los mecanismos internos de control y asegurar la adherencia de estándares de software y procedimientos. El objetivo del proceso es asegurar la conformidad a los requerimientos, reducir riesgos, medir controles internos y mejorar la calidad mientras se ajusta a los programas definidos y límites de presupuesto” (Owens & Khazanchi, 2009). Citando a Galin, los autores listan las tareas del aseguramiento de la calidad de software como:

- Asegurar la planificación del proyecto de software: Las prácticas de calidad deben ser planificadas con suficiente antelación para que haya tiempo para que se apliquen.
- Asegurar los requisitos del usuario: Los requisitos deben ser revisados desde el principio hasta el final para asegurar el cumplimiento de los estándares establecidos y la conformidad con las necesidades del usuario.
- Asegurar el proceso de diseño: Garantizar que se sigan las metodologías y que los requisitos son satisfechos por el diseño.

- Asegurar las prácticas de codificación: Primero deben establecerse las normas de codificación, las prácticas y las directrices y, a continuación, cumplirlas.
- Asegurar la integración y las pruebas de software: Integración de software y las pruebas deben ser planificadas, implementadas y ejecutadas.

2.2.3.1. Behat

Behat es un sistema de pruebas automatizado. Las pruebas de Behat están escritas en frases en inglés que se combinan en escenarios legibles para humanos. Behat es la estructura central utilizada para ejecutar pruebas. Es capaz de probar varios tipos de sistemas: comandos de terminal, *APIs REST*, etc. Para que Behat pueda probar las páginas web, es necesario añadir Mink y un emulador de navegador. Mink funciona como conector entre Behat y los emuladores de navegador, y proporciona una API de pruebas consistente (Lanier, 2014).

2.3. HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO

2.3.1. DRUPAL

Drupal es un sistema gestor de contenidos *open source* con un diseño modular que permite la extensión de su funcionalidad, mediante el uso de módulos y el cambio de su interfaz, mediante la instalación de temas. El núcleo de Drupal está escrito en PHP e incluye archivos escritos en JavaScript y CSS. Funciona bajo diferentes sistemas operativos, como: Linux, Windows y Mac OS. Puede ser implementado en un servidor web Apache, Nginx o IIS y es compatible con las bases de datos: PostgreSQL, SQLite, MySQL, MariaDB y Percona (Drupal.org, 2017).

Drupal 8 es la versión estable más reciente de Drupal, fue lanzada el 19 de noviembre de 2015 y es la primera versión del sistema en adoptar el framework Symfony. Entre las características más importantes se encuentran (Drupal.org, 2017):

- Integración con Twig para la creación de diseños seguros, rápidos flexibles.
- Reutilización de componentes con la integración de Symfony.
- Administración de dependencias con Composer.

2.3.1.1. Distribuciones de Drupal

Las distribuciones de Drupal son paquetes que incluyen el software base, módulos, temas, perfiles de instalación y configuración predefinida para la creación de sitios con un fin determinado. Las distribuciones permiten la instalación de sitios complejos reduciendo los pasos de instalación y configuración de cada elemento individualmente (Drupal.org, 2017).

Las principales diferencias entre los distintos tipos de distribuciones de Drupal, se muestran en el cuadro siguiente.

Cuadro N° 2.1.
Tipos de distribuciones de Drupal

| Tipo de distribución | Usuarios finales | Distribución | Módulos de contribución | Casos de uso |
|-------------------------------|---|---------------------|--------------------------------|----------------------------|
| Drupal core | Desarrolladores Creadores de sitios | Paquete | Ninguno | Genérico |
| Distribuciones completas | Creadores de sitios Usuarios finales | Paquete | Usualmente muchos | Especializado |
| Otras distribuciones | Desarrolladores Creadores de sitios | Paquete | Pocos o ninguno | Especializado |
| SaaS (Software como servicio) | Usuarios finales | Nube | Muchos | Genéricos o especializados |

Fuente: Drupal.org, 2017.

2.3.2. PHP

PHP es un lenguaje de programación *open source* que se centra en el desarrollo web y puede ser incrustado en contenido HTML. Se distingue de lenguajes de ejecución del lado del cliente, como JavaScript, debido a que se ejecuta en el lado del servidor. (Php.net, 2017).

El código PHP es usado en tres principales áreas (Php.net, 2017):

- Funcionamiento del lado del servidor: Requiere de tres elementos: un analizador PHP, un servidor web y un navegador web.
- Funcionamiento en línea de comandos: El código PHP puede ser ejecutado sin un servidor

y navegador web para tareas de procesamiento de información.

- Diseño de aplicaciones de escritorio: Mediante el uso de la librería PHP-Gtk para interfaces multiplataforma.

PHP funciona en la mayoría de Sistemas Operativos, como Linux, Unix, Windows y Mac OS. Puede ser ejecutado en servidores Apache e IIS, entre otros. Permite la programación orientada a objetos y programación procedural. Además, tiene soporte para la comunicación con servicios que usan protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP y COM (Php.net, 2017).

2.3.3. *MYSQL*

MySQL, es un sistema administrador de Bases de Datos *open source* que es desarrollado, distribuido y respaldado por Oracle. Entre sus características más importantes están (Dev.mysql.com, 2017):

- Base de datos relacional: Almacena la información en tablas separadas.
- Funciona en sistemas cliente/servidor y sistemas embebidos: Consiste de un servidor SQL de multihilo que soporte diferentes entornos *back-end*, diferentes librerías y herramientas administrativas.

Además, está escrito en C y C++, usa un diseño multicapa con módulos independientes, cuenta con motores transaccionales y no transaccionales, permite la conexión mediante sockets TCP/IP en cualquier plataforma y cuenta con soporte para la revisión y optimización de declaraciones SQL (Dev.mysql.com, 2017).

2.4. CONOCIMIENTO

Cheong (2011) afirma que “no existe una única definición consensuada del conocimiento” debido a que existen diversas definiciones para el mismo, entre las cuales, podemos nombrar:

- Van y Spijkervet (1997, p. 36): “Conjunto de percepciones, experiencias y procedimientos que se consideran correctos y verdaderos y que, por lo tanto, guían los pensamientos, el comportamiento y la comunicación de las personas”.
- Davenport y Prusak (2000, p. 5): “Mezcla de experiencia enmarcada, valores, información contextual y visión experta que proporciona un marco para evaluar e incorporar nuevas

experiencias e información. Se origina y se aplica en las mentes de los conocedores".

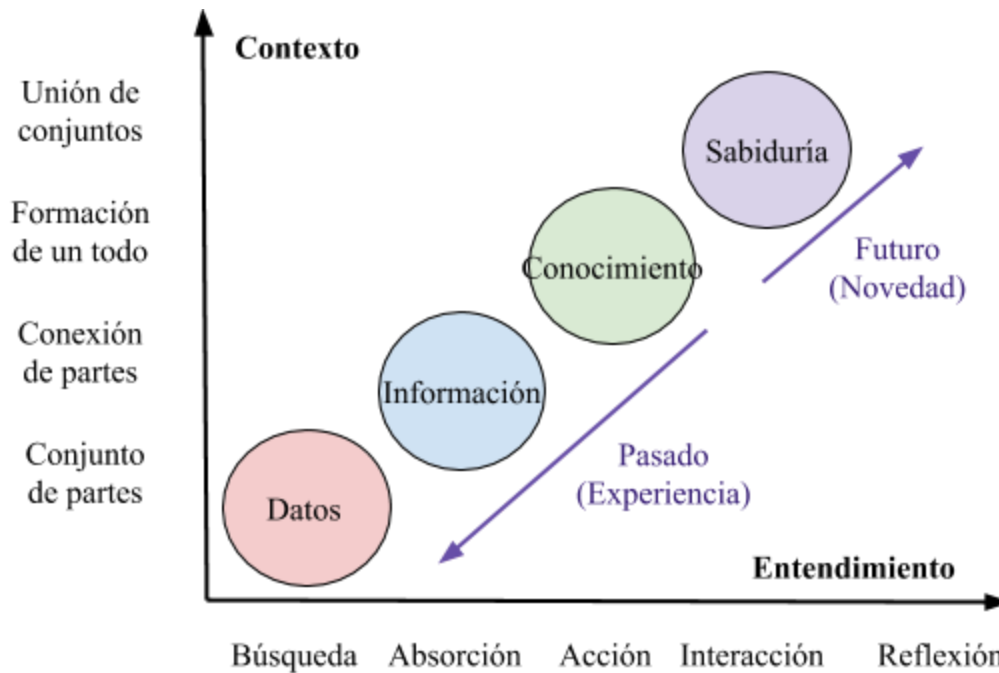
De esa manera, Ackoff (1989) define una clasificación para la mente humana que representa los diferentes niveles de entendimiento y se describe a continuación:

- Datos: Conjunto de hechos discretos y objetivos sobre eventos. Son productos de la observación y son símbolos que representan propiedades de objetos, eventos y su ambiente. No tiene significado por su cuenta debido a que representan observaciones o hechos fuera de contexto.
- Información: Puede considerarse como datos organizados sistemáticamente y es el resultado de colocarlos en un contexto significativo. Puede responder a preguntas que comienzan con quién, qué, dónde, cuándo y cuántos.
- Conocimiento: Es la colección apropiada de información a tal punto que es útil. Representa el *know-how* y es aquello que hace posible la transformación de información en instrucciones. Además, "el conocimiento que cada individuo posee, es producto de su experiencia y comprende las normas por las cuales la persona evalúa nuevos aportes de su ambiente. Es una mezcla de experiencia enmarcada, valores, información contextual, visión experta e intuición fundamentada que proporciona un entorno y un marco para evaluar e incorporar nuevas experiencias e información" (Kupetiene, 2015).
- Entendimiento: Es un sistema hombre-máquina para facilitar y acelerar el aprendizaje y adaptación. Es un proceso cognitivo y analítico.
- Sabiduría: Aquellos que añaden valor y requieren de la función mental del juicio en el que se discierne entre lo correcto e incorrecto o lo bueno y lo malo.

La siguiente figura muestra los niveles de entendimiento definidos por Ackoff.

Figura N° 2.1.

Diferencia entre datos, información, conocimiento, sabiduría.



Fuente: Ackoff, 1989.

2.4.1. CONOCIMIENTO TÁCITO Y CONOCIMIENTO EXPLÍCITO

Según Nonaka y Takeuchi (2004) existen dos tipos de conocimiento, que se diferencian de la siguiente manera:

- El conocimiento explícito “puede ser expresado en palabras, números o sonidos, y compartido en la forma de datos, fórmulas científicas, visuales, grabaciones de audio, especificaciones de productos o manuales. El conocimiento explícito puede transmitirse fácilmente a los individuos de manera formal y sistemática” (Nonaka & Takeuchi, 2008).
- El conocimiento tácito “no es visible ni expresable. El conocimiento tácito es altamente personal y difícil de formalizar... Está profundamente enlazado a las acciones y experiencias de un individuo, así como en los ideales, valores o emociones que abarca”.

2.4.2. TRANSFORMACIÓN DE CONOCIMIENTO

Nonaka y Takeuchi (1995) definen el modelo SECI (Socialización, externalización, combinación,

e internalización) para la transformación de conocimiento, el cual, describe cuatro maneras para la ampliación de los conocimiento tácito y explícito en términos de calidad y cantidad. El modelo en espiral, se muestra a continuación.

Figura N° 2.2.
Modelo SECI



Fuente: Nonaka & Takeuchi, 1995.

Nonaka (2008) explica los modos de conversión de conocimiento del modelo CESI.

- De conocimiento tácito a tácito: Es el compartimiento de conocimiento directamente con otra persona. A pesar de que el alumno puede aprender del maestro, ninguno obtiene conocimiento sistemático del área de conocimiento. Si el conocimiento no se transforma a ser Explícito, no puede ser nivelado.
- De conocimiento explícito a explícito: Una persona puede combinar piezas discretas de conocimiento, pero esto, no implica la extensión de la base de conocimiento existente.
- De conocimiento tácito a explícito: Cuando una persona es capaz de definir las bases de su conocimiento tácito, significa la conversión a un conocimiento explícito que puede ser

compartido.

- De conocimiento explícito a conocimiento tácito: Cuando el conocimiento explícito es compartido, los receptores extienden, amplían y redefinen su propio conocimiento tácito.

2.5. GESTIÓN DE CONOCIMIENTO

La gestión del conocimiento es “un proceso que se ocupa del desarrollo, almacenamiento, recuperación y difusión de información y conocimientos especializados dentro de una organización para apoyar y mejorar su rendimiento empresarial” (Gupta, Iyer & Aronson, 2000).

La gestión del conocimiento “engloba todos los métodos, instrumentos y herramientas que contribuyen a la promoción de un proceso integrado de conocimiento básico, con las cuatro actividades básicas siguientes como mínimo: generar conocimiento, almacenar conocimiento, distribuirlo y aplicarlo, en todas las áreas y niveles de la organización, con el fin de mejorar el rendimiento organizacional centrándose en la creación de valor del proceso empresarial” (Mertins, Heisig & Vorbeck, 2003, p. 11).

King (2009) afirma que el objetivo de la gestión de conocimiento es “el nivelamiento y mejora de los activos de conocimiento para efectuar mejores prácticas de conocimiento, mejorar las actitudes organizacionales, tomar mejores decisiones y mejorar el rendimiento organizacional”.

Wiig (1997) indica que la Gestión de Conocimiento se enfoca en ocho importantes áreas operativas:

- Inspección, desarrollo, mantenimiento y aseguramiento de los recursos intelectuales y de conocimiento de la empresa.
- Promover la creación de contenido e innovación para todos.
- Determinar el conocimiento y las habilidades requeridas para tener un desempeño efectivo, organizarlo, hacer disponible el conocimiento requerido, empaquetarlo (en cursos de entrenamiento, manuales de procedimientos o sistemas basados en conocimiento) y distribuirlo a los puntos de acción relevantes.
- Modificar y reestructurar la empresa para usar el conocimiento más eficientemente, aprovechar las oportunidades para explotar los activos de conocimiento, minimizar las brechas de conocimiento y cuellos de botella, y maximizar el contenido de valor añadido

de los productos y servicios.

- Crear, gobernar y monitorear actividades y estrategias futuras y de largo plazo basadas en el conocimiento, especialmente nuevas inversiones en conocimiento, alianzas estratégicas, adquisiciones, programas importantes de contratación, etc., en base a oportunidades, prioridades y necesidades identificadas.
- Salvaguardar el conocimiento propio y competitivo y controlar el uso del conocimiento para asegurarse de que sólo se utiliza el mejor conocimiento, que el conocimiento valioso no se atrofia y que el conocimiento no se da a los competidores.
- Proveer capacidades de la Gestión de Conocimiento y un arquitectura de conocimiento para que las instalaciones, procedimientos, guías, directrices, estándares y prácticas de la empresa, faciliten y apoyen activamente la Gestión de Conocimiento como para de las prácticas y cultura de la organización.
- Medir el rendimiento de todos los activos de conocimiento como activos capitalizados para ser construidos, explotados, renovados y administrados como parte del cumplimiento de la misión y los objetivos de la organización.

2.5.1. GESTIÓN DE CONOCIMIENTO PERSONAL

La gestión de conocimiento personal es “un marco conceptual para organizar e integrar la información que nosotros, como individuos, consideramos importante para que se convierta en parte de nuestra base de conocimiento personal” (Frاند y Hixon, 1999). Además, es saber “qué conocimiento tenemos y cómo lo podemos organizar, movilizar y usar a fin de lograr nuestras metas y cómo podemos continuar creando conocimiento” (Sutton, 2008).

Jain (2011) afirma que el principal motor de la gestión de conocimiento personal “es aprovechar una base de datos de conocimientos personales y, a través de ella, el conocimiento organizacional y la productividad. Se ha demostrado empíricamente que existe una estrecha relación entre el conocimiento personal y la gestión del conocimiento organizacional”.

Por su parte, Jefferson (2006) indica que la gestión de conocimiento personal "se centra en un enfoque ascendente, con una perspectiva individual de gestión de conocimiento. El objetivo es permitir que los individuos elijan qué información recoger, cómo estructurarla y con quién

compartirla. Los individuos necesitan ser capaces de manejar su propia información para que sea significativa, accesible cuando sea necesario, pueda ser fácilmente explotada. La gestión de conocimiento personal permite a los trabajadores organizar tanto el contenido digital como el papel de tal manera que les permite dar sentido a la avalancha a la que están continuamente expuestos".

Jain (2011) también, lista los beneficios organizacionales e individuales de la gestión de conocimiento personal.

Cuadro N° 2.2.
Beneficios de la gestión de conocimiento personal

| Nivel individual | Nivel organizacional |
|--|---------------------------------------|
| Enfrenta el problema de la sobrecarga de la información. | Mejor productividad y rendimiento. |
| Los individuos reconocen su propio valor y consecuentemente tomar mejores decisiones para el desarrollo propio. | Innovación continua. |
| Los individuos están mejores equipados para su trabajo y son más productivos. | Toma de decisiones efectiva. |
| Permite la toma de decisiones informada y basada en el conocimiento. | Diseminación interna de conocimiento. |
| Los empleados se sienten motivados si se les dan herramientas y métodos de Gestión de Conocimiento Personal para hacer sus vidas más fáciles y más empleables. | Conciencia de la información externa. |
| Identifica las brechas de habilidades y de conocimiento personal, se basa en las capacidades. | |

Fuente: Jain, 2011.

2.5.1.1. Modelo de Völkel

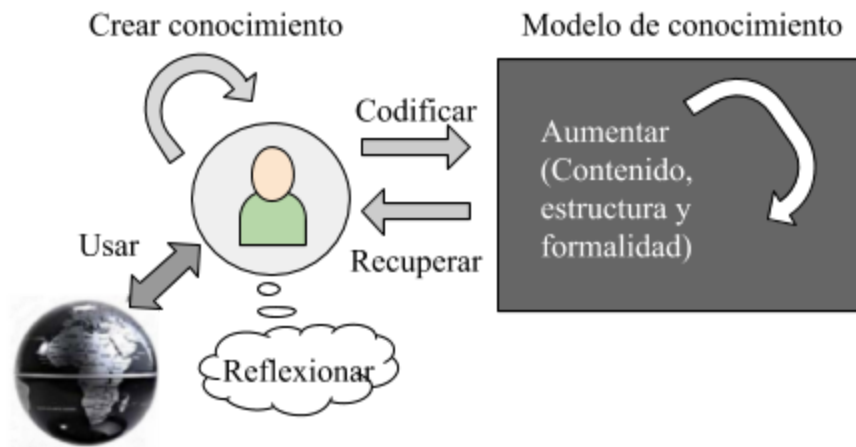
Völkel (2010) define un modelo para la gestión de conocimiento personal que se enfoca en la interacción entre el usuario y la herramienta. Parte de la perspectiva individual y se basa en la

administración de Señales de conocimiento². Los procesos que conforman el modelo, cuando el usuario realiza la gestión de su conocimiento personal aisladamente, son:

- Adquisición de conocimiento: Es un proceso mental donde el nuevo conocimiento es adquirido.
- Codificación de conocimiento: Proceso donde se asocia un modelo mental con un patrón creado de forma digital. Representa la creación inicial de Señales de conocimiento.
- Extensión de conocimiento: Es un proceso complejo que envuelve la extensión de las Señales de conocimiento previamente creadas y el cambio de estructura entre múltiples Señales de conocimiento. Incluye tres subprocesos que se describen a continuación:
 - Extender contenido: Durante este subproceso se extiende la secuencia de caracteres de las Señales de conocimiento.
 - Estructuración: Significa la creación de relaciones entre Señales de conocimiento,
 - Formalización: Requiere la existencia de estructura para la asignación de metadatos con el fin de explicar las semánticas de las estructura.
- Recuperación de conocimiento: En este proceso el usuario interacciona con su herramienta para la gestión de conocimiento personal para encontrar Señales de conocimiento previamente creadas.
- Uso de conocimiento: Representa el uso del conocimiento en situaciones del mundo real.

La figura siguiente explica gráficamente el ciclo de vida de las Señales de conocimiento, cuando el usuario realiza la gestión de conocimiento personal aisladaamente.

² Traducción literal de *Knowledge cues*.

Figura N° 2.3.**Ciclo de vida de Señales de conocimiento aisladamente**

Fuente: Völkel, 2010.

2.6. WEB SEMÁNTICA

La web semántica es “un extensión de la web actual, en la que se le da a la información un significado definido, facilitando el trabajo cooperativo de las computadoras y las personas.” (Berners-Lee, Hendler, & Lassila, 2001).

La visión de la web semántica que indica Barceló, Sánchez y Pérez (2006), en referencia al sitio oficial del “Consortio mundial de la red”, es “tener datos en la red definidos y unidos de manera que puedan ser usados por computadoras, no solo con el propósito de visualización, sino para la automatización, integración y uso repetido de los datos en múltiples aplicaciones...”. Implica, además, que los usuarios puedan clasificar y descubrir la información de tareas que consumen mucho tiempo, de manera *off-line* y *on-line*.

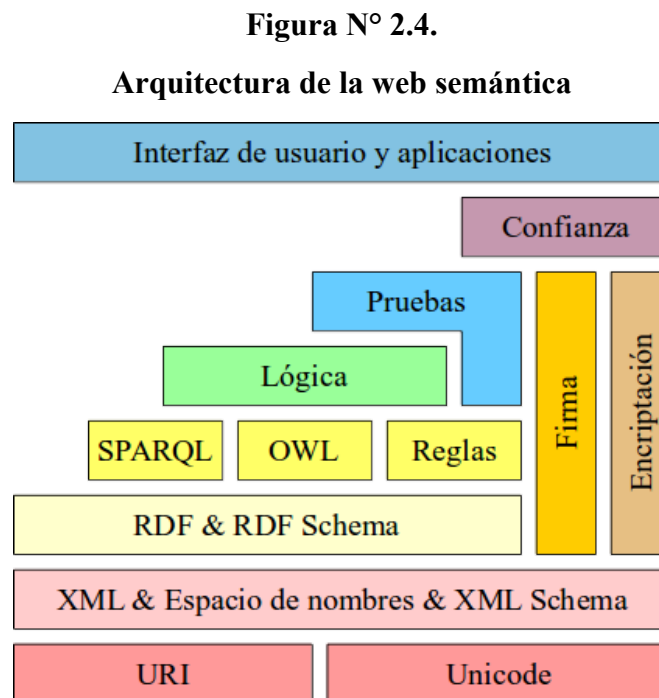
Típicos casos de uso de la web semántica, según Antoniou y van Harmelen (2008), son “mejorar los motores de búsqueda, la personalización de sitios Web y el enriquecimiento semántico de los sitios existentes.”. Además, los autores señalan que las principales fuentes de metadatos semánticos son fuentes automatizadas, como ser: la extracción de conceptos, reconocimiento de entidades y clasificación automática.

Según Hidalgo y Rodríguez (2013) la web semántica puede ser aplicada en:

- Gestión de documentos digitales: mediante el uso de anotaciones semánticas.

- Tesoros documentales: mediante servicios web basados en RDF.
- Visualización de información: usando las ventajas que ofrecen las ontologías.
- Gestión de información financiera y económica: para búsquedas más precisas.
- Entornos universitarios: mediante su uso en herramientas de gestión de conocimiento.
- Gestión de referencias bibliográficas: para el tratamiento de referencias.
- Buscadores semánticos: para la aplicación funcional en el uso diario.

Además, la W3C define la arquitectura de la web semántica de la siguiente manera:



Fuente: W3.org, 2007.

Obitko (2007) explica la función de cada capa de la arquitectura de la web semántica:

- URI y Unicode: Unicode es un estándar para la codificación de conjuntos de caracteres internacionales y permite que todos los lenguajes humanos puedan ser usados en la web de una manera estandarizada. URI, es una cadena de forma estandarizada que permite la identificación única de recursos.
- XML: Asegura el uso de una sintaxis común en la web semántica. XML es un lenguaje de marcado para documentos que contienen información estructurada.

- RDF: Es un marco de trabajo para la representación de información sobre recursos de manera gráfica. Está basado en triples (sujeto, predicado y objeto) que forman el grafo de datos.
- OWL: Es un lenguaje derivado de la lógica descriptiva. Está embebido en RDF y provee un vocabulario estandarizado adicional.
- *Rules*: Permiten la definición de condiciones más allá de aquellas provistas por RDF y OWL.
- SPARQL: Es un lenguaje similar a SQL que utiliza triples y recursos para la igualación de consultas y para la devolución de resultados. Puede ser usado para consultar ontologías y bases de conocimiento directamente.
- *Proof* y *Trust*: Permite definir la confiabilidad de los resultados mediante pruebas formales y entradas confiables.
- *Signature* y *Encryption*: Para contar con entradas confiables, estas capas permiten la verificación de los orígenes de información.
- *User interface*: Encima de las capas previamente mencionadas, se puede construir una interfaz para la interacción con el usuario.

2.6.1. ONTOLOGÍAS

Gruber (1993) define las ontologías como “una especificación formal y explícita de una conceptualización compartida. La "conceptualización" se refiere a un modelo abstracto de fenómenos en el mundo, al haber identificado los conceptos relevantes de esos fenómenos. “Explícito” significa que se definen explícitamente el tipo de conceptos utilizados y las limitaciones de su uso. “Formal” se refiere al hecho de que la ontología debe ser legible en máquina. “Compartido” reflexiona que la ontología debe captar el conocimiento consensual aceptado por las comunidades”. Además, son “esquemas de metadatos que proveen un vocabulario controlado de conceptos, cada uno con una semántica definida explícitamente y procesable por máquinas. Con la definición de teorías de dominios compartidas y comunes, las ontologías ayudan a ambos, personas y máquinas a comunicarse concisamente, apoyando al intercambio de semántica y no sólo sintaxis.” (Maedche & Staab, 2001).

Por su parte, Ding (2001) define a la ontología como “el término que hace referencia al entendimiento compartido de algunos dominios de interés. los cuales son a menudo concebidos como un conjunto de clases (conceptos), relaciones, funciones, axiomas e instancias.”. Además, una ontología “consiste en una lista finita de términos y las relaciones entre los mismos. Los términos denotan conceptos importantes (clases de objetos) del dominio. Las relaciones típicamente incluyen jerarquías de clases” (Antoniou & Van Harmelen, 2008).

2.6.1.1. Tipos de ontologías

Roussey, Pinet, Kang & Corcho (2011) explican la clasificación de ontologías basada en el alcance de las mismas:

- Ontologías locales: Son especializaciones de las ontologías de dominio en las que no puede haber consenso o intercambio de conocimientos. Este tipo de ontología representa el modelo particular de un dominio según un único punto de vista de un usuario o desarrollador.
- Ontologías de dominio: Sólo son aplicables a un dominio con un punto de vista específico. Es decir, este punto de vista define cómo un grupo de usuarios conceptualizan y visualizan un fenómeno específico.
- Ontologías de referencias centrales: Este tipo de ontología está vinculada a un dominio pero integra diferentes puntos de vista relacionados con un grupo específico de usuarios. Es el resultado de la integración de varias ontologías de dominio.
- Ontologías generales: Las ontologías generales no se dedican a un dominio o campos específicos. Contienen el conocimiento general de un sólo área.
- Ontología fundamentales: Son ontologías genéricas aplicables a varios dominios.

2.6.1.2. Ontologías facetadas ligeras

Una ontología facetada ligera es “una ontología ligera en la que los términos, presentes en cada etiqueta de nodo, y sus conceptos, están disponibles en la base de conocimiento que está organizada como un conjunto de facetas” (Giunchiglia, Dutta & Maltese, 2009).

Feroz (2010) explica la definición formal, como:

$$FLO = \langle LN, LE, LT, LC^{FL}, BK^F \rangle$$

Donde:

- LN : Es un conjunto de nodos finito.
- LE : Es un conjunto de aristas que representan las relaciones entre nodos.
- LT : Es un conjunto de términos.
- LC^{FL} : Es un conjunto finito de conceptos codificados en un lenguaje formal.
- BK^F : Es la base de conocimiento organizada como un conjunto de facetas.

Feroz (2010) además, afirma que la base de conocimiento juega un rol importante en las ontología facetadas ligeras. Define un ejemplo:

- Existe un término “pescado” en una etiqueta de un nodo en la jerarquía de una ontología ligera. Representa a los vertebrados acuáticos cuando la base de conocimiento es referente al dominio de animales. El mismo término representa la carne de pescado usada para el consumo humano cuando la base de conocimiento es sobre el dominio de comida.

La principal ventaja del enfoque facetado es que “explicita las relaciones lógicas entre los conceptos y grupos conceptuales y elimina las limitaciones de las jerarquías tradicionales. Permite visualizar una entidad compleja desde diferentes perspectivas o ángulos” (Giunchiglia, Dutta & Maltese, 2009).

2.6.1.3. Patrones de diseño ontológicos

Un patrón de diseño ontológico es “una solución de modelado para resolver un problema recurrente de diseño ontológico” (Gangemi & Presutti, 2009). Además, Gangemi y Presutti (2009) describen los tipo de patrones de diseño ontológico:

- Patrones de diseño lógicos: Patrones que son expresados en términos de vocabularios lógicos. Son independientes de un específico dominio de interés y dependen de la expresividad del formalismo lógico usado para la representación.
- Patrones de diseño arquitectónicos: Aquellos que afectan la forma general de la ontología. Pueden ser de dos tipos: Internos, definidos como colecciones de Patrones lógicos y Externos, definidos como constructores de meta-nivel, donde cada ontología forma un módulo.

- Patrones de diseño de razonamiento: Son aplicaciones de patrones de diseño lógicos orientados a la obtención de resultados de razonamiento determinados, basados en la conducta del motor de razonamiento.
- Patrones de diseño de correspondencia: Incluyen dos tipos de patrones: Patrones de reingeniería, que son reglas de transformación aplicadas para la creación de nuevas ontologías desde un modelo de origen y Patrones de mapeado, que hacen referencia a las posibles relaciones semánticas entre elementos mapeables.
- Patrones de diseño de presentación: Aquellos que lidian con la usabilidad y nivel de lectura de las ontologías desde las perspectiva del usuario.
- Patrones de diseño léxico sintácticos: Son estructuras lingüísticas que contienen ciertos tipos de palabras consecutivas en un orden específico.
- Patrones de diseño de contenido: Propone patrones para la resolución de problemas de diseño del dominio de clases y propiedades de la ontología.

2.6.1.4. Metodología NeOn

Según Mora y Segara (2016), la metodología NeOn, es una metodología para la construcción de ontologías que “emplea un proceso iterativo incremental, basado en escenarios, cada uno de ellos presenta actividades establecidas para obtener una ontología consistente y confiable, permitiendo la reutilización de recursos ontológicos y no ontológicos ya existentes sobre el dominio, con el fin de integrarlos a la ontología propuesta, generando una red ontológica”.

La metodología NeOn tiene por objetivo la construcción de redes ontológicas evitando los inconvenientes de metodologías anteriores a la misma. NeOn incluye los beneficios de colaboración de *Diligent*, la propuesta del análisis del alcance de *MethOntology* y el uso de preguntas de competencia de *On-To-Knowledge*. (Escandón, M. y Ñauta, F., 2011).

Gómez y Suárez (2009) presentan los diferentes escenarios que la metodología NeOn define, los cuales, fueron creados debido a la existencia de diversas formas de construcción de ontologías. A continuación se listan los nueve escenarios definidos por la metodología:

- Escenario 1: De la especificación a la implementación. La red ontológica es creada desde cero, sin la reutilización de recursos de conocimiento disponibles.

- Escenario 2: Reutilización y reingeniería de recursos no ontológicos. Se decide qué recursos no ontológicos pueden ser reusados en la red ontológica. También se cubre la tarea de la reingeniería de los recursos seleccionados, en ontologías.
- Escenario 3: Reutilización de recursos ontológicos. Los desarrolladores reusan los recursos ontológicos.
- Escenario 4: Reutilización y reingeniería de recursos ontológicos. Los desarrolladores reusan y realizan la reingeniería de los recursos ontológicos.
- Escenario 5: Reutilización y unión de recursos ontológicos. Este escenario es usado cuando los desarrolladores desean crear nuevos recursos ontológicos a partir de dos o más recursos existentes.
- Escenario 6: Reutilización, unión y reingeniería de recursos ontológicos. Los desarrolladores realizan la reingeniería de los recursos unidos.
- Escenario 7: Reutilización patrones de diseño ontológicos. Los desarrolladores acceden a los repositorios de patrones de diseño ontológicos para su reutilización.
- Escenario 8: Reestructurando recursos ontológicos. Los desarrolladores reestructuran los recursos ontológicos para integrarlos en la red ontológica.
- Escenario 9: Localizando recursos ontológicos. Los desarrolladores adaptan una ontología a otros idiomas y comunidades culturales, produciendo ontologías multilingües.

Suárez, Gómez y Fernández (2012) plantean un ciclo de vida en cascada para la construcción de redes ontológicas, donde se representa el desarrollo de la red como fases secuenciales. Las fases de este ciclo de vida son:

- Fase de inicio, en esta fase, es necesario crear un documento de especificación de requerimientos ontológicos (ORSO) que incluya los requerimientos que la red ontológica debe satisfacer, incluyendo conocimiento de un dominio concreto. En esta etapa se identifica al equipo de desarrollo, se establecen los recursos, responsabilidades y tiempo.
- Fase de diseño, el resultado de esta fase incluye dos modelos, un modelo informal y un modelo formal que satisface los requerimientos obtenidos en la fase anterior. El modelo formal no puede ser usado por computadoras pero puede ser reusado en otras redes ontológicas.

- Fase de implementación, en esta fase, el modelo formal es implementado en un lenguaje ontológico. El resultado es una ontología implementada en algún lenguaje para aplicaciones semánticas.
- Fase de mantenimiento, si en el uso de redes ontológicas se encuentran errores o conocimiento faltante, la equipo de desarrollo de la ontología debe regresar a la fase de diseño.

Los autores, además, identifican situaciones que se ajustan al desarrollo de redes ontológicas con el ciclo de vida en cascada previamente descrito:

- Proyectos ontológicos de corta duración.
- Proyectos ontológicos que quieran desarrollar una ontología existente en un idioma diferente.
- Proyectos ontológicos que se basen en recursos incluidos en el conocimiento, con un consenso previo.
- Proyectos ontológicos que incluyan un dominio pequeño.

2.6.2. ETIQUETAS SEMÁNTICAS

Según Xu, Fu, Mau y Su (2006), las etiquetas en la web semántica “permiten la clasificación y organización de la información para usar directamente los aportes de los usuarios finales, permitiendo el procesado automático del contenido Web”. Además, presentan propiedades de las etiquetas semánticas que servirán para el descubrimiento y la recuperación de contenido:

- Alta cobertura de múltiples facetas, las etiquetas genéricas facilitan la selección del contenido ingresado por el usuario debido a que puede ser usado en una mayor cantidad de objetos.
- Menor-esfuerzo, los usuarios deben poder encontrar objetos con la menor cantidad de etiquetas.
- Uniformidad, para evitar el uso de diferente etiquetas que refieran al mismo concepto.
- Exclusión de ciertos tipos de etiquetas, evitar el uso de etiquetas que dan sentido organizacional a la información, para el uso de otros usuarios.

2.6.2.1. Servicio web Open Calais

Open Calais es un servicio web de Thomson Reuters que agrega etiquetas de metadatos inteligentes a contenido no estructurado, permitiendo un potente análisis de texto. El motor de procesamiento de lenguaje natural de OpenCalais analiza y etiqueta automáticamente los archivos de entrada de tal manera que la aplicación que usa el servicio pueda identificar fácilmente datos relevantes y aprovechar de manera efectiva la inteligencia e ideas contenidas en el texto (OpenCalais API User guide, 2017).

OpenCalais analiza el contenido semántico de los archivos de entrada utilizando una combinación de estadísticas, aprendizaje automático y métodos personalizados basados en patrones. Desarrollado por el grupo de Servicios de metadatos de texto en Thomson Reuters, OpenCalais produce metadatos altamente precisos y detallados. (OpenCalais API User guide, 2017).

Los procesos que realiza durante el análisis de los datos de entrada, son los siguientes (OpenCalais API User guide, 2017):

- Reconocimiento de entidades nombradas: Open Calais obtiene los siguientes tipos de entidades:
 - Etiquetas de instancia, cada mención encontrada por Open Calais es expresada como una etiqueta de Instancia.
 - Etiquetas de identificación de entidades: Cada grupo de una instancia que se considera un único objeto se expresa como una Entidad.
 - Etiquetas de relevancia: Indican la relevancia de cada entidad respecto al documento analizado.
 - Etiquetas de confianza: Etiqueta que indica el nivel de identificabilidad de cada entidad.
 - Etiquetas de desambiguación: Etiqueta que permite la identificación de una entidad en base al conjunto de datos de Thomson Reuters, asignando un identificador único.
- Etiquetado de calidad³: Open Calais extiende la información con etiquetas de metadatos

³ Traducción literal de: *aboutness tagging*.

que describen el contenido como un todo:

- Etiquetas sociales: Clasifican el documento basados en la folksonomía de Wikipedia.
- Etiquetas temáticas: Identifican los temas que se discuten en los documentos.
- Etiquetas industriales: Identifican las industrias relaciones al texto.

2.6.3. QUERYGEN

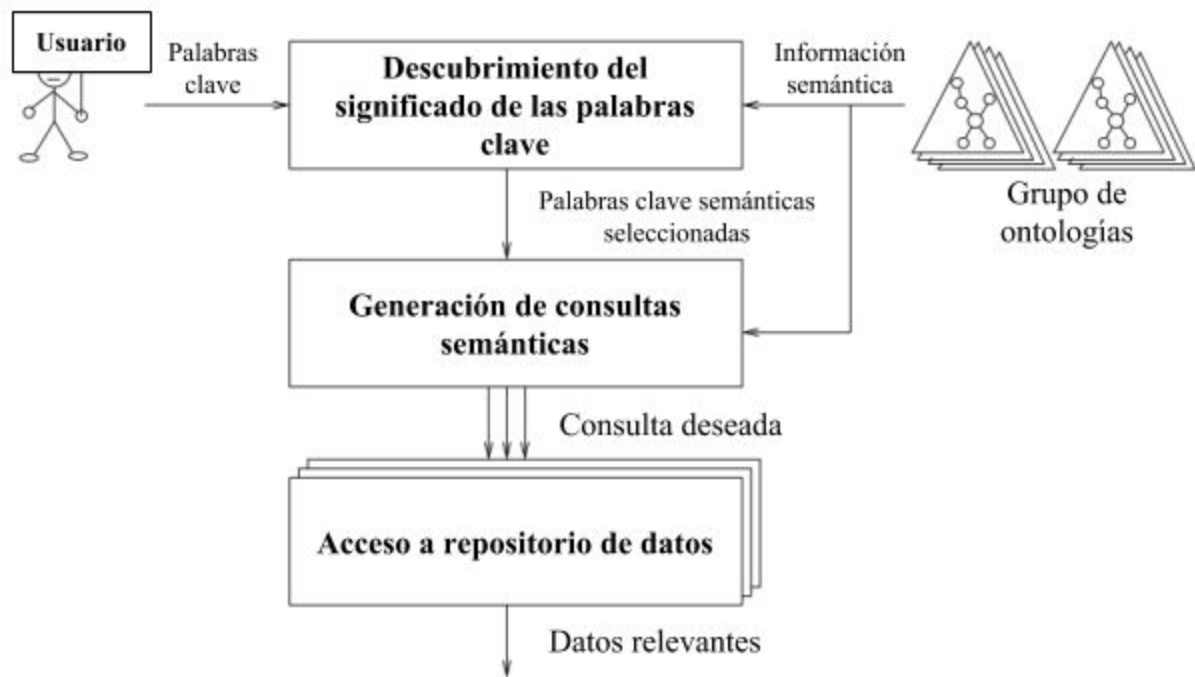
QueryGen (Bobed & Mena, 2016) es un sistema que propone un proceso generalizado de interpretación de palabras clave basado en modelos semánticos de distintos lenguajes de consulta para la búsqueda de contenidos.

El proceso incluye tres pasos, los cuales se describen a continuación (Bobed & Mena, 2016):

1. Definición de significado de palabras clave: En un primer paso, el sistema obtiene la semántica exacta de las palabras clave de entrada para transformarlas en palabras clave semánticas (palabras clave con semántica bien definida). Para ello, consulta un conjunto de ontologías para extraer los significados posibles de cada palabra clave, integrando los significados suficientemente similares para evitar redundancias. A continuación, el sistema aplica diferentes técnicas de desambiguación para establecer finalmente el significado de cada palabra clave teniendo en cuenta su contexto (los posibles significados del resto de las palabras clave).
2. Generación de Consultas Semánticas: Una vez que el significado de cada palabra clave ha sido establecido, el sistema construye automáticamente un conjunto de consultas formales que, combinando todas las palabras clave, representan la posible semántica que el usuario trata de representar cuando escribe la lista de palabras clave simples.
3. Acceso a los repositorios de datos: Por último, una vez validada la consulta generada que mejor se adapte a su significado, el sistema la remite a los repositorios de datos estructurados (bases de datos, puntos finales de datos enlazados, etc.) que dispondrán de la semántica de dicha consulta.

A continuación se muestra el proceso de creación de consultas semánticas, propuesto por el sistema QueryGen.

Figura N° 2.5.
Proceso de QueryGen



Fuente: Bobed & Mena, 2016.

2.7. BASES DE DATOS ORIENTADAS A GRAFOS

Las bases de datos orientadas son, según Robinson, Webber y Eifrem (2015), “sistemas de administración de base de datos con métodos de creación, lectura, modificación y eliminación que revelan un modelo de datos de grafos... Son generalmente optimizadas para el rendimiento transaccional.”. Además, identifican dos propiedades importantes para la tecnologías de bases de datos gráficas:

- El almacenamiento, algunas base de datos utilizan un almacenamiento gráfico nativo optimizado para la administración de grafos. Sin embargo, algunas herramientas suelen serializar los grafos en bases de datos relacionales.
- El motor de procesamiento, toda base de datos que se comporte como un grafo, desde la perspectiva del usuario, puede ser considerada una base de datos gráfica.

2.7.1. NEO4J

Neo4j es una base de datos orientada a grafos nativa robusta y escalable, desarrollada por Neo Technologies. Neo4j 2.0 fue lanzado en diciembre de 2013 e incluye un banco de trabajo para desarrolladores, una herramienta de visualización básica y su propio lenguaje de consulta gráfica: Cypher (Cambridge Intelligence, 2013).

Es una base de datos NoSQL orientada a grafos *open source* implementada en Java y Scala. Implementa el modelo de grafo de propiedades a nivel de almacenamiento, provee de características de bases de datos completas incluyendo transacciones *ACID*, soporte para *clusters* y conmutación por error en tiempo de ejecución con el fin de utilizarse en producción (Neo4j.com, 2017).

Algunas de las características de Neo4j, son (Neo4j.com, 2017):

- Materialización de relaciones de ordenación en el momento de la creación.
- Desplazamiento temporal constante para las relaciones en el grafo tanto en profundidad como en amplitud debido a la representación eficiente de nodos y relaciones.
- Almacenamiento compacto y almacenamiento en caché de memoria para grafos, que resulta en una ampliación eficiente y millones de nodos en una base de datos en hardware moderado.
- Codificado sobre la máquina virtual de Java (JVM).

Las principales diferencias con bases de datos relacionales, se muestran a continuación (Neo4j.com, 2017):

Cuadro N° 2.3.

Diferencias de Neo4j con Bases de datos relacionales

| | Base de datos relacional | Neo4j |
|---------------------------------|--|---|
| Almacenamiento de datos | Almacenamiento en tablas predefinidas con filas y columnas donde la información conectada está dividida en distintas tablas. | Estructura de grafo con adyacencia sin índices que resulta en transacciones más rápidas. |
| Modelo de datos | Transformado desde un modelo lógico a uno físico. Los tipos de datos deben ser conocidos previamente. | Coincidencia entre modelo lógico y físico. Los tipos de datos pueden ser cambiados o añadidos en cualquier momento. |
| Rendimiento de consultas | Depende de la cantidad de <i>JOINS</i> . | Independiente del nivel de profundidad de las relaciones. |
| Lenguaje de consulta | SQL | Cypher |

Fuente: Neo4j.com, 2017.

2.7.1.1. Grafo de propiedades

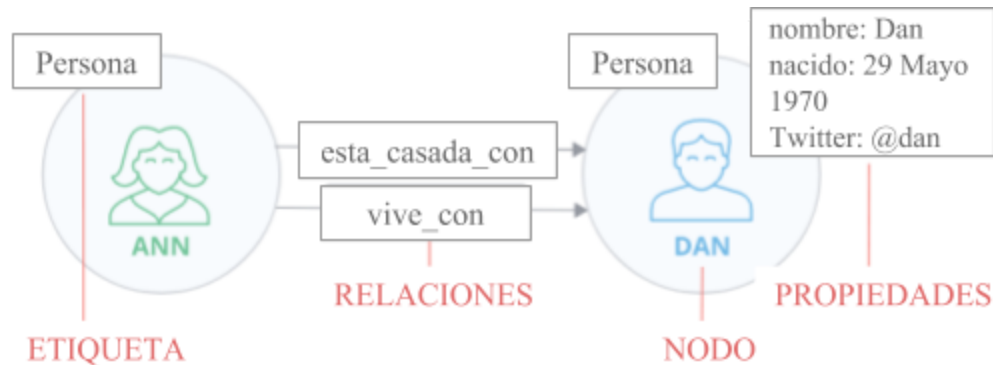
Los grafos de propiedades contienen entidades conectadas (nodos) que pueden contener cualquier cantidad de atributos (pares de clave - valor). Los nodos pueden tener etiquetas que representan sus diferentes roles en su dominio. Además de contextualizar las propiedades de los nodos y las relaciones, las etiquetas también pueden servir para adjuntar información de índice o restricción de metadatos a ciertos nodos (Neo4j.com, 2017).

Las relaciones proporcionan conexiones semánticamente relevantes dirigidas y nombradas entre dos entidades-nodo. Una relación siempre tiene una dirección, un tipo, un nodo de inicio y un nodo final. Al igual que los nodos, las relaciones pueden tener propiedades. En la mayoría de los casos, las relaciones tienen propiedades cuantitativas, como pesos, costos, distancias, calificaciones, intervalos de tiempo o fortalezas. Como las relaciones se almacenan de manera eficiente, dos nodos pueden compartir cualquier número o tipo de relaciones sin sacrificar el rendimiento. Tenga en cuenta que, aunque están dirigidos, las relaciones siempre se pueden

navegar independientemente de la dirección. (Neo4j.com, 2017)

Figura N° 2.6.

El modelo de grafo de propiedades



Fuente: Neo4j.com, 2017.

Donde (Neo4j.com, 2017):

- Nodos: Son el principal elemento de datos y están conectados a entre sí mediante relaciones. Pueden tener una o más propiedades y cuentan con etiquetas que describen su papel en el grafo.
- Relaciones: Conectan dos nodos, pueden ser direccionales y pueden tener propiedades. Se permiten las relaciones recursivas.
- Propiedades: Son valores con cadenas de caracteres como nombres. Pueden ser indexadas y restringidas.
- Etiquetas: Se usan para la agrupación de nodos y cada nodo puede contar con más de una etiqueta.

2.8. COMPETENCIAS

La norma UNE 66173 (2003) define competencia como un conjunto de "atributos personales y la aptitud demostrada para aplicar conocimientos y habilidades". Además, la normativa mencionada señala las dimensiones que integran el concepto de competencia:

- Atributos personales como comunicación, conocimiento, motivación y talento (Saber).

- Habilidades de las personas (Saber hacer).
- Las aptitudes demostradas y observables (Hacer) además de habilidades.

Según la OCDE (2005), una competencia “implica la capacidad de satisfacer demandas complejas, aprovechando y movilizand recursos psicosociales (incluyendo habilidades y actitudes) en un contexto particular. Por ejemplo, la capacidad de comunicarse eficazmente es una competencia que puede basarse en el conocimiento del idioma, las habilidades prácticas de TI y las actitudes hacia aquellos con los que se está comunicando”.

2.9. MEDICIÓN DE EFICACIA DE LEWANDOWSKI

Lewandowski (2012) define un marco de trabajo para la medición de la eficacia aplicable en la recuperación de contenidos. El proceso define un conjunto de cinco etapas que se muestran a continuación.

Cuadro N° 2.4.

Etapas del marco de trabajo de Lewandowski

| Etapas | Tareas |
|-----------------------------|--|
| Selección de <i>queries</i> | <ul style="list-style-type: none"> - Temáticas, que deben reflejar las necesidades del usuario. - Descripción de consultas. - Generación de aspectos de consulta y temáticas. |
| Obtención de resultados | <ul style="list-style-type: none"> - Descripción de resultados. - Clasificación de resultados. - Otros elementos. |
| Presentación de resultados | <ul style="list-style-type: none"> - Posición de resultados. - Descripción del estado actual de la vista. - Elementos gráficos. |
| Valoración de resultados | <ul style="list-style-type: none"> - Selección de jurados. - Escalas. |
| Análisis de datos | <ul style="list-style-type: none"> - Relevancia de resultados. - Descripción de resultados. - Diversidad. - Otros análisis. |

Fuente: Lewandowski, 2012 (Adaptado).

2.9.1. VALOR-F

La precisión (P) y la sensibilidad (R) se han utilizado regularmente para medir el rendimiento de los sistemas de recuperación y extracción de información. La precisión se ocupa de los errores de sustitución e inserción, mientras que la sensibilidad se ocupa de los errores de sustitución y borrado. La medida valor-F, se ha definido como una combinación ponderada de P y R. (Makhoul, Kubala, Schwartz & Weischedel, 1999).

La fórmula de Valor-F se define a continuación (Hripcsak & Rothschild, 2005):

$$F = \frac{(1 + \beta^2) \times \text{recall} \times \text{precision}}{(\beta^2 \times \text{precision}) + \text{recall}}$$

Donde β permite asignar mayor peso a la precisión o sensibilidad, y se equilibran cuando $\beta = 1$.

De esa manera, el Valor-F es la representación del equilibrio existente entre la sensibilidad y la precisión.

2.10. UNIVERSIDAD DEL VALLE

En la actualidad cerca de 11.000 estudiantes nacionales y extranjeros en la Sede Central y Subsedes Académicas de Univalle desarrollan sus estudios superiores en la infraestructura universitaria con mayor avance tecnológico de Bolivia (Univalle.edu, 2017).

La facultad de Informática y electrónica cuenta con la carrera de Ingeniería de Sistemas Informáticos y la Universidad permite las modalidades de titulación: Examen de grado, Proyecto de grado y Trabajo dirigido.

La carrera de Ingeniería de Sistemas Informáticos cuenta con materias para el desarrollo de Proyectos de software. Las mismas, tienen por objetivo el desarrollo de Proyectos aplicados al área de trabajo y sirven de herramienta de integración de conocimientos para los estudiantes de la carrera. La última materia de Taller de Sistemas Informáticos dentro del pensum estudiantil, es enfocada al desarrollo de Proyectos de Grado, combinando ambos objetivos a fin de tener una presentación única para su evaluación. Los estudiantes deben proponer Proyectos que representen su aprendizaje en la carrera e incluir todas las fases del ciclo de vida del desarrollo de software.

2.10.1. MISIÓN

Divulgación y propagación del conocimiento científico y tecnológico, en la sociedad boliviana, de tal manera que, con estos instrumentos todos se conviertan en partícipes e impulsores del desarrollo nacional.

2.10.2. VISIÓN

La Universidad asume como visión institucional el concepto de que la formación profesional, junto con la utilización y aplicación plenas del avance científico y tecnológico contribuirán a la creación de bienestar social.

CAPÍTULO III

INGENIERÍA DEL PROYECTO

CAPÍTULO III

INGENIERÍA DEL PROYECTO

Este capítulo presenta la definición de las variables que influyen en el proceso de gestión de conocimiento personal, diseña el modelo ontológico que es utilizado para la implementación de la herramienta y modela el método de búsqueda semántica, con la implementación de un sistema usando la metodología ICONIX. Finalmente, se mide la eficacia de la recuperación de conocimiento personal mediante el desarrollo de ontologías de prueba con la metodología NeOn.

3.1. DEFINICIÓN DEL PROCESO DE GESTIÓN DE CONOCIMIENTO PERSONAL

El modelo para la gestión de conocimiento personal que implementa el presente proyecto sigue el ciclo de vida de Señales de conocimiento definido por Völkel⁴.

3.1.1. MODELO DE VÖLKE

El modelo define un conjunto de procesos que se enfocan en la interacción entre personas y herramientas cuando el usuario realiza la gestión de conocimiento aisladamente. A continuación se describe detalladamente la aplicación de los procesos del modelo de Völkel en la herramienta desarrollada en el presente documento.

3.1.1.1. Adquisición de conocimiento

Proceso que el estudiante realiza mediante la interacción con el entorno. Es dependiente de nivel de aprendizaje del estudiante, nivel de enseñanza del docente, cultura, estado socioeconómico y el entorno del estudiante en general. En consecuencia, la herramienta desarrollada no puede influir sobre los factores previamente mencionados y por ende no realiza el soporte al proceso de adquisición de conocimiento.

3.1.1.2. Codificación de conocimiento

Proceso que representa la creación inicial de Señales de conocimiento de manera digital. En este proceso cada estudiante plasma el conocimiento adquirido en el proceso anterior. Se ajusta al

⁴ Véase en Capítulo 2: Modelo de Völkel.

proceso de Externalización del modelo SECI⁵ que permite la transformación de conocimiento tácito a conocimiento explícito. El principal caso de uso para la codificación de conocimiento se centra en la toma de notas, las cuales representan entidades que pueden ser extendidas y recuperadas en procesos siguientes del modelo de Völkel. Durante este proceso, la herramienta tiene impacto directo, como soporte para el estudiante, al crear una interfaz de creación de notas permitiendo la codificación de las mismas.

3.1.1.3. Extensión de conocimiento

Proceso que implica la extensión de las Señales de conocimiento creadas durante el proceso anterior. Representa al proceso de Combinación del modelo SECI. Incluye tres subprocesos.

- Extensión de contenido: Este subproceso implica la extensión de contenido de las Señales de conocimiento existentes mediante la adición de contenido textual, definición de formato del texto e inclusión de imágenes y archivos. La herramienta permite la edición de Señales de conocimiento previamente creadas y crea una interfaz para el manejo de revisiones.
- Estructuración: Subproceso que incluye la creación de relaciones entre Señales de conocimiento. Permite la referencia entre entidades para formar una estructura de forma de árbol. Cada Señal de conocimiento existente puede extender su significado mediante la creación de nuevas Señales de conocimiento que refieran a la entidad original. La herramienta permite la creación de relaciones entre Señales de conocimiento mediante referencias textuales y referencia de entidades.
- Formalización: En este subproceso se agrega metadatos a las Señales de conocimiento creadas anteriormente, con el fin de explicar la semántica de los contenidos. En este subproceso se hace la asignación de etiquetas semánticas a las Señales de conocimiento. Las etiquetas semánticas asignadas en este subproceso debe contar con una estructura formal y representan una base de conocimiento extensible por el usuario. El análisis semántico de las Señales de conocimiento cuenta con una interfaz dedicada dentro de la herramienta y la misma permite además, la creación de etiquetas personalizadas.

⁵ Véase capítulo 2: Transformación de conocimiento.

3.1.1.4. Recuperación de conocimiento

Durante este proceso, el usuario interacciona con la herramienta para la gestión de conocimiento personal. De esa manera, es capaz de encontrar Señales de conocimiento que sean de utilidad para sus objetivos. Representa al proceso de Internalización del modelo SECI.

Se diseñó una arquitectura semántica para la implementación de la herramienta que permitirá la Recuperación de conocimiento mediante la ejecución de consultas a una base de datos orientada a grafos, donde se generan ontologías mediante las Señales de conocimiento que el usuario crea. La estructura resultante se define en el Diseño del modelo ontológico.

3.1.1.5. Uso de conocimiento

Proceso que representa el uso de conocimiento en situaciones del mundo real. Implica la aplicación de las Señales de conocimiento recuperadas después de haber sido extendidas, estructuradas y formalizadas. La herramienta desarrollada en el presente proyecto no crea un efecto directo en este proceso, puesto que las habilidades para la uso de conocimiento son características individuales de cada estudiante.

3.1.2. APLICACIÓN DEL MODELO DE VÖLKE

A continuación se listan los procesos del modelo de Völkel y su forma de aplicación en la herramienta desarrollada respecto a la descripción anterior.

Cuadro N° 3.1.

Variables de proceso del modelo de gestión de conocimiento de Völkel.

| Proceso | Variable | Forma de aplicación |
|-----------------------------------|--|--|
| Adquisición de conocimiento | Entorno Nivel de aprendizaje | Ninguno |
| Codificación de conocimiento | Creación de Señales de conocimiento | Interfaz para administración de notas |
| Extensión: Extensión de contenido | Estructura textual | Manejo de revisiones |
| Extensión: Estructuración | Relaciones entre Señales de conocimiento | Interfaz para administración de relaciones |
| Extensión: Formalización | Metadatos | Interfaz para administración de términos taxonómicos |
| Recuperación de conocimiento | Uso de semántica | Interfaz para ejecución de queries |
| Uso de conocimiento | Habilidades para la aplicación | Ninguno |

Fuente: Elaboración propia, 2017.

3.1.3. INDEXACIÓN DE SEÑALES DE CONOCIMIENTO

Siguiendo los procesos del modelo previamente descrito se debe definir una manera de indexar las Señales de conocimiento creadas por el usuario, mediante el uso de facetas. Giunchiglia, Dutta & Maltese (2009) definen tres intuiciones para este proceso, que ajustados al modelo de gestión de conocimiento personal del sistema propuesto, se definen como:

- Asociar a cada Señal de conocimiento las etiquetas semánticas correspondientes desde una Base de conocimiento.
- Para cada Señal de conocimiento en una faceta, el contexto se construye asociando todas las Señales de conocimiento desde la raíz hasta sí mismo, con el fin de reducir su ambigüedad.
- El usuario debe proveer al menos un conjunto de etiquetas para la creación de una faceta.

A continuación se muestra un ejemplo de indexación donde cada fila representa las facetas

pertenecientes y “>” identifica la extensión más específica de las ideas dentro de una faceta.

Cuadro N° 3.2.
Ejemplo de indexación de conocimiento

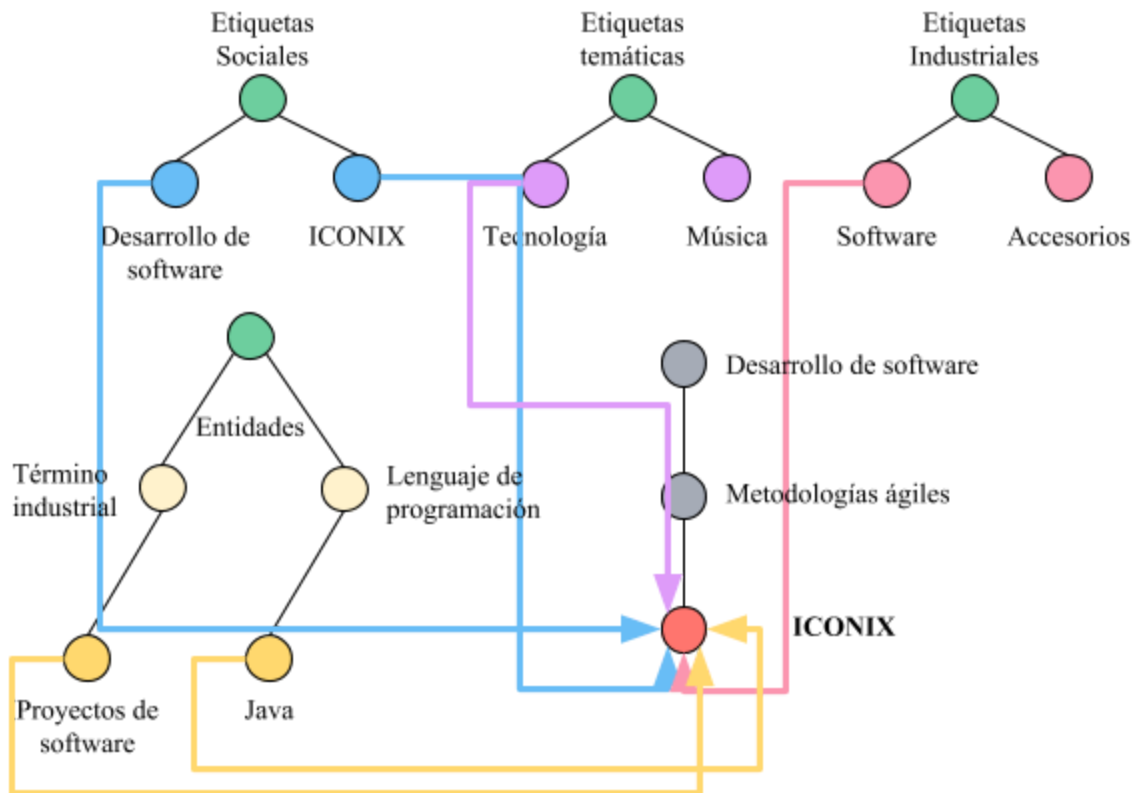
| | |
|------------------------|---|
| Señal de conocimiento | ICONIX |
| Entidades | Término industrial > Proyectos de software Lenguaje de programación > Java |
| Etiquetas temáticas | Tecnología |
| Etiquetas sociales | Desarrollo de software / ICONIX |
| Etiquetas industriales | Software |

Fuente: Elaboración propia, 2017.

De esa manera, cada concepto plasmado en una Señal de conocimiento puede ser indexado mediante su asociación con etiquetas semánticas, permitiendo así, la creación de una estructura de facetas y la recuperación de Señales de conocimiento desde distintas perspectivas. El sistema es capaz de reducir los pasos para una indexación en facetas mediante el análisis automático de los contenidos y la implementación de una interfaz para la selección de metadatos a asignar. En consecuencia, se crean ontologías facetadas ligeras con los contenidos ingresados por los usuarios.

La siguiente figura demuestra la creación de facetas mediante el uso de una base de conocimiento para la clasificación de las Señales de conocimiento creadas por el usuario.

Figura N° 3.1.
Indexación de Señales de conocimiento



Fuente: Elaboración propia, 2017.

De esa manera, se aprovechan las ventajas del enfoque por facetas para eliminar las limitaciones de jerarquías tradicionales y permite observar las entidades desde una variedad de perspectivas.

3.1.4. COMPETENCIAS

La Universidad Privada del Valle define competencias que los estudiantes deben adquirir durante su desarrollo académico y describen el enfoque de la Universidad en el desarrollo de Proyectos de software.

La evaluación por competencias permite medir las habilidades de los estudiantes en determinadas condiciones. En ese sentido, se consideran las siguientes dimensiones: el “saber” (conocimientos, enfoques y teorías) y el “saber hacer” (habilidades). Debido a que el “ser” envuelve las actitudes del individuo, no puede garantizarse la integridad de su medición.

Es así, que a través de las competencias definidas por la Universidad Privada del Valle para las materias de Taller de Programación I, II y Taller de Sistemas I, II y III, se realiza la identificación de los elementos que los componen, a continuación:

Cuadro N° 3.3.

Selección de conocimientos de las competencias

| Código | Conocimientos (Saber) |
|---------------|--|
| CP-S-1 | Describe las relaciones de un sistema informático a través de la elección de un banco de datos que permita estructurar un prototipo relacionado con el campo de trabajo del ingeniero. |
| CP-S-2 | Conoce e interpreta el desarrollo metodológico para la implementación, mantenimiento y puesta en marcha de un sistema informático aplicado que permita estructurar un prototipo relacionado con el campo de trabajo del ingeniero. |
| CP-S-3 | Describe el desarrollo del prototipo, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-S-4 | Conoce los procedimientos para realizar un proyecto informático para realizar alguna aplicación móvil, valorando su importancia en el campo de trabajo de la ingeniería. |
| CP-S-5 | Conoce los procedimientos que le permitan realizar un proyecto informático para desarrollar alguna de las siguientes aplicaciones: juegos educativos, recreativos, tutoriales interactivos y/o aplicaciones multimedia valorando su importancia en el campo de trabajo de la ingeniería. |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.4.
Selección de capacidades de las competencias

| Código | Capacidades (Saber hacer) |
|---------------|---|
| CP-SH-1 | Implementar un sistema informático aplicado, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-SH-2 | Realiza pruebas a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-SH-3 | Elabora documentación, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-SH-4 | Aplica algún lenguaje de programación, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-SH-5 | Aplica lenguajes de programación: Visual Basic, Visual C, C/C++/C#, Java, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |
| CP-SH-6 | Aplica la tecnología .Net incluyendo ADO .Net, así como formularios Web, ASP .Net, XML y web services para la implementación de un sistema informático relacionado con el campo de trabajo del ingeniero. |
| CP-SH-7 | Identificar el nivel de complejidad de elaboración de un sistema, a través de la resolución de problemas que se presentan en el campo de trabajo del profesional. |

Fuente: Elaboración propia, 2017.

De esa manera, se cuenta con el enfoque de la Universidad para el desarrollo de Proyectos de software y se pueden identificar las habilidades requeridas por los estudiantes para su aprobación de las materias de Taller de Programación y Taller de Sistemas.

3.2. DISEÑO DEL MODELO ONTOLÓGICO

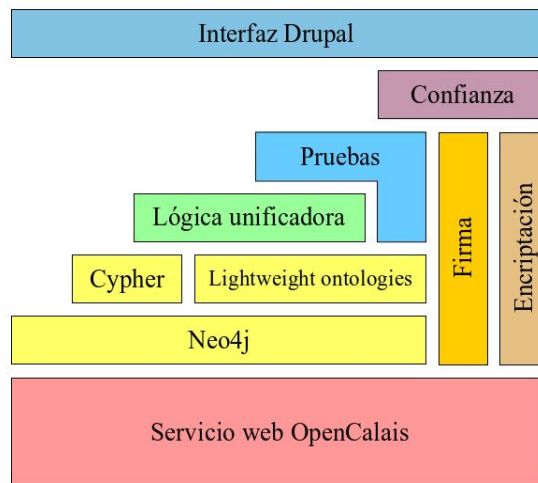
El diseño del modelo ontológico requiere la definición de la arquitectura semántica que el sistema implementa. Se describe la interacción entre capas de la arquitectura y el uso de las mismas para la gestión de conocimiento personal.

3.2.1. ARQUITECTURA SEMÁNTICA

El presente proyecto propone una arquitectura basada en la arquitectura de la Web Semántica⁶ mediante la integración de servicios, herramientas y tecnologías. La arquitectura resultante permite el desarrollo del proceso de gestión de conocimiento personal definido en el punto anterior y define el grado de formalización del conocimiento que se ingresa en la herramienta. A continuación se describen las diferentes capas de la arquitectura resultante y sus equivalencias.

Figura N° 3.2.

Arquitectura semántica resultante



Fuente: Elaboración propia (Basado en la arquitectura de la web semántica), 2017.

3.2.1.1. Servicio web Open Calais

El servicio web Open Calais forma la capa inferior de la arquitectura y reemplaza las capas de “URI”, “Unicode” y “XML & XML Schema” de la arquitectura original de la Web semántica. El servicio se encarga del análisis semántico de las Señales de conocimiento a fin de sugerir el uso de etiquetas semánticas para la clasificación de los contenidos.

Las llamadas al servicio web Open Calais cuentan con las siguientes características:

1. Se realizan mediante una interfaz HTTP REST a la dirección <https://api.thomsonreuters.com/permid/calais> con el método POST.

⁶ Véase Capítulo 2: Web semántica.

2. El tamaño de los datos de entrada al servicio están limitados a 100 KB por solicitud, excluyendo las cabeceras HTTP.
3. El servicio cuenta con soporte para los idiomas Inglés y Español. Dependiendo de la extensión del contenido a analizar, el servicio identifica automáticamente el lenguaje del texto. No se hará uso de la cabecera `x-calais-language` debido a que se espera que el usuario pueda ingresar contenido en ambos idiomas.
4. Las cabeceras de la solicitud incluyen los siguientes parámetros:
 - a. *Content-Type*: indica el tipo MIME del texto enviado para ser analizado. Se utilizan los tipos de contenido *text/html* y *text/raw* en las llamadas al servicio.
 - b. *outputFormat*: define el formato de salida de la información. El valor por defecto de la cabecera es: *application/json*.
 - c. *x-ag-access-token*: especifica el token que el usuario ha recabado desde la dirección: <http://www.opencalais.com/opencalais-api/> y es un campo obligatorio para el funcionamiento del servicio web.
5. Aquellos parámetros de cabecera que no se encuentran especificados previamente usan los valores por defecto definidos por Open Calais.

A continuación se muestra un ejemplo de solicitud al servicio web con los datos requeridos y aquellos que son utilizados por la herramienta.

Cuadro N° 3.5.
Ejemplo de solicitud a OpenCalais

| |
|---|
| Dirección URL para la solicitud POST |
| <code>https://api.thomsonreuters.com/permid/calais</code> |
| Cabeceras |
| <i>Content-Type: text/html</i> <i>outputFormat: application/json</i> <i>x-ag-access-token:</i> [El token de autorización recabado por el usuario] |
| Cuerpo del mensaje |
| <body> Este es un contenido de ejemplo que será analizado por el servicio web Open Calais. </body> |

Fuente: Elaboración propia (Basado en la documentación de Open Calais), 2017.

La respuesta que el servicio web otorga, sigue los códigos de error HTTP estándar⁷, tiene el formato *json* y contiene los siguientes elementos:

1. El nodo *Info*: incluye el contenido enviado originalmente.
2. El nodo *Meta*: incluye información sobre el procesado como fechas y horas de envío, además de un identificador único por cada solicitud realizada.
3. El nodo *ComponentVersions*: especifica las versiones de los componentes utilizados para el procesamiento del contenido ingresado.
4. *Aboutness tags*: que describen el contenido y pueden pertenecer a alguno de los siguientes tipos de etiquetas:
 - a. *SocialTags*: etiquetas que clasifican el documento como un todo. Se basan en la folksonomía⁸ de Wikipedia.
 - b. *TopicTags*: etiquetas que determinan el tema que está siendo discutido en el documento. Son definidos por el Esquema de codificación de Thomson Reuters⁹ y

⁷ Las respuestas HTTP pueden encontrarse en <https://www.w3.org/Protocols/HTTP/HTRESP.html>.

⁸ Se definen como colecciones de metadatos que se usan para la clasificación de contenidos.

⁹ Recabado de: http://www.opencalais.com/wp-content/uploads/folder/ThomsonReutersOpenCalaisAPIUserGuideR11_2.pdf

el Consejo de telecomunicaciones de prensa internacional.

- c. *IndustryTags*: etiquetas que indican las industrias que están relacionadas con las compañías mencionadas en el contenido ingresado. Son definidos por la Clasificación empresarial de Thomson Reuters¹⁰.

5. Entidades nombradas: etiquetas resultantes del análisis del contenido ingresado y son definidas por Open Calais (cf. Anexo 1), mediante la búsqueda de mención de entidades.

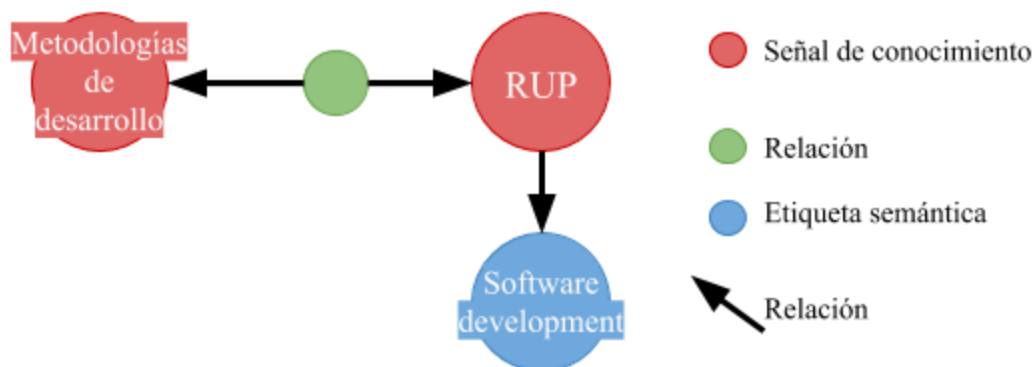
La presente capa extrae las etiquetas semánticas referentes al contenido ingresado y sirve de soporte al subproceso de Formalización en el proceso de gestión de conocimiento personal definido en el punto anterior. La estructura de etiquetas semánticas devuelta por el servicio formará la base de conocimiento para la clasificación del mismo.

3.2.1.2. Neo4j

Neo4j reemplaza la capa de “RDF y RDF Schema” de la arquitectura original de la web semántica. Almacena las etiquetas definidas en el punto anterior y las Señales de conocimiento ingresadas por el usuario con sus relaciones. Sigue el modelo de datos de los grafos de propiedades y permite la consulta mediante su propio lenguaje: Cypher. A continuación se muestra un ejemplo de grafo creado en Neo4j.

Figura N° 3.3.

Ejemplo de grafo Neo4j con datos de Drupal



Fuente: Elaboración propia, 2017.

¹⁰ Recabado de: <https://financial.thomsonreuters.com/content/dam/openweb/documents/pdf/financial/trbc-fact-sheet>

Los elementos que forman parte de los grafos de Neo4j y que son utilizados en la herramienta, se describen a continuación:

- **Nodos:** Los nodos en Neo4j representan las Señales de conocimiento que el usuario ingresa, los etiquetas semánticas resultantes del análisis de la capa anterior y las relaciones que se crean desde la interfaz gráfica. A continuación se muestran los diferentes tipos de nodos y sus características de almacenamiento en Neo4j.

Cuadro N° 3.6.

Tipos de nodos y características de almacenamiento

| Tipo de nodo | Etiqueta (<i>label</i>) | Propiedades |
|-----------------------|--------------------------------|--|
| Señal de conocimiento | entity:node | Título Descripción <i>Nid</i> (No visible por el usuario) <i>Path</i> (No visible por el usuario) <i>Content created</i> (No visible por el usuario) <i>Field URL</i> (No visible por el usuario) |
| Etiquetas semánticas | entity:taxonomy_term | Nombre Vocabulario <i>Tid</i> (No visible por el usuario) |
| Relación | entity:relation | Tipo de relación <i>Relation ID</i> (No visible por el usuario) |

Fuente: Elaboración propia, 2017.

- **Relaciones:** Las relaciones almacenadas en Neo4j representan los campos de referencia definidos en Drupal¹¹ que pueden contar con un número infinito¹² de referencias. Cada Señal de Conocimiento referenciada definirá una relación en Neo4j con una entidad media de tipo Relación. A su vez, las etiquetas semánticas obtenidas con la capa anterior estarán relacionadas directamente con las notas que etiquetan.

Es así, que la capa de Neo4j permite almacenar las etiquetas devueltas por Open Calais y las

¹¹ Campos de tipo *Entity reference*.

¹² En base a la interfaz gráfica de Drupal que permite los valores ilimitados (*unlimited*).

notas de Señales de conocimiento ingresadas por el usuario con sus respectivas relaciones. Crea una estructura para el almacenamiento de los contenidos que se ingresan en la herramienta y es de utilidad para la representación de los mismos.

3.2.1.3. Cypher

Cypher reemplaza a la capa del lenguaje “SPARQL” de la arquitectura original de la web semántica. Se encarga de la selección de conocimiento mediante consultas a la base de datos Neo4j. Es aplicable en toda la extensión del grafo de propiedades y servirá de herramienta para la recuperación de conocimiento. Permite diferenciar los diferentes tipos de nodos, las relaciones entre los mismos y el filtrado mediante sus propiedades. Además, permite la consulta y selección de todos los elementos creados en la capa de Neo4j. El uso del lenguaje Cypher para la búsqueda de conocimiento se describe durante la definición del Método de búsqueda semántica en el presente capítulo.

3.2.1.4. Ontologías facetadas ligeras

La capa de Ontologías facetadas ligeras¹³ reemplaza las capas de “OWL” y “Rules” de la arquitectura original de la web semántica. Tiene el objetivo de clasificar las Señales de conocimiento que el usuario codifica en la herramienta.

Para la construcción de ontologías facetadas ligeras se define: el Esquema de representación facetado¹⁴, que representa el conocimiento previo para la clasificación de entidades y el Patrón de diseño ontológico¹⁵ (ODP), que es usado para el desarrollo de ontologías.

El Esquema de representación facetado, utilizado en el presente proyecto, sigue la clasificación de dominios que el servicio web Open Calais ofrece y representa el conocimiento previo que permite el uso de semántica. El servicio separa los dominios en Entidades, Etiquetas temáticas, Etiquetas industriales y Etiquetas sociales. De esa manera, el conocimiento que el sistema administra, puede ser clasificado bajo un esquema general que agrupa los conceptos en categorías específicas.

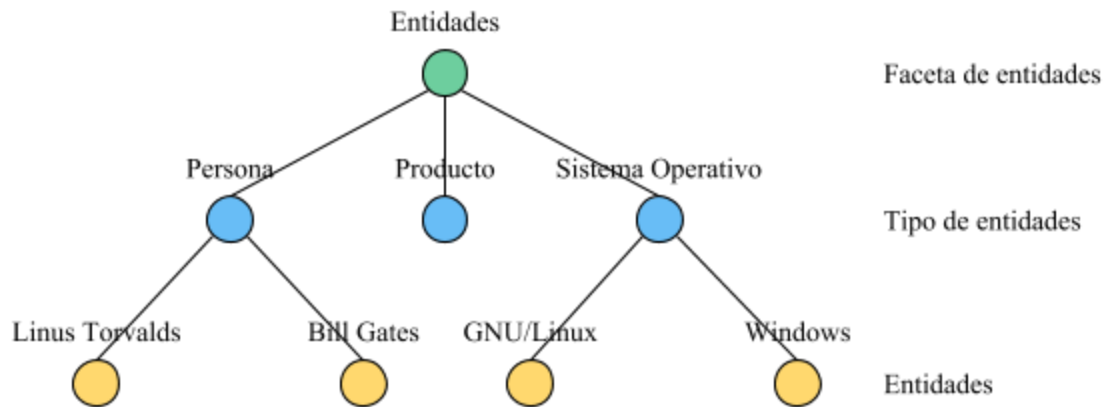
¹³ Traducción literal de *Faceted lightweight ontologies*.

¹⁴ Traducción literal de *Faceted representation scheme*.

¹⁵ Traducción literal de *Ontology design pattern*.

A continuación, se muestran ejemplos de facetas definidas mediante el uso del servicio web Open Calais.

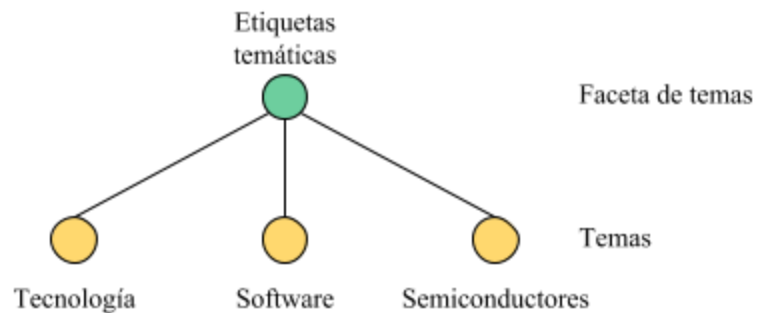
Figura N° 3.4.
Ejemplo de facetas de entidades



Fuente: Elaboración propia, 2017.

Las Señales de conocimiento son clasificadas mediante el uso de las etiquetas de entidades definidas e identificadas por el servicio web Open Calais, creando así, la faceta de entidades.

Figura N° 3.5.
Ejemplo de facetas de temas

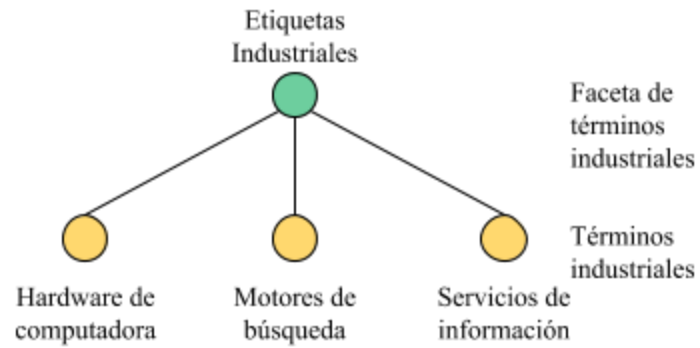


Fuente: Elaboración propia, 2017.

Las etiquetas temáticas definidas e identificadas por Open Calais permiten la creación de la

faceta de temas para la clasificación de Señales de conocimiento.

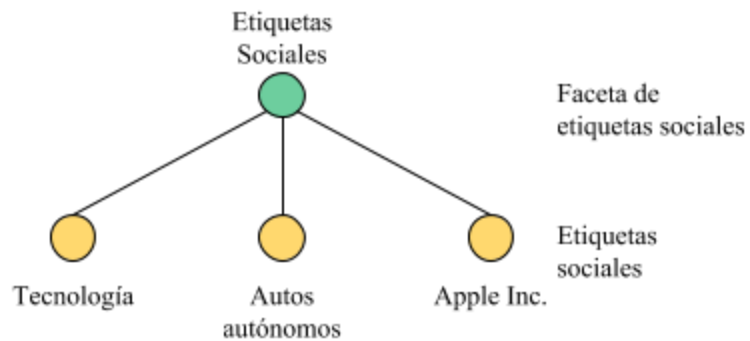
Figura N° 3.6.
Ejemplo de facetas de términos industriales



Fuente: Elaboración propia, 2017.

Las etiquetas industriales definidas e identificadas por Open Calais conforman la faceta de términos industriales.

Figura N° 3.7.
Ejemplo de facetas de etiquetas sociales



Fuente: Elaboración propia, 2017.

Las etiquetas sociales que define e identifica Open Calais permiten la creación de las facetas de términos sociales.

En consecuencia, la presente capa sirve de soporte en el subproceso de Estructuración de Señales de conocimiento definido en el modelo de gestión de conocimiento personal que se implementa en la herramienta. Define el Esquema de representación facetado a través del uso de las etiquetas semánticas definidas en la capa Open Calais, además de los tipos de relaciones existentes entre Señales de conocimiento y etiquetas para la creación de facetas.

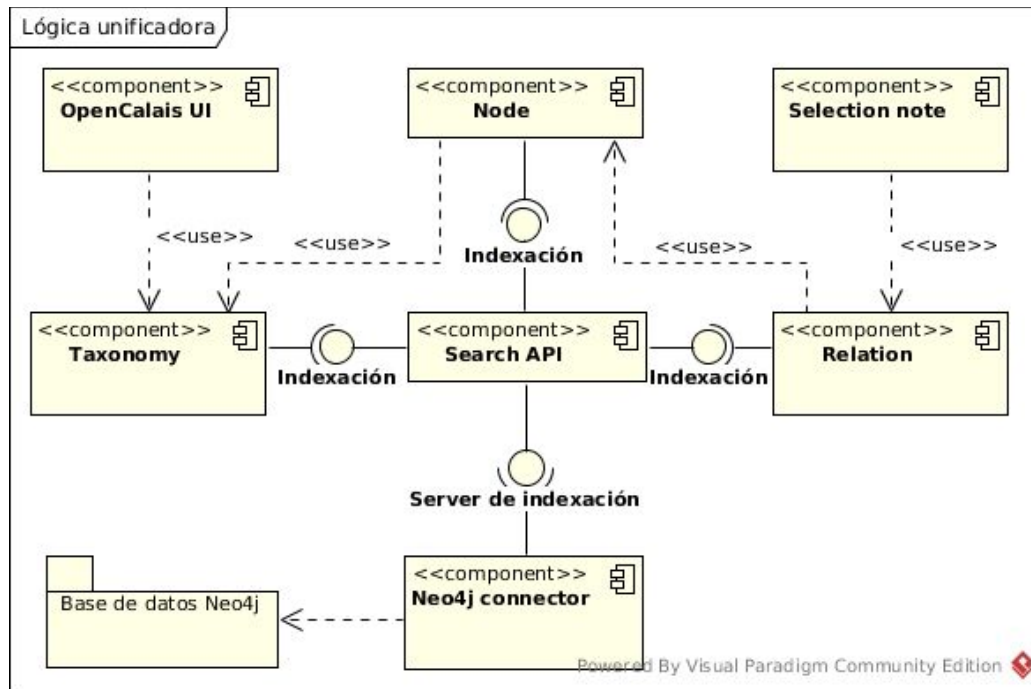
3.2.1.5. Lógica unificadora

La capa de lógica unificadora se encarga de la unión de funcionalidades para el proceso de gestión de conocimiento personal. Permite la interacción entre la base de datos Neo4j, las etiquetas semánticas definidas por Open Calais, las Señales de conocimiento creadas por el usuario las relaciones creadas mediante la definición de ontologías facetadas ligeras y el núcleo de Drupal para la funcionalidad del sistema. Hace uso de módulos disponibles para la comunidad, a fin de permitir la interacción entre capas.

A continuación se muestran los componentes que formarán parte de la capa de lógica unificadora.

Figura N° 3.8.

Diagrama de componentes de la capa lógica unificadora



Fuente: Elaboración propia, 2017.

De esa manera, las etiquetas semánticas que el servicio Open Calais extrae de los contenidos analizados, son almacenadas como términos taxonómicos, las Señales de conocimiento que el usuario crea, se almacenan como entidades de tipo Nota y las relaciones definidas entre Señales de conocimiento son almacenadas como entidades de tipo Relación.

En consecuencia, los siguiente módulos forman parte de la capa unificadora:

- *Selection note*: Permite la creación de relaciones entre entidades de tipo Nota en la interfaz Drupal. Utiliza el módulo *Relation* que permite la referenciación de entidades entre sí y el módulo *Node* del núcleo de Drupal para la creación de entidades de tipo contenido. Cada relación creada con el módulo puede contener campos y cuenta con un identificador único.
- *OpenCalais UI*: Módulo que se encarga de realizar las solicitudes al servicio web Open Calais, se encarga de almacenar el token del usuario para el uso del servicio y muestra los resultados como términos taxonómicos posibles. El módulo Open Calais usa las

funcionalidades que brinda el núcleo de Drupal con el módulo *Taxonomy* para la creación de vocabularios taxonómicos.

- *Neo4j connector*: Módulo que realiza la sincronización de entidades Drupal con nodos Neo4j. El módulo cuenta con los siguiente elementos para la indexación de entidades Drupal:
 - *Backend plugin*: hace uso del módulo *search api* para poder transferir las entidades de Drupal a nodos de Neo4j.
 - *Mapping processor*: se encarga de mapear los campos para agregarlos como propiedades de nodo en neo4j.

La capa de lógica unificadora permite la interacción de las capas inferiores y la interfaz de usuario. A fin de integrar en la arquitectura los módulos previamente mencionados, se ha realizado solicitudes de funcionalidad¹⁶ y se ha contribuido a los mismos con código escrito en PHP, con el objetivo de agregar las funcionalidades requeridas faltantes.

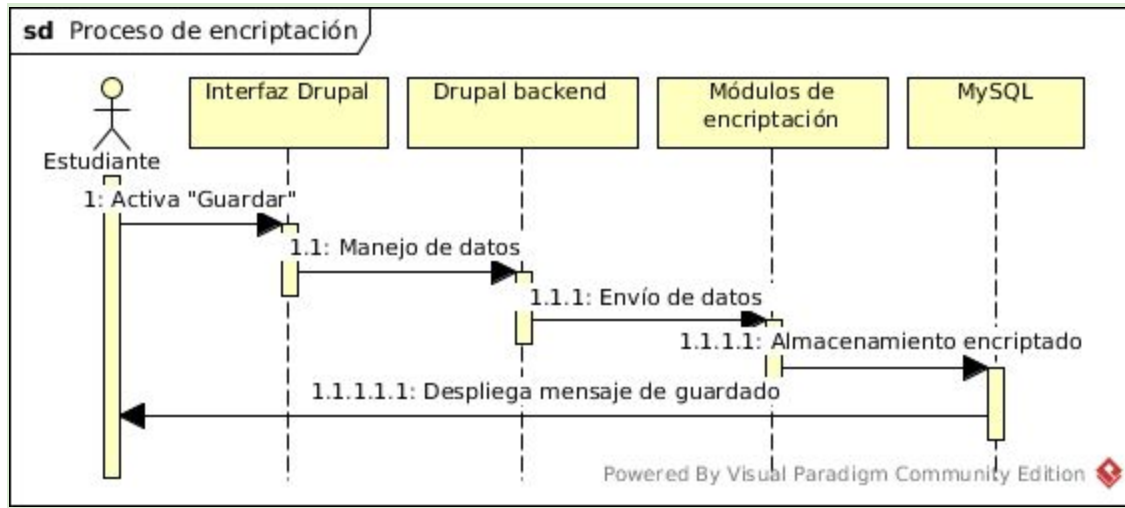
3.2.1.6. Pruebas y Encriptación

La capa de pruebas permite verificar que la información proviene del origen definido y no ha sido alterada. En el desarrollo de la herramienta del presente documento se combinaron las capas de Prueba y Encriptación a fin de asegurar la confidencialidad de la información y la fuente de la misma.

Dentro de los módulos existentes en la comunidad de Drupal, existen aquellos que ofrecen un *API* para realizar la encriptación de campos de una entidad. El módulo utilizado en el desarrollo de la herramienta es: *Encrypt* que hace uso del método de encriptación *Mcrypt AES* y provee una interfaz mediante el módulo *Field encryption*. A continuación se muestra un diagrama de secuencias que describe el proceso de encriptación de la herramienta.

¹⁶ Traducción literal de *Feature request*.

Figura N° 3.9.
Proceso de encriptación



Fuente: Elaboración propia, 2017.

3.2.1.7. *Confianza*

A diferencia de las anteriores capas, no existe una especificación para su implementación. La capa de confianza tiene el objetivo de probar que información encontrada en la Web Semántica sea correcta.

A fin de implementar la capa de Confianza en la arquitectura resultante de la presente herramienta, se hizo uso de los parámetros que ofrece Open Calais en cada etiqueta extraída con el servicio web. A continuación se describen los tipos de etiquetas extraídas por Open Calais y sus atributos de confianza.

Cuadro N° 3.7.**Atributos de confianza definidos por Open Calais**

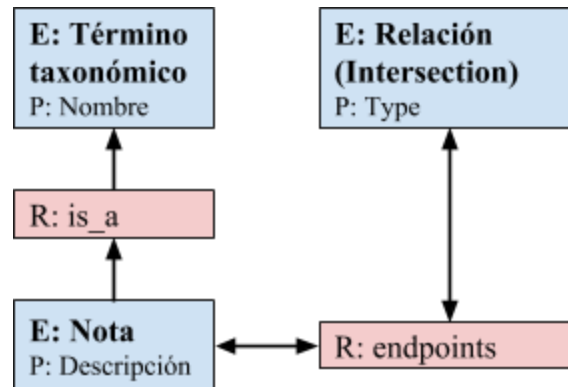
| Etiqueta | Atributo | Descripción | Escalas |
|---------------------|-------------------------|--|---|
| Etiqueta social | <i>Importance</i> | Indica la centralidad del tema mencionado por la etiqueta es para el documento. | 1 (Muy central) 2 (Central) 3 (Menos central) |
| Etiqueta temática | <i>Score</i> | Indica la probabilidad de que el tema sea discutido en el documento. A mayor valor, mayor probabilidad. | Entre 0 y 1. |
| Etiqueta industrial | <i>Relevance</i> | Indica cuán relevante es la industria para el documento. A mayor valor, mayor relevancia. | Entre 0 y 1. |
| Entidad | <i>Confidence level</i> | Indica el valor en que una compañía es, por ejemplo, realmente una compañía. | Entre 0 y 1. |

Fuente: Elaboración propia, 2017.

En la interfaz para la selección de etiquetas, se muestran los atributos definidos en el cuadro anterior. Por ende, el usuario es el responsable de definir qué etiquetas son asociadas a sus notas de Señales de conocimiento.

3.2.2. PATRÓN DE DISEÑO ONTOLÓGICO

La organización de relaciones de las Ontologías facetadas ligeras requiere la definición de un Patrón de diseño ontológico, que se utiliza para la construcción de Ontologías con la herramienta propuesta. A continuación se describen los elementos básicos.

Figura N° 3.10.**Patrón de diseño ontológico para Ontologías facetadas ligeras**

Fuente: Elaboración propia, 2017.

- Nota (Entidad): Representa las Señales de conocimiento creadas por el usuario, tiene la propiedad Descripción.
- Término taxonómico (Entidad): Representa las etiqueta semánticas obtenidas mediante Open Calais y que son extensibles por el usuario.
- Relación (Entidad): Representa las relaciones creadas por el usuario entre Señales de conocimiento.
- Is a (Relación): Es el tipo de relación existente entre Notas y Término taxonómicos.
- Endpoints (Relación): Es el tipo de relación existente entre dos Notas.

3.3. MÉTODO DE BÚSQUEDA SEMÁNTICA

El método de búsqueda semántica modelado en la herramienta sigue los procesos del sistema QueryGen¹⁷, para la interpretación de palabras clave y construcción de consultas semánticas en la búsqueda de contenidos. Además, se implementan interfaces para la interacción del usuario con los procesos del modelo de gestión de conocimiento personal. A continuación, se describen los procesos que el método de búsqueda realiza para la construcción de consultas semánticas y posteriormente, se presentan las fases de la metodología ICONIX para el desarrollo de las interfaces.

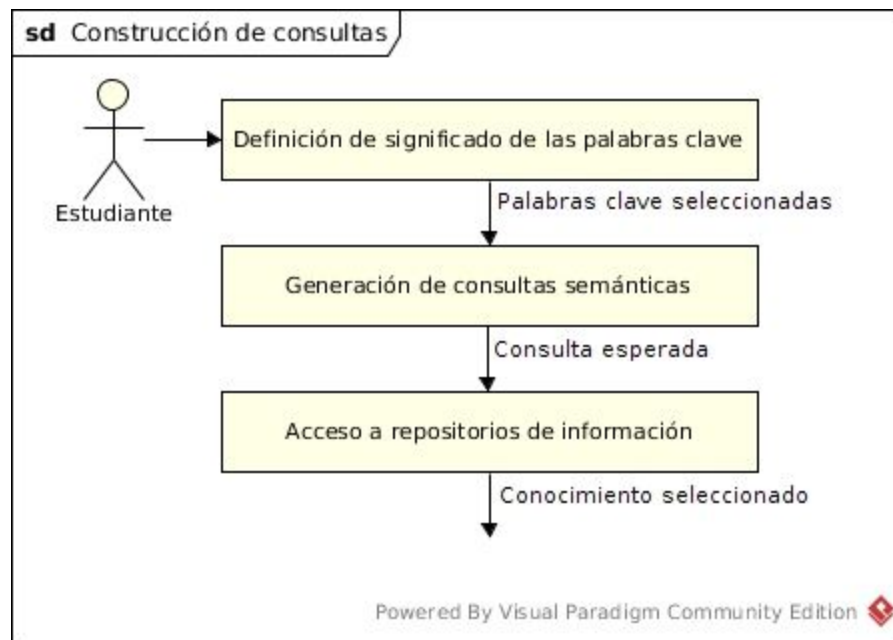
¹⁷ Véase Capítulo 2: QueryGen

3.3.1. CONSTRUCCIÓN DE CONSULTAS SEMÁNTICAS

El método de búsqueda propuesto por QueryGen, se inicia con la definición del significado de las palabras clave ingresadas por el usuario, se prosigue con la construcción de consultas semánticas y se ejecutan las mismas para el acceso a los repositorios de conocimiento. La siguiente figura muestra los procesos del método de construcción de consultas semánticas y las salidas esperadas.

Figura N° 3.11.

Método de construcción de consultas



Fuente: Elaboración propia (Basado en Bobed & Mena), 2016.

A continuación se describe a detalle, los procesos para la construcción de consultas semánticas y su implementación en la herramienta.

3.3.1.1. Definición de significado de palabras clave

Durante el proceso de Definición de significado de palabras clave, se siguen tres subprocesos que se describen a continuación.

- Extracción del significado de palabras clave: Este subproceso representa la separación de

las palabras ingresadas por el usuario, a fin de comenzar el proceso de construcción de consultas semánticas. Se realiza mediante la implementación de una función en el lenguaje JavaScript que se muestra a continuación.

Figura N° 3.12.

Función de definición de significado

```
Drupal.neo4jVisualizer.splitUserInput = function (input) {
    var keywords = input.split(',');
    var index;
    for (index = 0; index < keywords.length; ++index) {
        keywords[index] = keywords[index].trim();
    }
    return keywords;
};
```

Fuente: Elaboración propia, 2017.

- Enriquecimiento del sentido de palabras clave y eliminación de redundancia: Durante este subproceso se define la forma de búsqueda de los términos ingresados por el usuario, a fin de crear consultas que eliminen la distinción de mayúsculas y minúsculas. A continuación se muestra la búsqueda a realizar, en el lenguaje Cypher.

(toLower(n.title) CONTAINS toLower("EjemploPalabraClave"))

Donde el método *toLower*, en el lenguaje Cypher, se utiliza para convertir una cadena de caracteres a minúsculas. De esa manera, si se busca el término “Java”, la consulta devolverá los resultados que incluyan los términos “java”, “Java” y “JAVA”, además de otras variantes.

- Desambiguación de los sentidos de palabras clave: En este subproceso se incluyen condiciones que seleccionen las posibles facetas existentes relacionadas con un nodo, la estructura de las mismas se presenta a continuación.

Cuadro N° 3.8.

Selección de facetas asociadas a un término

| |
|---|
| nodoInicial.vocabulario = "social_tags" nodoInicial.vocabulario = "topic_tags" nodoInicial.vocabulario = "industry_tags" nodoInicial.vocabulario = "markup_tags" |
| Donde: <ul style="list-style-type: none"> - <i>nodoInicial</i>: Nodo inicial de la faceta asociada a un nodo. - <i>nodoInicial.vocabulario = "[nombreDeFaceta]"</i>: Define la selección de los nodos iniciales para la selección de facetas. |

Fuente: Elaboración propia, 2017.

Este proceso tiene como resultado la desambiguación de los términos ingresados por el usuario para su inclusión en la construcción de consultas semánticas.

3.3.1.2. Generación de consultas semánticas

El Proceso de generación de consultas de QueryGen se encarga de construir las consultas semánticas a partir de los términos definidos en el proceso anterior, sigue los siguientes subprocesos:

- Constructor con tablas de análisis: Debido al uso de un único lenguaje (Cypher) de consulta para la implementación del presente proceso, no se incluyeron las tablas de análisis.
- Generación de consultas y Procesado semántico: Ambos subprocesos se encargan de construir consultas con los elementos necesarios para su ejecución. El subproceso de Procesado semántico, busca enriquecer semánticamente el significado del contenido recuperado. De esa forma, la consulta generada, extiende su alcance para incluir los términos taxonómicos directamente relacionados a los nodos recuperados. En consecuencia, las consultas generadas con la herramienta, cuentan con los elementos que se describen a continuación.

Cuadro N° 3.9.
Elementos de consultas generadas

| Elemento | Variables |
|--|--|
| Nombre: Selección de facetas facetas = (nodoInicial)-[*2]-(nodo:`entity:node`) | <ul style="list-style-type: none"> - <i>nodo</i>: Nodo enlazado que representa señales de conocimiento. - <i>nodoInicial</i>: Inicio de faceta. |
| Nombre: Condición facetada nodoInicial.vid = '[nombreFaceta]' | <ul style="list-style-type: none"> - <i>vid</i>: Faceta de nodo. - <i>[nombreFaceta]</i>: Nombre definidos por Open Calais (entidades, términos sociales, temáticos, industriales). |
| Nombre: Condición de búsqueda de nodos toLower(nodo.title) CONTAINS toLower("[palabraClave]") | <ul style="list-style-type: none"> - <i>title</i>: Propiedad que incluye término clave. - <i>[palabraClave]</i>: Ingresado por el usuario. |
| Nombre: Selección de nodos aislados (nodoAislado:`entity:node`) | <ul style="list-style-type: none"> - <i>nodoAislado</i>: Nodo aislado que representa señales de conocimiento. |
| Nombre: Condición de búsqueda en nodos aislados toLower(nodoAislado.title) CONTAINS toLower("[palabraClave]") | <ul style="list-style-type: none"> - <i>title</i>: Propiedad que incluye término clave. - <i>[palabraClave]</i>: Ingresado por el usuario. |
| Nombre: Selección de etiquetas en nodos aislados (nodoAislado)-[relacionAislada]->(etiquetas:`entity:taxonomy_term`) | <ul style="list-style-type: none"> - <i>nodoAislado</i>: Nodo aislado que representa señales de conocimiento. - <i>relacionAislada</i>: Variable de relación de nodos aislados y etiquetas. - <i>etiquetas: `entity:taxonomy_term`</i>: Selección de etiquetas directamente relacionadas a los nodos recuperados. |
| Nombre: Selección de etiquetas en nodos enlazados (nodo)-[relacionFacetada]->(etiquetas:`entity:taxonomy_term`) | <ul style="list-style-type: none"> - <i>nodo</i>: Nodo enlazado que representa señales de conocimiento. - <i>relacionFacetada</i>: Variable de relación de nodos enlazados y etiquetas. - <i>etiquetas: `entity:taxonomy_term`</i>: Selección de etiquetas directamente relacionadas a los nodos recuperados. |

Fuente: Elaboración propia, 2017.

De esa forma, las consultas cuentan con la siguiente estructura:

Cuadro N° 3.10.
Estructura de consultas generadas

| Estructura de consulta |
|--|
| <p>OPTIONAL MATCH [<i>Selección de facetas</i>] WHERE ([<i>Condición facetada</i>] OR ...[<i>Condición facetada</i>]) AND ([<i>Condición de búsqueda de nodos</i>] OR ...[<i>Condición de búsqueda de nodos</i>]) WITH facetas, nodo OPTIONAL MATCH [<i>Selección de nodos aislados</i>] WHERE ([<i>Condición de búsqueda en nodos aislados</i>] OR ...[<i>Condición de búsqueda en nodos aislados</i>]) WITH facetas, nodoAislado, nodo OPTIONAL MATCH [<i>Selección de etiquetas en nodos aislados</i>] OPTIONAL MATCH [<i>Selección de etiquetas en nodos enlazados</i>] RETURN facetas, etiquetas, nodoAislado, relacionAislada, relacionFacetada</p> |

Fuente: Elaboración propia, 2017.

- **Presentación de consultas:** Durante este subproceso se presenta al usuario las consultas generadas durante los subprocesos anteriores. A continuación, se muestra un ejemplo de consulta generada a partir de palabras clave ingresadas por el usuario.

Cuadro N° 3.11.**Ejemplo de generación de consulta**

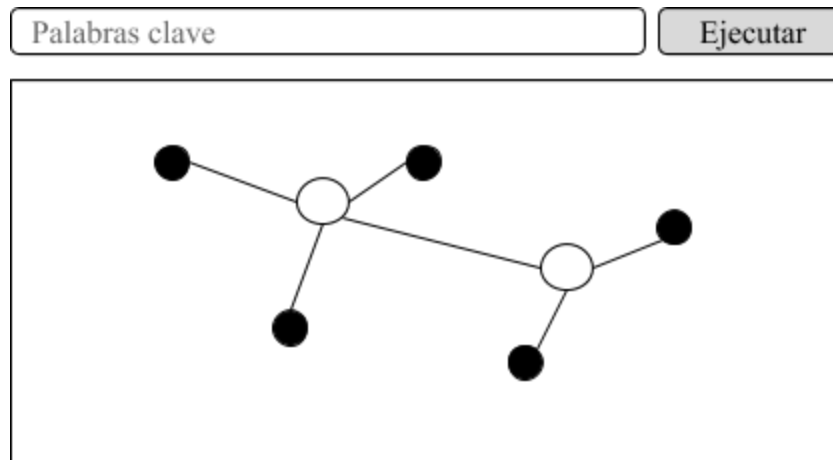
| |
|--|
| Palabras clave ingresadas: IP, redes |
| <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') AND (toLower(node.title) CONTAINS toLower('IP') OR toLower(node.title) CONTAINS toLower('redes')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('IP') OR toLower(singleNode.title) CONTAINS toLower('redes')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

El resultado de este proceso incluye consultas ejecutables con la sintaxis del lenguaje Cypher.

3.3.1.3. Acceso a repositorios de información

Las consultas construidas durante los procesos anteriores se ejecutan sobre las ontologías resultantes de la creación de notas por el usuario, a fin de recuperar y seleccionar el conocimiento almacenado. Los resultados de las consultas ejecutadas se muestran como grafos en la interfaz de usuario. Cada nodo que forma parte de los resultados es clicable y redirecciona a la nota creada en la interfaz Drupal. A continuación, se describen los elementos generales de la interfaz para la ejecución de consultas.

Figura N° 3.13.**Interfaz de selección de conocimiento**

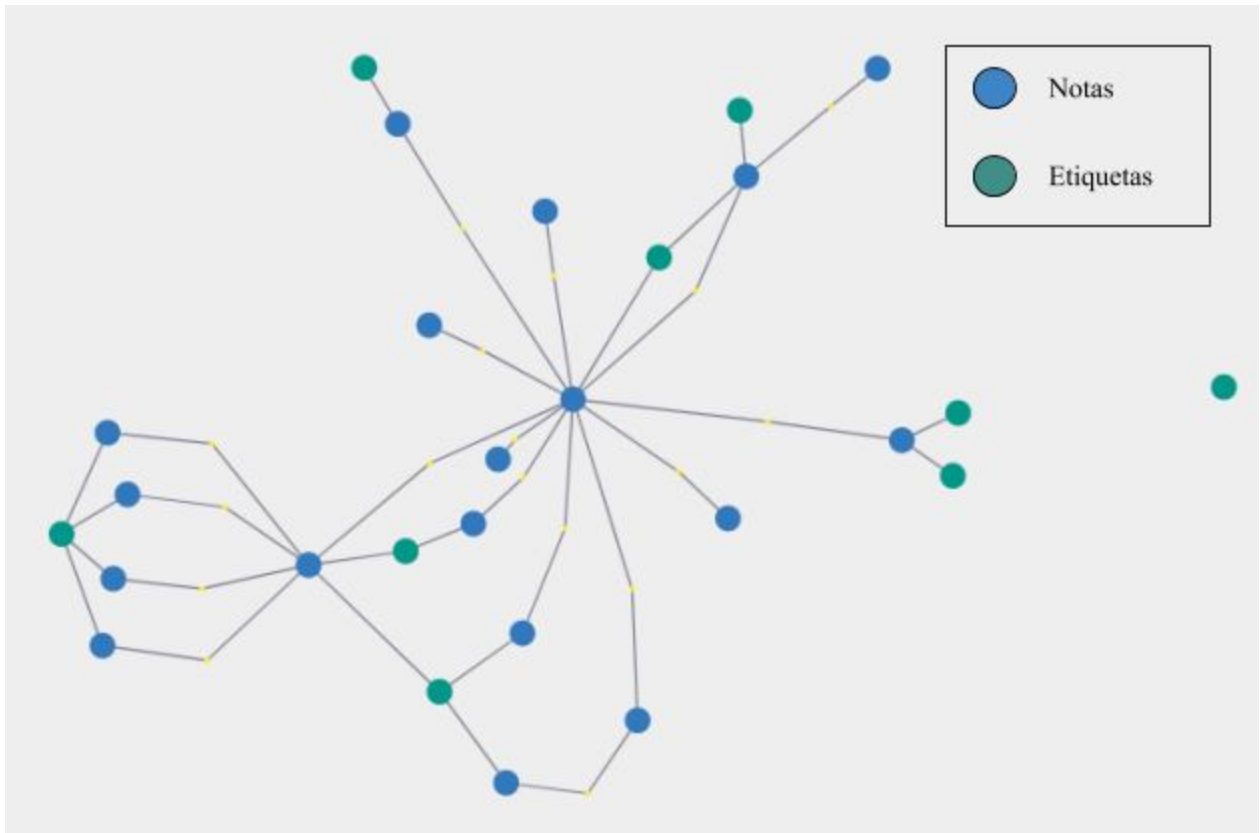
Fuente: Elaboración propia, 2017.

El cuadro de texto para palabras clave conforma la primera interacción del usuario con la vista de selección de conocimiento. El usuario debe ingresar en el cuadro los términos seleccionados, separados por comas.

A continuación, el usuario debe presionar el botón “Ejecutar” para que el proceso de construcción de consultas semánticas se lleve a cabo. Una vez se obtengan resultados, el sistema desplegará los mismos en la interfaz, como grafos de propiedades.

La siguiente figura muestra los resultados de la ejecución de consultas en una base de datos de prueba.

Figura N° 3.14.
Ejemplo de resultados de consultas



Fuente: Elaboración propia, 2017.

3.3.2. IMPLEMENTACIÓN DE INTERFACES

3.3.2.1. Fase de requerimientos

En esta sección se describen los requerimientos funcionales y no funcionales de la herramienta. A continuación se lista los requerimientos funcionales identificados para el soporte de la gestión de conocimiento personal.

| | | | |
|---|---|--------|-------------------------|
| Número | 1 | Título | Administración de notas |
| El sistema permitirá a los usuarios registrados la creación de entidades de tipo Nota con título y descripción. El campo de descripción debe permitir la entrada de archivos e imágenes y contar con el formato html. | | | |

| | | | |
|---|---|--------|---------------------------|
| Número | 2 | Título | Comparación de revisiones |
| El sistema permitirá la comparación de campos entre revisiones de una entidad de tipo Nota. | | | |

| | | | |
|---|---|--------|----------------------|
| Número | 3 | Título | Relación entre notas |
| El sistema permitirá la creación de entidades de tipo Relación entre entidades de tipo Nota mediante la selección de texto de la entidad Nota original. | | | |

| | | | |
|---|---|--------|--|
| Número | 4 | Título | Administración de términos taxonómicos |
| El sistema permitirá el análisis semántico de las entidades de tipo Nota con el fin de sugerir el uso de términos taxonómicos mediante el servicio web Open Calais. Además, permitirá la creación de términos propios administrables. | | | |

| | | | |
|---|---|--------|----------------------------------|
| Número | 5 | Título | Indexación de entidades en Neo4j |
| El sistema indexará en la base de datos Neo4j, las entidades de la herramienta (notas, términos taxonómicos y relaciones) después de la creación de las mismas. | | | |

| | | | |
|---|---|--------|--|
| Número | 6 | Título | Búsqueda mediante <i>Wiki</i> personal |
| El sistema permitirá la búsqueda como <i>Wiki</i> personal de las entidades existentes en la herramienta. | | | |

| | | | |
|--|---|--------|--------------------------|
| Número | 7 | Título | Búsqueda mediante Grafos |
| El sistema permitirá la búsqueda de conocimiento mediante la ejecución de consultas a la base de datos Neo4j. El sistema desplegará los resultados en la herramienta como grafos de propiedades. | | | |

Las características generales de la herramienta se describen a continuación como requerimientos no funcionales.

| | | | |
|--|---|--------|------------------------|
| Número | 1 | Título | Sistema de entorno web |
| El sistema será desarrollado como un sistema de entorno web para permitir su funcionalidad en diferentes plataformas y permitir la portabilidad del contenido ingresado. Además, con el fin de permitir su implementación como una plataforma SaaS (<i>Software as a service</i>) en etapas posteriores a su implementación. | | | |

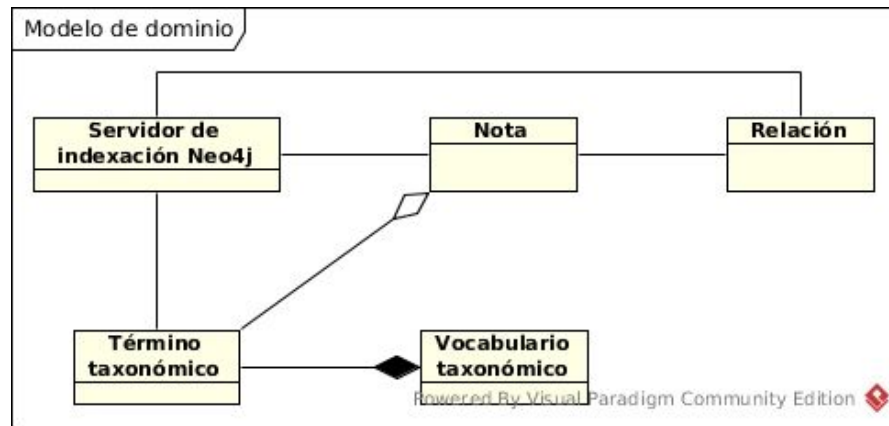
| | | | |
|---|---|--------|---|
| Número | 2 | Título | Actualizaciones / Ruta de actualización |
| El sistema deberá recibir actualizaciones de seguridad y corrección de errores manteniendo la consistencia de la información, mediante una Ruta de actualización. | | | |

| | | | |
|---|---|--------|---------------------|
| Número | 3 | Título | Estructura de rutas |
| El sistema propuesto deberá mantener una estructura simple de rutas basados en los tipos de contenido, vocabularios taxonómicos y relaciones entre entidades. | | | |

| | | | |
|--|---|--------|-----------------------|
| Número | 4 | Título | Configuración inicial |
| El sistema deberá proporcionar la configuración inicial de los siguientes aspectos: <ul style="list-style-type: none"> - Tipos de contenido para las entidades Nota. - Tipos de entidades de relación. - Vocabularios de términos taxonómicos. - Servidor de indexación para la base de datos Neo4j. - Estructura de menús. | | | |

Para evitar la ambigüedad en el entendimiento de la herramienta, ICONIX propone el desarrollo del Modelo de dominio previo a la creación de Casos de uso. Se debe usar el diagrama como un glosario del proyecto y debe limitarse a abstraer los conceptos clave en el dominio de la herramienta. El diagrama resultante, se muestra a continuación.

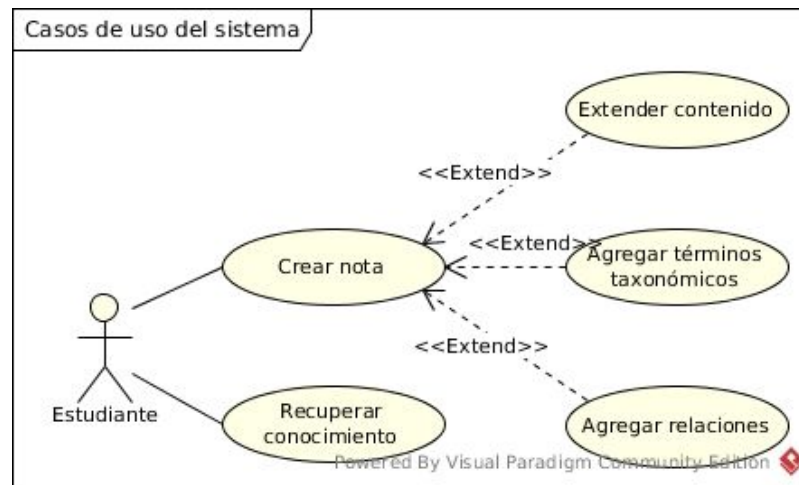
Figura N° 3.15.
Modelo de dominio



Fuente: Elaboración propia, 2017.

Los casos de uso que se desarrollan a continuación siguen la recomendación de la metodología ICONIX para ser creados en base al modelo de dominio. Deben permitir la captura de los requerimientos de comportamiento del sistema y deben ser escritos de manera que se enfoquen en la captura de las acciones del usuario y las respuestas del sistema. A continuación, se muestran los casos de uso para el proceso de gestión de conocimiento personal.

Figura N° 3.16.
Diagrama de casos de uso del sistema



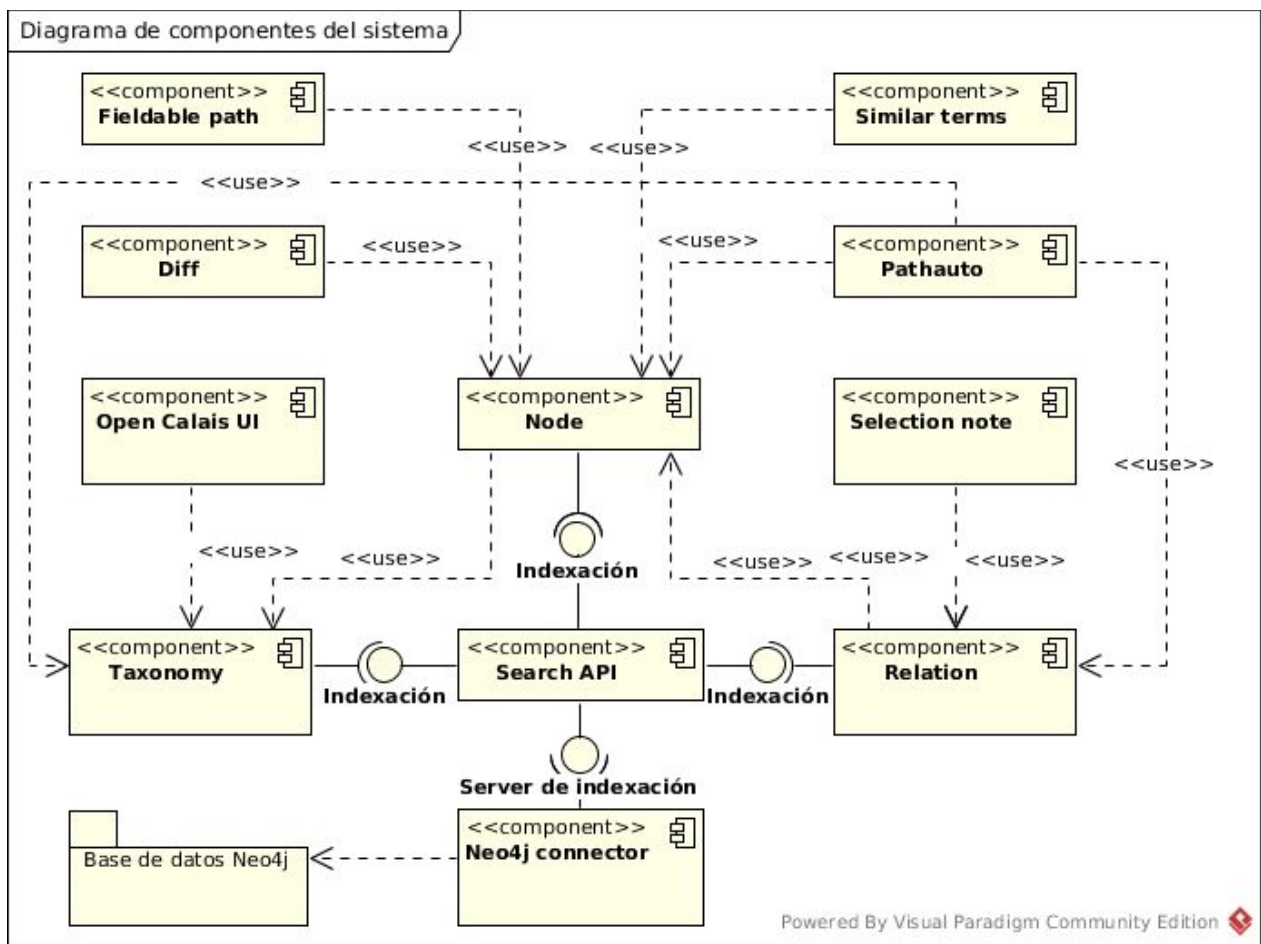
Fuente: Elaboración propia, 2017.

3.3.2.2. Fase de análisis y diseño preliminar

Para la fase de análisis y diseño preliminar, ICONIX sugiere el uso de diagramas de robustez para la posterior especificación de los casos de uso. Sin embargo, debido al uso de módulos de Drupal en la herramienta del presente proyecto, se sustituye el diagrama de robustez por el diagrama de componentes. A continuación se muestra el diagrama resultante que refleja la interacción entre módulos de Drupal.

Figura N° 3.17.

Diagrama de componentes del sistema



Fuente: Elaboración propia, 2017.

A continuación se realiza la especificación de los casos de uso definidos en el punto anterior.

Cuadro N° 3.12.
Caso de uso: Crear Nota

| | |
|--|-------------|
| Nombre | Crear Nota |
| Autor | John Choque |
| Fecha | 27-09-2017 |
| Descripción: Este caso de uso describe las acciones que el usuario Estudiante realiza para la creación de Notas en la herramienta. | |
| Actores: Estudiante | |
| Precondiciones: El usuario Estudiante debe estar registrado en el sistema y haber ingresado al mismo. | |
| Flujo normal: Este caso de uso comienza cuando el usuario Estudiante desea crear una entidad de tipo Nota: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de bienvenida de la herramienta y presiona el menú de “Señales de conocimiento” que muestra un submenú desplegable con la opción “Crear”. 2. El usuario presiona la opción “Crear” que despliega la formulario para agregar una nota. 3. El usuario ingresa el contenido en los campos del formulario y presiona “Guardar” para que el sistema guarde los datos. 4. El sistema despliega un mensaje de “La nota [título de la nota] ha sido creada”. | |
| Flujo alternativo: Si el usuario Estudiante quiere crear nota a partir de la selección de un texto en una nota existente: <ol style="list-style-type: none"> 1. Se siguen los pasos del flujo normal del caso de uso: Agregar relaciones. | |
| Pos-condiciones: Si el flujo normal del caso de uso fue exitoso, el estudiante ha creado una Nota. | |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.13.**Caso de uso: Extender contenido**

| | |
|--|--------------------|
| Nombre | Extender contenido |
| Autor | John Choque |
| Fecha | 27-09-2017 |
| Descripción: Este caso de uso describe las acciones que el usuario Estudiante realiza para la extensión de las Notas creadas. | |
| Actores: Estudiante | |
| Precondiciones: <ul style="list-style-type: none"> - El usuario Estudiante debe estar registrado en el sistema y haber ingresado al mismo. - Se debe haber seguido el proceso del caso de uso: Crear Nota. | |
| Flujo normal: Comienza cuando el usuario Estudiante desea extender el contenido textual de una Nota: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de bienvenida de la herramienta y presiona el menú de “Administrar” que muestra un submenú desplegable con la opción “Señales de conocimiento”. 2. El usuario presiona la opción “Señales de conocimiento” que despliega una vista con todas las señales de conocimiento listados por fecha. 3. El usuario selecciona la Nota que desea extender, presiona el botón “Editar” y el sistema despliega el formulario de edición. 4. El usuario ingresa el contenido en los campos del formulario y presiona “Guardar” para que el sistema guarde los datos. 5. El sistema despliega un mensaje de “La nota [título de la nota] ha sido actualizada”. | |
| Flujo alternativo: Si el usuario Estudiante requiere agregar imágenes o archivos a las Notas: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de edición de una nota y desea agregar un archivo al contenido de la misma. 2. El usuario selecciona el ícono de imagen del campo, el sistema despliega un campo extra para seleccionar la imagen. 3. El usuario selecciona la imagen, ingresa un texto alternativo y presiona “Guardar”. 4. El sistema muestra la imagen agregada en el campo de descripción de la nota. | |
| Pos-condiciones: Si el flujo normal del caso de uso fue exitoso, el estudiante ha extendido una Nota. | |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.14.**Caso de uso: Asignar términos taxonómicos**

| | |
|--|------------------------------|
| Nombre | Asignar términos taxonómicos |
| Autor | John Choque |
| Fecha | 27-09-2017 |
| Descripción: Este caso de uso describe las acciones que el usuario Estudiante realiza para asignar Términos taxonómicos a las Notas. | |
| Actores: Estudiante | |
| Precondiciones: <ul style="list-style-type: none"> - El usuario Estudiante debe estar registrado en el sistema y haber ingresado al mismo. - Se debe haber seguido el flujo del caso de uso: Crear Nota. | |
| Flujo normal: Este caso de uso comienza cuando el usuario Estudiante desea asignar Términos taxonómicos a una Nota: <ol style="list-style-type: none"> 1. El usuario se encuentra en la vista de una Nota previamente creada y selecciona la pestaña “Asignar etiquetas”. 2. El sistema despliega el formulario para asignar términos taxonómicos y el usuario presiona el botón “Sugerir etiquetas”. 3. El sistema analiza el texto y despliega las etiquetas posibles para ser asignadas. 4. El usuario selecciona las etiquetas que desea asignar a la Nota analizada y presiona el botón “Guardar” para que el sistema asigne los términos taxonómicos. 5. El sistema despliega un mensaje de “Todos los cambios han sido guardados.”. | |
| Flujo alternativo: Si el usuario Estudiante busca agregar Términos taxonómicos personalizados: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de edición de una Nota. 2. El usuario ingresa el Término taxonómico requerido en el campo “Etiquetas” y presiona el botón guardar. 3. El sistema crea el Término taxonómico y lo asigna a la Nota. 4. El sistema despliega un mensaje de “Todos los cambios han sido guardados.”. Si el usuario Estudiante busca agregar Términos taxonómicos sin asignarlos a Notas: <ol style="list-style-type: none"> 1. El usuario presiona el menú de “Administrar” que muestra un submenú desplegable con la opción “Etiquetas” que es seleccionada por el usuario. 2. El sistema despliega la lista de vocabularios taxonómicos registrados en la herramienta y el usuario presiona el botón “Listar términos”. | |

3. El sistema despliega la lista de términos del vocabulario seleccionado y el usuario presiona el botón “Agregar término” que despliega el formulario de creación de término.
4. El usuario ingresa todos los datos y presiona “Guardar”.
5. El sistema despliega el mensaje de “Nuevo término [Nombre de término] creado.”.

Pos-condiciones:

Si el flujo normal del caso de uso fue exitoso, el estudiante ha asignado términos taxonómicos a una Nota.

Fuente: Elaboración propia, 2017.

Cuadro N° 3.15.**Caso de uso: Agregar relaciones**

| | |
|---|--------------------|
| Nombre | Agregar relaciones |
| Autor | John Choque |
| Fecha | 27-09-2017 |
| Descripción: Este caso de uso describe las acciones que el usuario Estudiante realiza para la creación de relaciones entre Notas. | |
| Actores: Estudiante | |
| Precondiciones: <ul style="list-style-type: none"> - El usuario Estudiante debe estar registrado en el sistema y haber ingresado al mismo. - Se debe haber seguido el flujo del caso de uso: Crear Nota. | |
| Flujo normal: Este caso de uso comienza cuando el usuario Estudiante desea agregar relaciones entre Notas: <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de vista de una Nota y selecciona texto del campo Descripción de la misma, el sistema despliega el botón “Agregar nota”. 2. El usuario presiona el botón “Agregar nota” y el sistema despliega el formulario para agregar nota. 3. El usuario ingresa los datos y presiona “Guardar”, el sistema crea la Nota y una entidad de tipo Relación entre ambas entidades. 4. El sistema despliega un mensaje de “La nota [título de la nota] ha sido creada”. | |
| Flujo alternativo: Si el usuario estudiante quiere agregar relaciones que no estén basadas en la selección de texto: <ol style="list-style-type: none"> 1. El usuario presiona el menú de “Administrar” que muestra un submenú desplegable con la opción “Relaciones” que es seleccionada por el usuario. 2. El sistema despliega el formulario de creación de Relaciones y el usuario ingresa los datos requeridos. 3. El usuario presiona el botón “Guardar” y el sistema despliega un mensaje de “La relación ha sido creada”. | |
| Pos-condiciones: Si el flujo normal del caso de uso fue exitoso, el estudiante ha agregado relaciones entre Notas. | |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.16.**Caso de uso: Recuperar conocimiento**

| | |
|---|------------------------|
| Nombre | Recuperar conocimiento |
| Autor | John Choque |
| Fecha | 27-09-2017 |
| Descripción: Este caso de uso describe las acciones que el usuario Estudiante realiza para la recuperación de conocimiento. | |
| Actores: Estudiante | |
| Precondiciones: <ul style="list-style-type: none"> - El usuario Estudiante debe estar registrado en el sistema y haber ingresado al mismo. - Se deben haber seguido los flujos de los casos de uso: Crear Nota. | |
| Flujo normal: Este caso de uso comienza cuando el usuario estudiante desea realizar la Recuperación de conocimiento. <ol style="list-style-type: none"> 1. El usuario se encuentra en la pantalla de bienvenida de la herramienta y presiona el menú de “Búsqueda” que muestra un submenú desplegable con la opción “Grafo”. 2. El usuario presiona la opción “Grafo” y el sistema despliega una vista que renderiza todos los nodos y relaciones almacenados en Neo4j. 3. El usuario escribe la consulta en el lenguaje cypher en el cuadro de texto en la pantalla y la herramienta realiza la búsqueda en el grafo de conocimiento. 4. El sistema despliega los resultados como subgrafos y el usuario puede acceder al contenido de cada nodo con un click doble. | |
| Flujo alternativo: Ninguno | |
| Pos-condiciones: Si el flujo normal del caso de uso fue exitoso, el estudiante ha seleccionado y recuperado conocimiento. | |

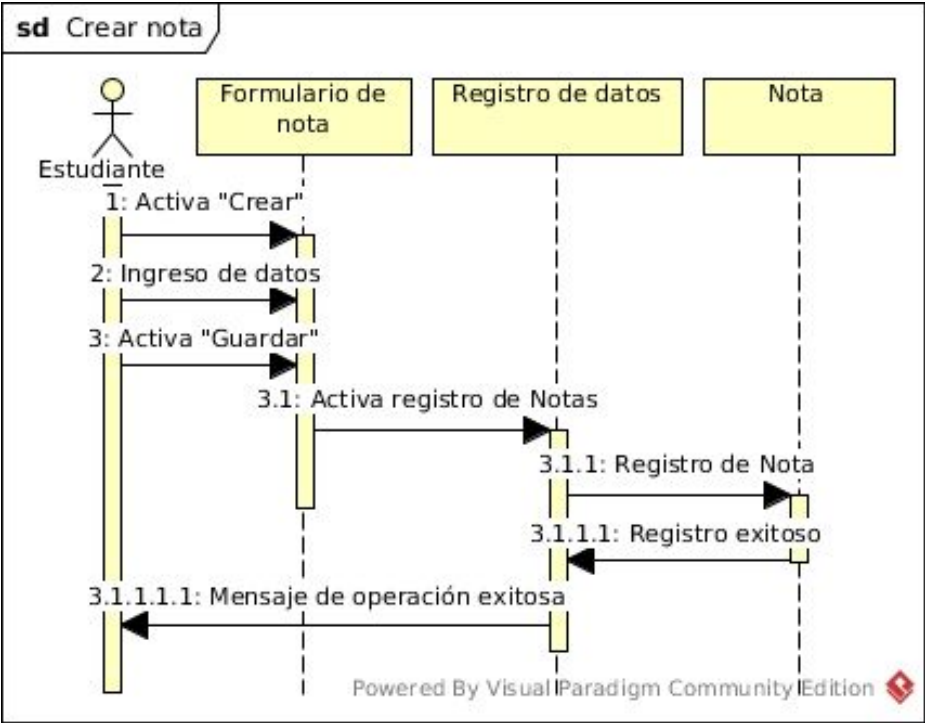
Fuente: Elaboración propia, 2017.

3.3.2.3. Fase de diseño detallado

La metodología ICONIX hace uso de los diagramas de secuencia como herramienta para la exploración del diseño detallado del sistema. A continuación se muestran los diagrama de

secuencias para los diferentes casos de uso especificados en el punto anterior.

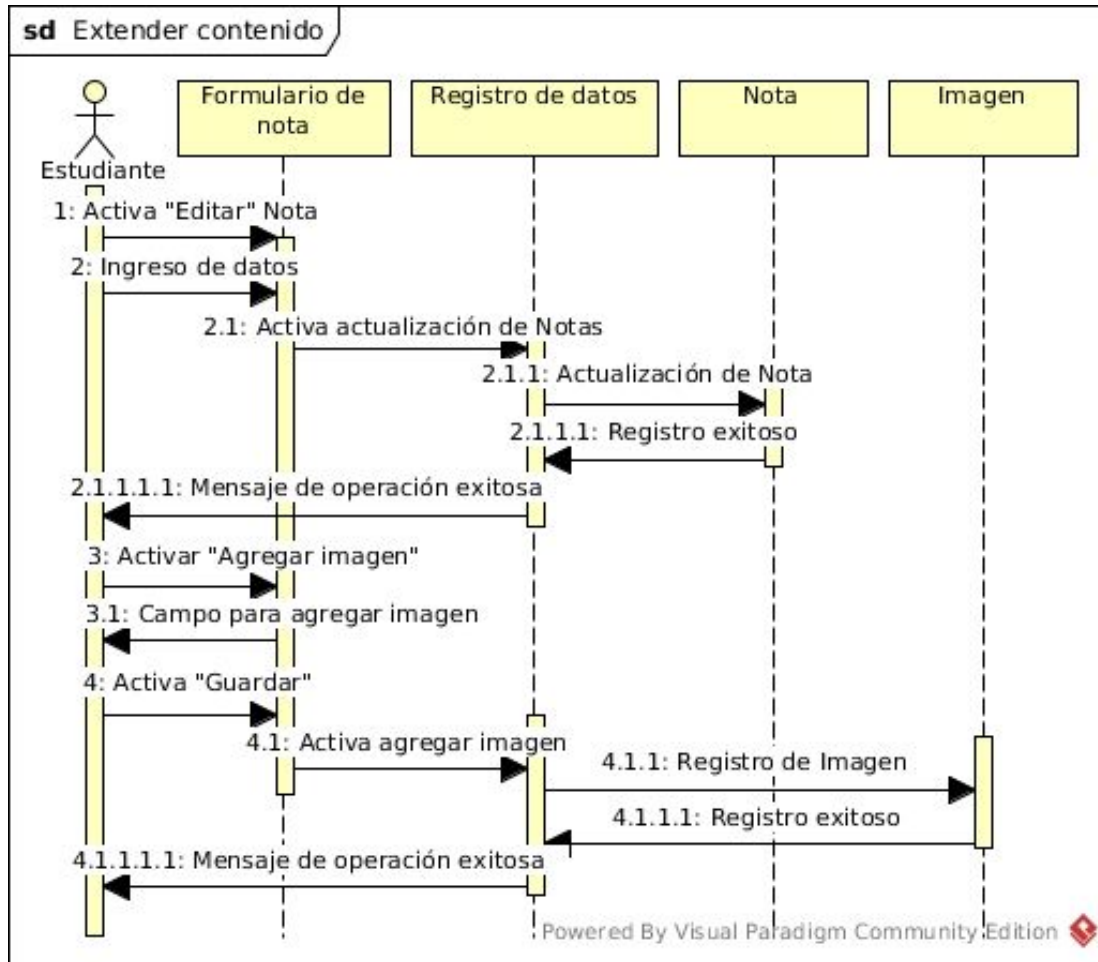
Figura N° 3.18.
Diagrama de secuencia: Crear Nota



Fuente: Elaboración propia, 2017.

Figura N° 3.19.

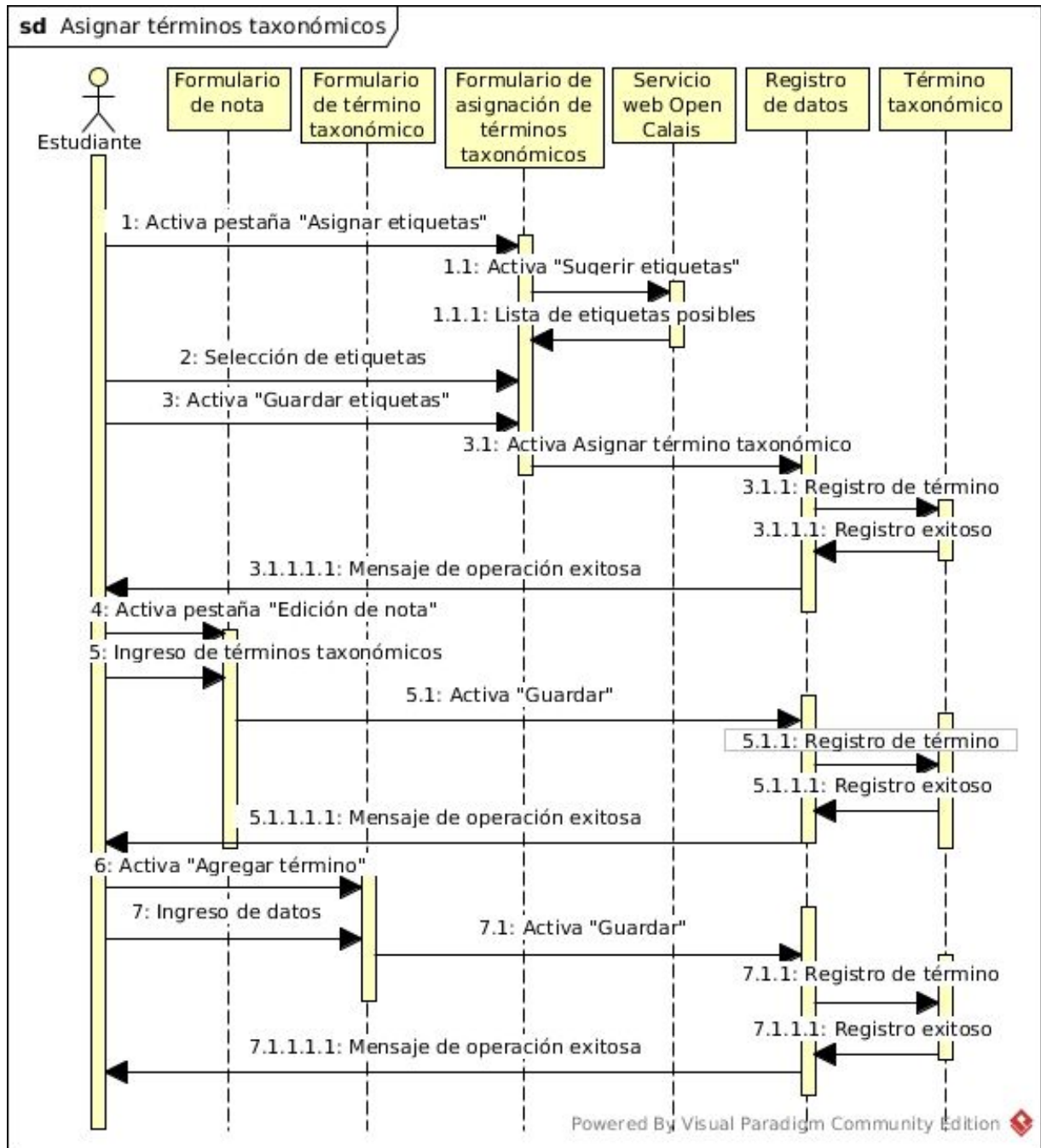
Diagrama de secuencia: Extender contenido



Fuente: Elaboración propia, 2017.

Figura N° 3.20.

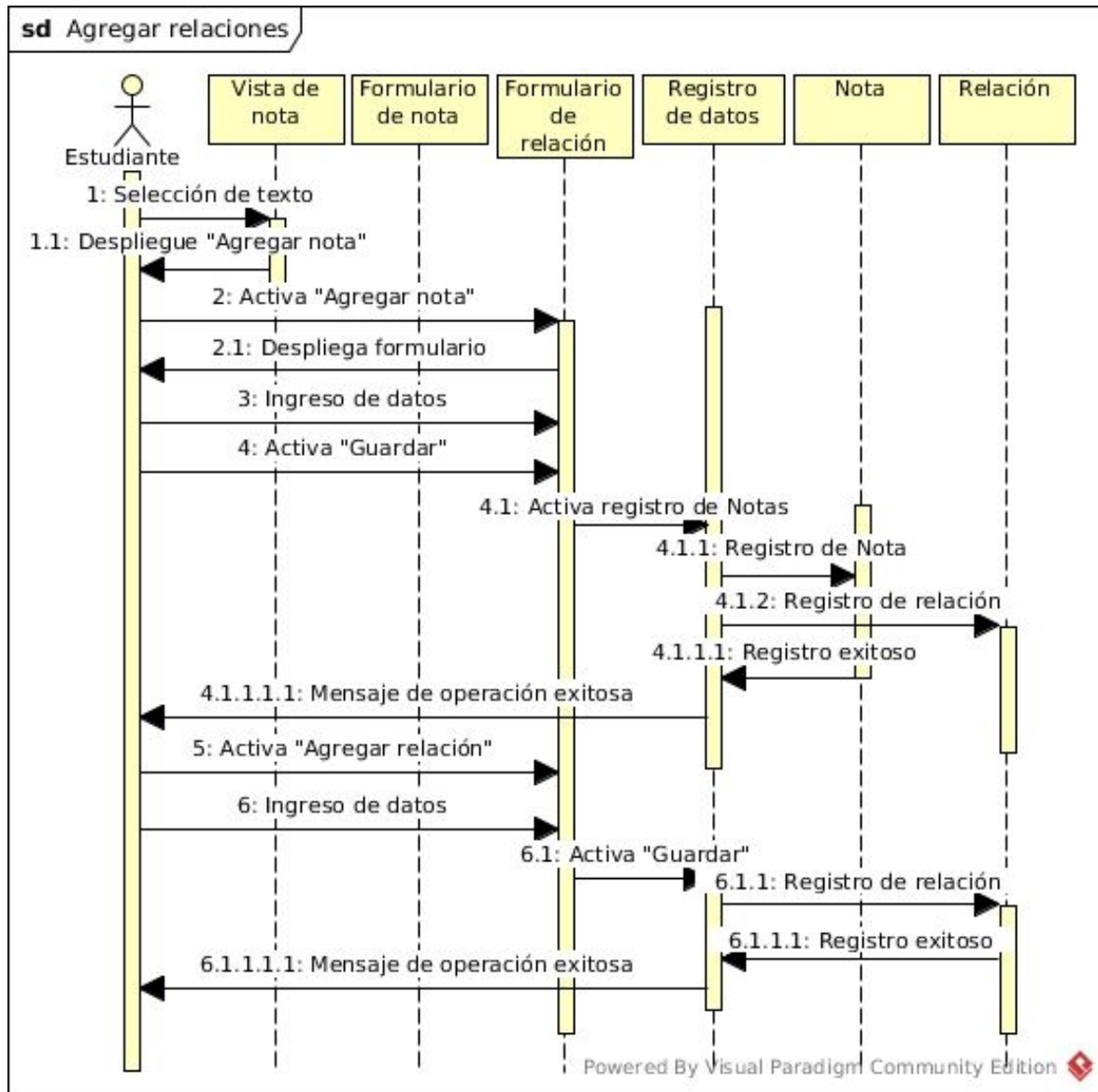
Diagrama de secuencia: Asignar términos taxonómicos.



Fuente: Elaboración propia, 2017.

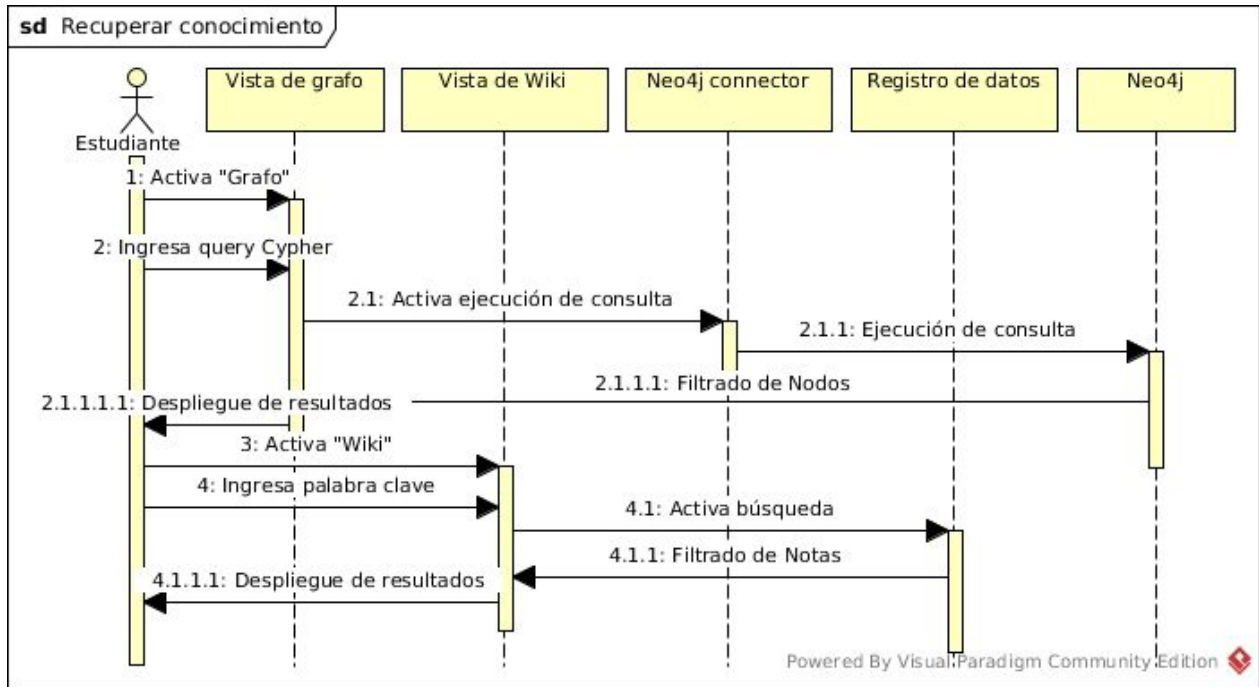
Figura N° 3.21.

Diagrama de secuencia: Agregar relaciones



Fuente: Elaboración propia, 2017.

Figura N° 3.22.

Diagrama de secuencia: Recuperar conocimiento

Fuente: Elaboración propia, 2017.

3.3.2.4. Fase de implementación

Durante la fase de implementación de la metodología ICONIX se procede con la codificación de la herramienta. La misma, se implementa como una distribución de Drupal que incluye módulos de la comunidad, considerados como cajas negras y extienden su funcionalidad mediante el aporte con código para obtener la mínima funcionalidad requerida. El desarrollo de la herramienta se concentra en la construcción del perfil de instalación, el manejo de dependencias y el modelado de las interfaces gráficas.

El perfil de instalación define la configuración inicial al instalarse la herramienta y consta de los siguientes elementos:

- Archivo *info*: Es un archivo de formato yml que define y configura el perfil de instalación, incluye la definición de los módulos que serán habilitados durante la instalación de la herramienta, además, del tema por defecto que usará la interfaz. El apéndice 3 (cf. Apéndice 3) incluye el contenido del archivo *info* del perfil de instalación

de la herramienta.

- Archivo *install*: El archivo *install* contiene una secuencia de comandos que se ejecutan al instalarse la herramienta. El contenido del archivo se muestra a continuación.

Figura N° 3.23.

Archivo install del perfil de instalación

```
<?php

/**
 * @file
 * Install, update and uninstall functions for the Student install profile.
 */

use Drupal\user\Entity\User;

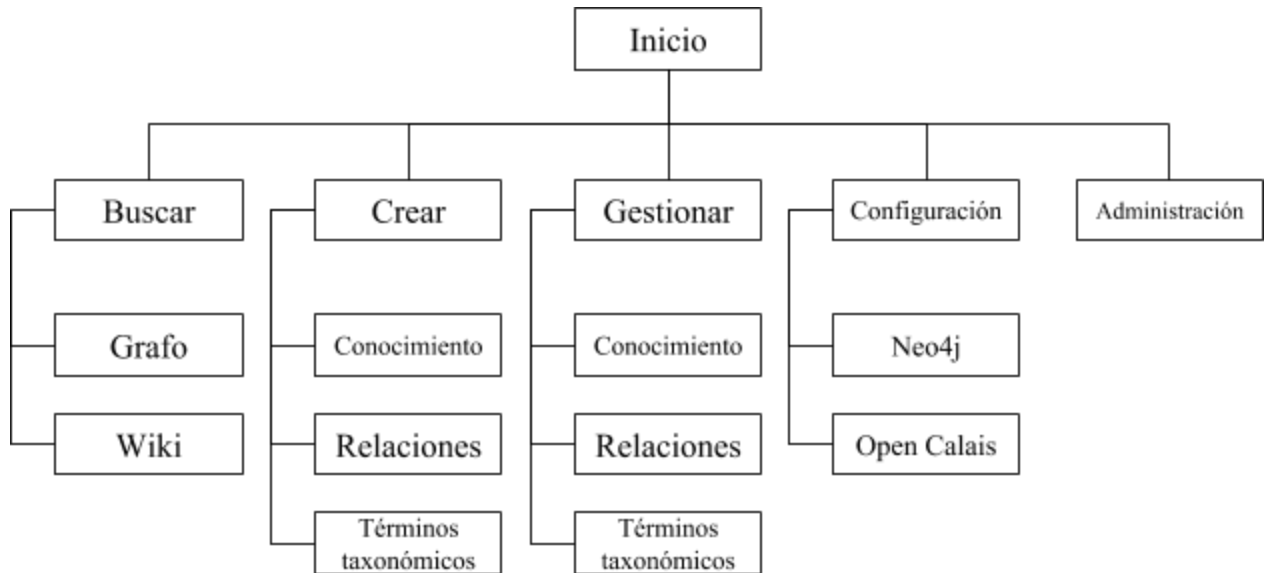
/**
 * Implements hook_install().
 *
 * Perform actions to set up the site for this profile.
 *
 * @see system_install()
 */
function student_install() {
  // Assign user 1 the "administrator" role.
  $user = User::load(1);
  $user->roles[] = 'administrator';
  $user->save();
}
```

Fuente: Elaboración propia, 2017.

- Archivo *links.menu*: Contiene la definición de los enlaces que forman parte del menú de navegación. Incluyen submenús que son habilitados durante la instalación del sistema. El apéndice 4 (cf. Apéndice 4) incluye el contenido del archivo *links.menu* del perfil de instalación de la herramienta. Además, a continuación se muestra el mapa de navegación de la herramienta.

Figura N° 3.24.

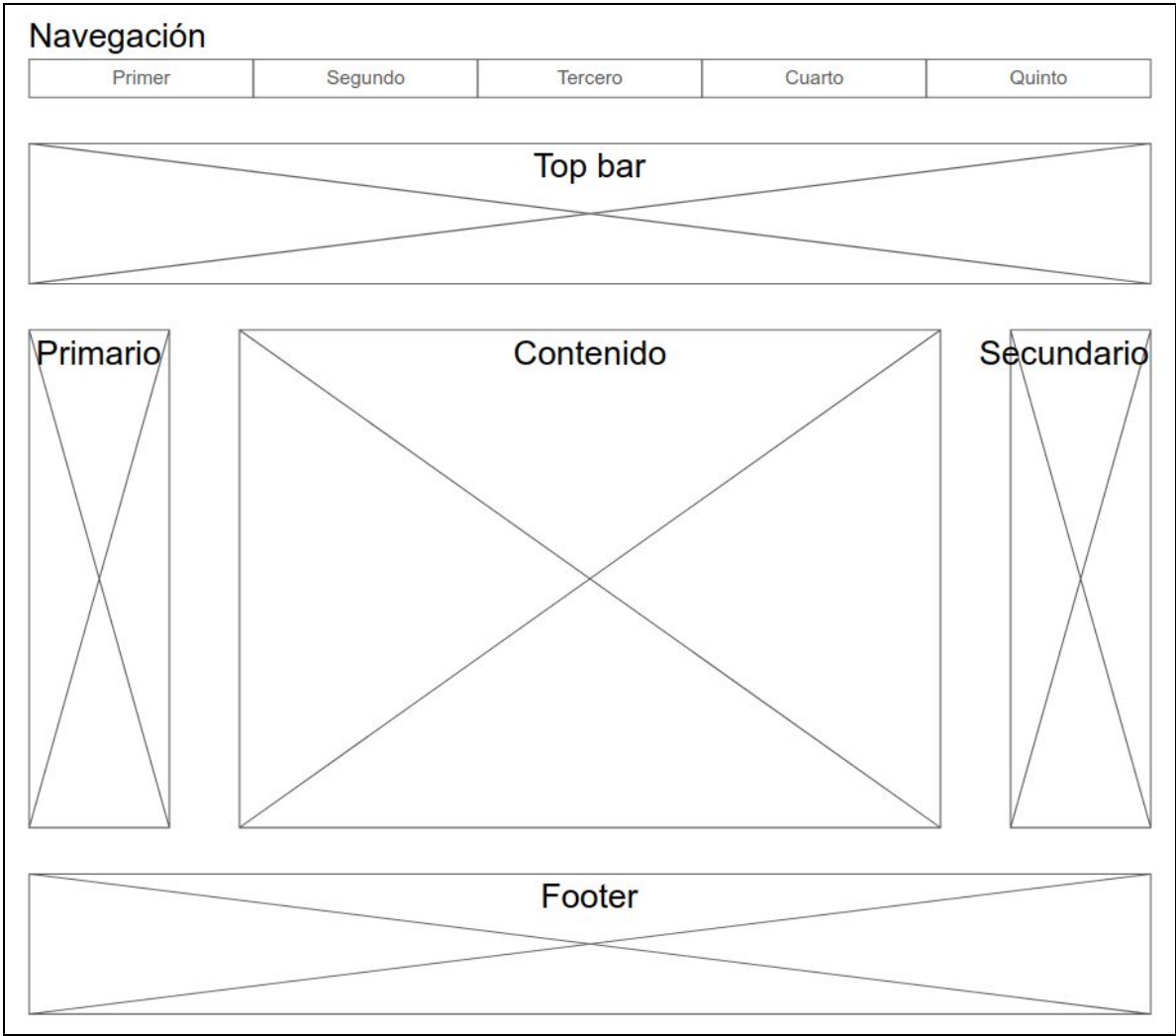
Mapa de navegación de la herramienta



Fuente: Elaboración propia, 2017.

- Directorio *install*: El directorio *install* contiene los archivos de configuración inicial que se utilizan al instalarse la herramienta, define la configuración de bloques, temas, vistas de formulario, vistas de entidades, tipos de contenido, campos de entidades y vocabularios taxonómicos. A continuación se describen los elementos del directorio *install*.
 - Configuración de bloques: Permiten el modelado de las interfaces gráficas y son dependientes del tema usado en la herramienta. A continuación se muestra la estructura de bloques del sistema.

Figura N° 3.25.
Estructura de bloques del sistema



Fuente: Elaboración propia, 2017.

- Tipos de contenido y campos: Las notas de Señales de conocimiento serán almacenadas en entidades de tipo Nota con los campos Título, Descripción y Etiquetas.

Figura N° 3.26.
Tipo de contenido: Nota

Formulario de creación de una Nota:

- Campo de texto: Título
- Campo de texto: Descripción
- Campo de texto: Etiquetas
- Botón: Agregar más
- Botón: Guardar

Fuente: Elaboración propia, 2017.

- Servidores de indexación para Neo4j: Todo contenido ingresado en la herramienta debe ser transferido a Neo4j durante el proceso de guardado. El archivo del perfil de instalación que constituye el servidor de indexación, se encuentra en el Apéndice 5 (cf. Apéndice 5).

El manejo de dependencias de la herramienta se realiza mediante la herramienta Composer. El apéndice 6 (cf. Apéndice 6) incluye los contenidos del archivo composer.json de la distribución de Drupal.

El siguiente proceso dentro de la fase de Implementación de la metodología ICONIX implica la implementación de un plan de pruebas del sistema. A continuación se muestra el plan de pruebas resultante.

Cuadro N° 3.17.**Plan de pruebas**

| Código | Detalle de prueba | Fecha de ejecución | Pasó / Falló |
|---------------|------------------------------|---------------------------|---------------------|
| P001 | Crear nota | 06-11-2017 | Pasó |
| P002 | Extender contenido | 06-11-2017 | Pasó |
| P003 | Asignar términos taxonómicos | 06-11-2017 | Pasó |
| P004 | Agregar relaciones | 06-11-2017 | Pasó |
| P005 | Recuperar conocimiento | 06-11-2017 | Pasó |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.18.**Prueba de aceptación: Crear nota**

| Título: Crear nota | | |
|---|--|---|
| Resumen: Creación de una entidad Nota. | | |
| Criterio de aceptación: Entidad nota creada. | | |
| Test | Nombre del test | Resultado esperado |
| 1. | Accede a “Crear” en el submenú de “Conocimiento” | El sistema debe desplegar el formulario de creación de Nota |
| 2. | Ingreso de datos requeridos | Validación de campos requeridos |
| 3. | Presiona “Guardar” | Validar los datos enviados |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.19.**Prueba de aceptación: Extender contenido**

| Título: Extender contenido | | |
|---|-----------------------------|---|
| Resumen: Creación de revisiones de Notas. | | |
| Criterio de aceptación: Revisión de nota existente creada. | | |
| Test | Nombre del test | Resultado esperado |
| 1. | Accede a la edición de nota | El sistema despliega el formulario de edición de Nota |
| 2. | Ingreso de datos requeridos | Validación de campos requeridos |
| 3. | Presiona “Guardar” | Creación de revisión |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.20.**Prueba de aceptación: Asignar términos taxonómicos**

| Título: Asignar términos taxonómicos | | |
|--|---|--|
| Resumen: Asignación de términos taxonómicos de Open Calais a Notas. | | |
| Criterio de aceptación: Término taxonómico asignado y referenciado. | | |
| Test | Nombre del test | Resultado esperado |
| 1. | Accede a formulario de asignación de términos taxonómicos con Open Calais | Sistema despliega formulario de sugerencia de términos taxonómicos |
| 2. | Presiona “Sugerir términos” | Envío de información al servicio web Open Calais, análisis y despliegue de sugerencias |
| 3. | Selecciona términos para agregar y presiona “Guardar” | Envío de datos a registro y referencia de término taxonómico |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.21.**Prueba de aceptación: Agregar relaciones**

| Título: Agregar relaciones | | |
|---|-------------------------|---|
| Resumen: Agregar relaciones entre dos Notas. | | |
| Criterio de aceptación: Relación creada entre dos notas. | | |
| Test | Nombre del test | Resultado esperado |
| 1. | Selección de texto | Despliegue de botón “Agregar nota” |
| 2. | Presiona “Agregar nota” | Despliegue de formulario de creación de Nota. |
| 3. | Ingreso de datos | Validación de campos requeridos |
| 4. | Presiona “Guardar” | Relación creada. |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.22.**Prueba de aceptación: Recuperar conocimiento**

| Título: Recuperar conocimiento | | |
|--|--|---|
| Resumen: Recuperación de conocimiento con consultas semánticas. | | |
| Criterio de aceptación: Grafo filtrado desplegado. | | |
| Test | Nombre del test | Resultado esperado |
| 1. | Accede a “Grafo” en el submenú de “Buscar” | Despliegue de formulario de ejecución de consultas Cypher |
| 2. | Ingresas query y presiona “Buscar” | Envío de consulta, ejecución y despliegue de datos. |

Fuente: Elaboración propia, 2017.

3.4. EFICACIA DE LA RECUPERACIÓN DE CONOCIMIENTO PERSONAL

Para la medición de la eficacia de la recuperación de conocimiento personal, se desarrollaron ontologías de prueba en base a Proyectos de Grado aprobados de la Universidad Privada del

Valle de la carrera de Ingeniería de Sistemas. Se crearon quince¹⁸ ontologías que siguieron el proceso de desarrollo que se describe a continuación.

3.4.1. DESARROLLO DE ONTOLOGÍAS DE PRUEBA

Para el desarrollo de las ontologías de prueba se hizo uso de la metodología NeOn mediante la selección de los escenarios que la misma ofrece. Los escenarios seleccionados fueron: Escenario 1: “De la especificación a la implementación” y Escenario 7: “Reutilización de patrones de diseño ontológicos”.

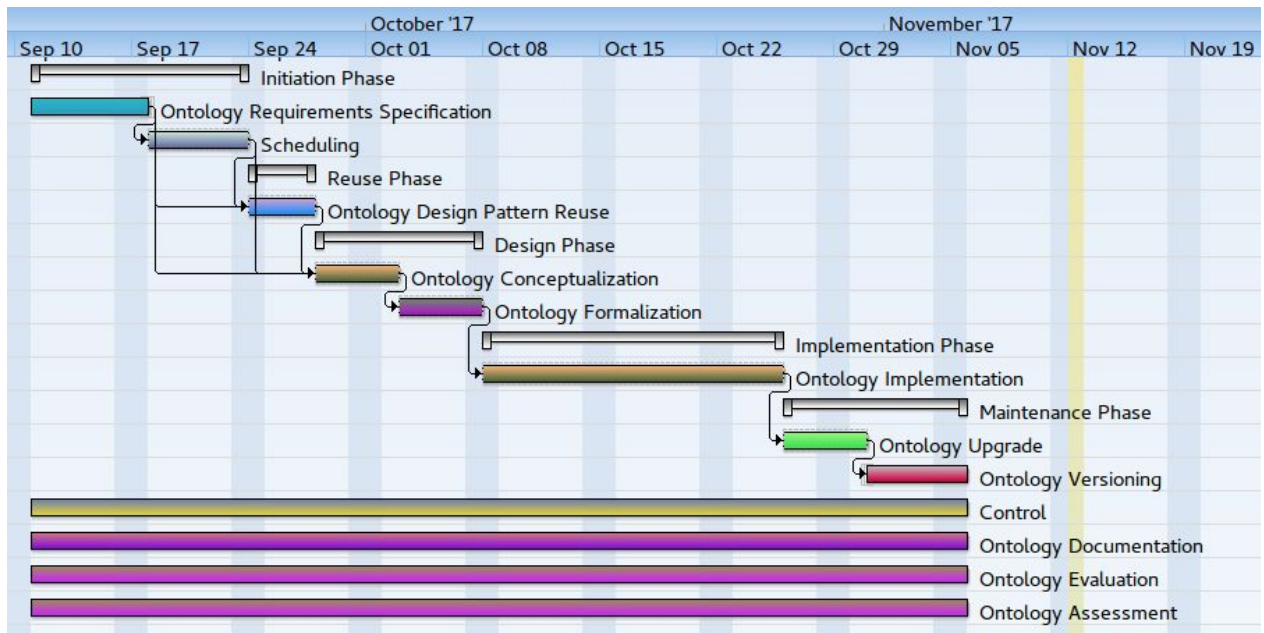
3.4.1.1. Fase de Inicio

Durante la fase de inicio de la metodología NeOn se desarrolla el Documento de Especificación de Requisitos Ontológicos (DERO) y se realiza la planificación del desarrollo de las ontologías. Se crearon quince Documentos de especificación de Requisitos Ontológicos (cf. Apéndice 7) en base a los Proyectos de grado que conforman la muestra. Los objetivos generales de los Proyectos de grado seleccionados conforman los requisitos funcionales de las ontologías creadas. Para el lenguaje de implementación se consideraron las líneas de diseño de las Ontologías facetadas ligeras y los usos previstos se ajustaron a las pruebas realizadas en el presente documento.

El siguiente proceso en la fase de inicio de la metodología NeOn define el desarrollo de la planificación de las ontologías. Para el presente proyecto se utilizó el plugin Gantt de la herramienta NeOn Toolkit. Mediante el asistente que provee la herramienta, se ha diseñado el Diagrama de planificación con la implementación de los dos escenarios previamente mencionados. Además, se seguirá el mismo calendario para el desarrollo de todas las ontologías de prueba. El diagrama resultante, se muestra a continuación.

¹⁸ La definición de la muestra se encuentra en el capítulo 1: Marco introductorio.

Figura N° 3.27.
Diagrama de planificación de la ontología



Fuente: Elaboración propia, 2017.

3.4.1.2. Fase de diseño

En la fase de diseño, la metodología NeOn define los procesos a través de los escenarios escogidos. El primer proceso de la fase de diseño se refiere a la Reutilización de patrones de diseño ontológicos, perteneciente al Escenario 7. Durante este proceso, se debe seleccionar el Patrón más adecuado para la construcción de la ontología.

Debido al uso de Ontologías facetadas ligeras, se incluyó el patrón definido durante el diseño del modelo ontológico (Figura 3.10.) a fin de mantener una estructura única en la construcción de conocimiento con la herramienta.

Posteriormente, la metodología NeOn define el proceso de Conceptualización de ontologías perteneciente al Escenario 1. Durante este proceso se define el modelo de datos a usar en el desarrollo de la ontología. A continuación se describen los elementos de la misma.

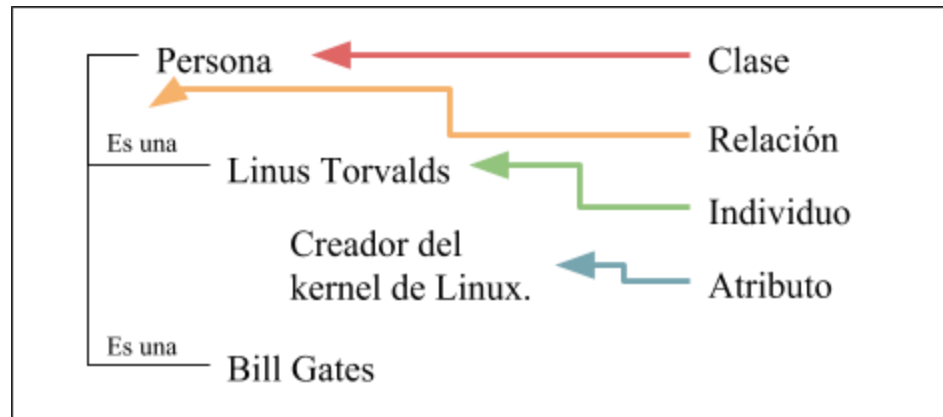
- Clases: Las clases son conjuntos de colecciones de objetos, permiten la agrupación de individuos y otras clases, pueden ser organizados como taxonomías.
- Individuos: Los individuos son instancias de las clases previamente mencionadas y

definen las representaciones de objetos.

- Atributos: Son las propiedades que cada individuo o relación tiene.
- Relaciones: Manera en que las Señales de conocimiento se relacionan unas con otras.

Figura N° 3.28.

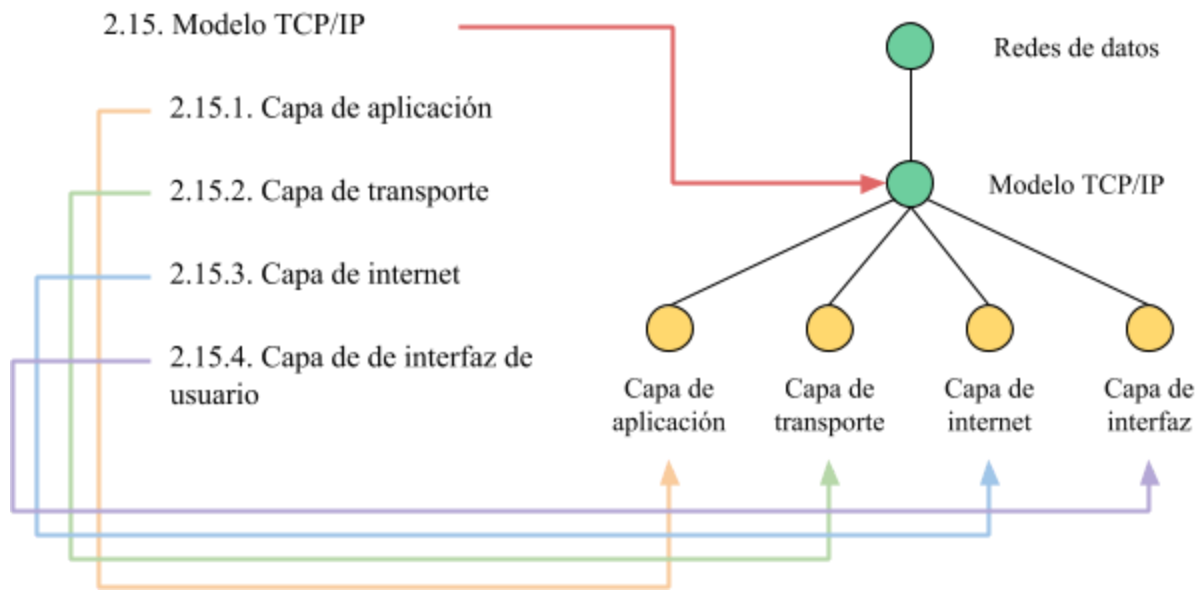
Ejemplo de Clases, Individuos, Atributos y relaciones



Fuente: Elaboración propia, 2017.

La formalización de ontologías constituye el siguiente proceso de la fase de diseño de la metodología NeOn. Permite la creación de un modelo semi computable de la ontología y será aquella que defina la extracción de los elementos requeridos para la fase de implementación, usando el patrón de diseño ontológico definido previamente.

Para la creación de la ontología se creará una réplica de los contenidos teóricos de Proyectos de Grado de estudiantes graduados de la Universidad del Valle, que forman parte de la muestra, manteniendo su orden jerárquico. A continuación se muestra un ejemplo de la conversión de contenidos a una ontología facetada ligera.

Figura N° 3.29.**Transformación de contenidos a ontologías facetadas ligeras**

Fuente: Elaboración propia, 2017.

3.4.1.3. Fase de implementación

Para la fase de implementación, NeOn define la creación de un modelo computable de las ontologías creadas en el presente proyecto, con la herramienta propuesta. Se describen a continuación los pasos necesarios a realizarse en la herramienta.

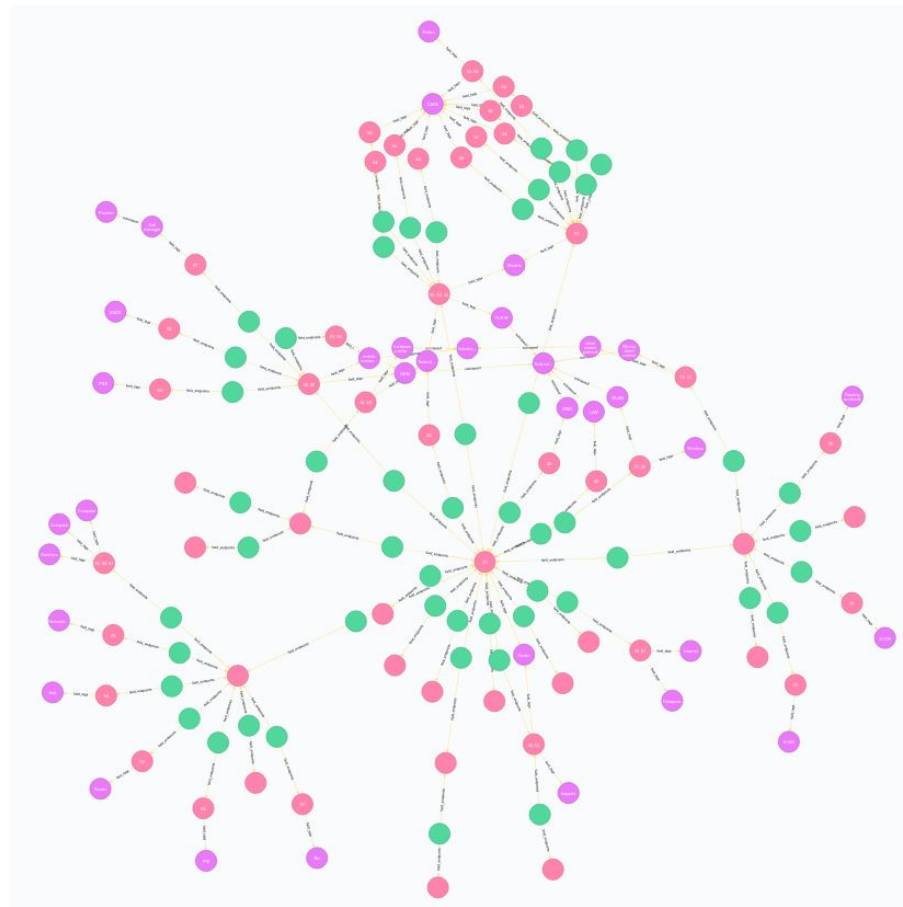
- Creación de clases: Se crea en la herramienta la Base de conocimiento definida por Open Calais. Cada tipo de entidad (cf. Anexo 1) se define como una clase dentro de las ontologías de prueba y las mismas son creadas automáticamente durante la instalación de la herramienta.
- Creación de individuos: Las instancias son representaciones de conceptos que son agregados como notas y son definidas por el usuario como Señales de conocimiento dentro del modelo de gestión de conocimiento personal de Völkel. En esta etapa, se realizó la conversión definida en el punto anterior de los subtítulos del capítulo Marco Teórico de los Proyectos de Grado de los estudiantes de la carrera de Ingeniería de Sistemas.

- Creación de propiedades: Durante este proceso, se crea la descripción de cada propiedad. Debido al tiempo requerido que implicaría copiar el contenido textual de los Proyectos de Grado, este paso ha sido omitido.

Las relaciones entre individuos y clases serán creadas automáticamente siguiendo la jerarquía de los conceptos definido por los autores de los Proyectos de Grado.

A continuación se muestra una vista general de una ontología de prueba:

Figura N° 3.30.
Vista general de una ontología de prueba



Fuente: Elaboración propia, 2017.

3.4.2. MEDICIÓN DE LA EFICACIA DE LA RECUPERACIÓN DE CONOCIMIENTO

Para medir la eficacia de la recuperación de conocimiento con la herramienta, se ha seguido el

marco de trabajo definido por Lewandowski¹⁹ mediante la medición de la precisión y sensibilidad de la recuperación de contenidos. Cabe notar que, para la medición de la eficacia de la recuperación de conocimiento, se ha considerado como un rango válido un porcentaje mayor a 71% como valor-F, debido a que es la nota mínima para la aprobación de Proyectos de grado. Considerar Proyectos aprobado implica que un jurado de expertos, conformado por tribunales internos y externos a la Universidad, han considerado que la aplicación de habilidades y conocimientos ha sido suficiente para el desarrollo de la propuesta. A continuación se listan los Proyectos de Grado que se consideraron como parte de la muestra.

¹⁹ Véase Capítulo 2: Medición de eficacia de Lewandowski.

Cuadro N° 3.23.
Listado de Proyectos de Grado

| Código | Título |
|---------------|--|
| PG-01 | Sistema de promoción de fondos de inversión utilizando dispositivos móviles. Caso: FORTALEZA SAFI S.A. |
| PG-02 | Aplicación de redes convergentes para grupos empresariales. Caso de estudio: Grupo empresarial Kantutani S.A. |
| PG-03 | Guía de implementación de portales colaborativos basados en la administración de contenidos empresariales. Caso de estudio: Pymes de telecomunicaciones. |
| PG-04 | Optimización de una red frame relay integrando voz y datos. Caso de estudio: RUAT. |
| PG-05 | Gobierno electrónico para el seguimiento de los procesos de saneamiento de tierras. Caso: Instituto Nacional de Reforma Agraria INRA. |
| PG-06 | Aplicación de DataSnap Multi - Tier para el desarrollo de un sistema informático que permita realizar el registro de datos del proceso de reclutamiento y selección de personal. |
| PG-07 | Sistema de información de seguimiento de equipos para el departamento de internetworking de la empresa Alpha Systems S.R.L. |
| PG-08 | Sistema de atención médica para la caja de salud CORDES. |
| PG-09 | Integración de una arquitectura Business Intelligence con el modelo Balanced ScoreCard para el control y seguimiento de la planificación estratégica. Caso: SOBOCE S.A. |
| PG-10 | Desarrollo de un portal web basado en E-salud. |
| PG-11 | Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| PG-12 | Sistema de métricas de seguridad con tableros de control integrados para medir la efectividad de los controles de entidades financieras. Caso de Estudio: Banco de Crédito de Bolivia S.A. |
| PG-13 | Modelo basado en SOA para la integración de Sistemas Informáticos que permita implementar una arquitectura de información. |
| PG-14 | Sistema experto para el apoyo de diagnóstico y tratamiento de infecciones de |

| | |
|-------|--|
| | transmisión sexual. |
| PG-15 | Sistema para la identificación de huellas dactilares para el estudio dactiloscópico de la unidad especializada “Fuera Especial de Lucha Contra el Crimen (FELCC)”. |

Fuente: Elaboración propia, 2017.

3.4.2.1. Selección de queries

En este apartado se define el tipo de *queries* que serán ejecutados. Según la clasificación de Broder²⁰, se consideraron los *queries* informativos.

Las temáticas de los *queries*, deben reflejar las necesidades del usuario, es por eso que serán creados a partir de los objetivos generales de los Proyectos de Grado que forman parte de la muestra. A fin de obtener las palabras clave para la construcción de consultas, se hizo uso del servicio web *Spanish Keyword Extractor*²¹. A continuación se describen los *queries* seleccionados para los quince Proyectos de Grado que forman parte de la muestra.

²⁰ Pueden incluir *queries* informativos, *queries* navegacionales y *queries* transaccionales.

²¹ <https://open.blockspring.com/bs/spanish-keyword-extractor-monkeylearn>

Cuadro N° 3.24.
Definición de query para PG-01

| Atributo | Valor |
|------------------|---|
| Objetivo | Diseñar y desarrollar un sistema de promoción de Fondos de Inversión utilizando dispositivos móviles, destinado a optimizar el desempeño de los Promotores de Inversión de la empresa FORTALEZA SAFI S.A. |
| Keyword | FORTALEZA SAFI, Inversión, Diseñar, desempeño, Promotores, promoción, Fondos, sistema, dispositivos |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('FORTALEZA')) OR toLower(node.title) CONTAINS toLower('SAFI') OR toLower(node.title) CONTAINS toLower('Inversión') OR toLower(node.title) CONTAINS toLower('Diseñar') OR toLower(node.title) CONTAINS toLower('desempeño') OR toLower(node.title) CONTAINS toLower('Promotores') OR toLower(node.title) CONTAINS toLower('promoción') OR toLower(node.title) CONTAINS toLower('Fondos') OR toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('dispositivos')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('FORTALEZA')) OR toLower(singleNode.title) CONTAINS toLower('SAFI') OR toLower(singleNode.title) CONTAINS toLower('Inversión') OR toLower(singleNode.title) CONTAINS toLower('Diseñar') OR toLower(singleNode.title) CONTAINS toLower('desempeño') OR toLower(singleNode.title) CONTAINS toLower('Promotores') OR toLower(singleNode.title) CONTAINS toLower('promoción') OR toLower(singleNode.title) CONTAINS toLower('Fondos') OR toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('dispositivos')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.25.**Definición de query para PG-02**

| Atributo | Valor |
|------------------|--|
| Objetivo | Diseñar una red convergente para grupos empresariales multi-servicios utilizando la tecnología de comunicaciones existente en Bolivia, para integrar los servicios de voz, datos y video, tomando en cuenta la calidad de servicio y seguridad para lograr una red escalable y flexible para reducir el costo administrativo de la red. |
| Keyword | servicio, costo administrativo, red escalable, red, red convergente, grupo empresarial, comunicaciones, Diseñar, multiservicios, seguridad, cuenta, voz, datos, video |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('servicio') OR toLower(node.title) CONTAINS toLower('costo administrativo') OR toLower(node.title) CONTAINS toLower('red escalable') OR toLower(node.title) CONTAINS toLower('red') OR toLower(node.title) CONTAINS toLower('red convergente') OR toLower(node.title) CONTAINS toLower('grupo empresarial') OR toLower(node.title) CONTAINS toLower('comunicaciones') OR toLower(node.title) CONTAINS toLower('Diseñar') OR toLower(node.title) CONTAINS toLower('multiservicios') OR toLower(node.title) CONTAINS toLower('seguridad') OR toLower(node.title) CONTAINS toLower('cuenta') OR toLower(node.title) CONTAINS toLower('voz') OR toLower(node.title) CONTAINS toLower('datos') OR toLower(node.title) CONTAINS toLower('video')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('servicio') OR toLower(singleNode.title) CONTAINS toLower('costo administrativo') OR toLower(singleNode.title) CONTAINS toLower('red escalable') OR toLower(singleNode.title) CONTAINS toLower('red') OR toLower(singleNode.title) CONTAINS toLower('red convergente') OR toLower(singleNode.title) CONTAINS toLower('grupo empresarial') OR toLower(singleNode.title) CONTAINS toLower('comunicaciones') OR toLower(singleNode.title) CONTAINS toLower('Diseñar') OR toLower(singleNode.title) CONTAINS toLower('multiservicios') OR toLower(singleNode.title) CONTAINS toLower('seguridad') OR toLower(singleNode.title) CONTAINS toLower('cuenta') OR toLower(singleNode.title) CONTAINS toLower('voz') OR toLower(singleNode.title) CONTAINS toLower('datos') OR toLower(singleNode.title) CONTAINS toLower('video')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.26.
Definición de query para PG-03

| Atributo | Valor |
|------------------|---|
| Objetivo | Formalizar y elaborar una guía de implementación de portales colaborativos basados en la administración de contenidos empresariales para contribuir a integrar los servicios que un ambiente colaborativo requiere en Pymes de telecomunicaciones. |
| Keyword | portal, colaborativo, contenidos empresariales, ambiente, implementación, guía, Pyme, Formalizar, telecomunicaciones, servicios, administración, ECM |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('portal') OR toLower(node.title) CONTAINS toLower('colaborativo') OR toLower(node.title) CONTAINS toLower('contenidos empresariales') OR toLower(node.title) CONTAINS toLower('ambiente') OR toLower(node.title) CONTAINS toLower('implementación') OR toLower(node.title) CONTAINS toLower('guía') OR toLower(node.title) CONTAINS toLower('Pyme') OR toLower(node.title) CONTAINS toLower('Formalizar') OR toLower(node.title) CONTAINS toLower('telecomunicaciones') OR toLower(node.title) CONTAINS toLower('servicios') OR toLower(node.title) CONTAINS toLower('administración') OR toLower(node.title) CONTAINS toLower('ECM')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('portal') OR toLower(singleNode.title) CONTAINS toLower('colaborativo') OR toLower(singleNode.title) CONTAINS toLower('contenidos empresariales') OR toLower(singleNode.title) CONTAINS toLower('ambiente') OR toLower(singleNode.title) CONTAINS toLower('implementación') OR toLower(singleNode.title) CONTAINS toLower('guía') OR toLower(singleNode.title) CONTAINS toLower('Pyme') OR toLower(singleNode.title) CONTAINS toLower('Formalizar') OR toLower(singleNode.title) CONTAINS toLower('telecomunicaciones') OR toLower(singleNode.title) CONTAINS toLower('servicios') OR toLower(singleNode.title) CONTAINS toLower('administración') OR toLower(singleNode.title) CONTAINS toLower('ECM')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.27.**Definición de query para PG-04**

| Atributo | Valor |
|------------------|---|
| Objetivo | Realiza la optimización y ampliación de servicios de la red Frame Relay del RUAT haciendo énfasis en la tecnología de VoFR para lograr satisfacer las necesidades de comunicación de voz y datos de la organización bajo los criterios de seguridad, fiabilidad y costos. |
| Keyword | red, Frame Relay, servicios, seguridad, RUAT, criterios, costos, organización, necesidades, ampliación, fiabilidad, voz, datos |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('red') OR toLower(node.title) CONTAINS toLower('Frame Relay') OR toLower(node.title) CONTAINS toLower('servicios') OR toLower(node.title) CONTAINS toLower('seguridad') OR toLower(node.title) CONTAINS toLower('RUAT') OR toLower(node.title) CONTAINS toLower('criterios') OR toLower(node.title) CONTAINS toLower('costos') OR toLower(node.title) CONTAINS toLower('organización') OR toLower(node.title) CONTAINS toLower('necesidades') OR toLower(node.title) CONTAINS toLower('ampliación') OR toLower(node.title) CONTAINS toLower('fiabilidad') OR toLower(node.title) CONTAINS toLower('voz') OR toLower(node.title) CONTAINS toLower('datos')) WITH facets, node</p> <p>OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('red') OR toLower(singleNode.title) CONTAINS toLower('Frame Relay') OR toLower(singleNode.title) CONTAINS toLower('servicios') OR toLower(singleNode.title) CONTAINS toLower('seguridad') OR toLower(singleNode.title) CONTAINS toLower('RUAT') OR toLower(singleNode.title) CONTAINS toLower('criterios') OR toLower(singleNode.title) CONTAINS toLower('costos') OR toLower(singleNode.title) CONTAINS toLower('organización') OR toLower(singleNode.title) CONTAINS toLower('necesidades') OR toLower(singleNode.title) CONTAINS toLower('ampliación') OR toLower(singleNode.title) CONTAINS toLower('fiabilidad') OR toLower(singleNode.title) CONTAINS toLower('voz') OR toLower(singleNode.title) CONTAINS toLower('datos'))</p> <p>WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.28.**Definición de query para PG-05**

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar e implementar una aplicación de Gobierno Electrónico para un control adecuado, ágil y segura tanto geográfica y alfanumérica, para coadyuvar a la toma de decisiones en la institución, las organizaciones sociales o sectoriales y quienes sean partes interesadas. |
| Keyword | partes interesadas, organizaciones sociales, Gobierno, Electrónico, control adecuado, alfanumérica, aplicación, sistema, decisiones, institución, toma, geográfica |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('partes interesadas') OR toLower(node.title) CONTAINS toLower('organizaciones sociales') OR toLower(node.title) CONTAINS toLower('Gobierno') OR toLower(node.title) CONTAINS toLower('Electrónico') OR toLower(node.title) CONTAINS toLower('control adecuado') OR toLower(node.title) CONTAINS toLower('alfanumérica') OR toLower(node.title) CONTAINS toLower('aplicación') OR toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('decisiones') OR toLower(node.title) CONTAINS toLower('institución') OR toLower(node.title) CONTAINS toLower('toma') OR toLower(node.title) CONTAINS toLower('geográfica')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('partes interesadas') OR toLower(singleNode.title) CONTAINS toLower('organizaciones sociales') OR toLower(singleNode.title) CONTAINS toLower('Gobierno') OR toLower(singleNode.title) CONTAINS toLower('Electrónico') OR toLower(singleNode.title) CONTAINS toLower('control adecuado') OR toLower(singleNode.title) CONTAINS toLower('alfanumérica') OR toLower(singleNode.title) CONTAINS toLower('aplicación') OR toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('decisiones') OR toLower(singleNode.title) CONTAINS toLower('institución') OR toLower(singleNode.title) CONTAINS toLower('toma') OR toLower(singleNode.title) CONTAINS toLower('geográfica')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.29.**Definición de query para PG-06**

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un sistema informático para el registro de datos personales del funcionario de acuerdo al proceso de reclutamiento y selección de personal administrado por el encargado de RR.HH. de la Autoridad de Agua Potable y Saneamiento Básico, aplicando DataSnap como tecnología Multicapa (Multi-Tier) para el desarrollo de aplicaciones de base de datos de gestión. |
| Keyword | sistema informático, Agua Potable, tecnología Multicapa, datos personales, personal, gestión, reclutamiento, proceso, Autoridad, aplicaciones, DataSnap |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('sistema informático') OR toLower(node.title) CONTAINS toLower('Agua Potable') OR toLower(node.title) CONTAINS toLower('tecnología Multicapa') OR toLower(node.title) CONTAINS toLower('datos personales') OR toLower(node.title) CONTAINS toLower('personal') OR toLower(node.title) CONTAINS toLower('gestión') OR toLower(node.title) CONTAINS toLower('reclutamiento') OR toLower(node.title) CONTAINS toLower('proceso') OR toLower(node.title) CONTAINS toLower('Autoridad') OR toLower(node.title) CONTAINS toLower('aplicaciones') OR toLower(node.title) CONTAINS toLower('DataSnap')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('sistema informático') OR toLower(singleNode.title) CONTAINS toLower('Agua Potable') OR toLower(singleNode.title) CONTAINS toLower('tecnología Multicapa') OR toLower(singleNode.title) CONTAINS toLower('datos personales') OR toLower(singleNode.title) CONTAINS toLower('personal') OR toLower(singleNode.title) CONTAINS toLower('gestión') OR toLower(singleNode.title) CONTAINS toLower('reclutamiento') OR toLower(singleNode.title) CONTAINS toLower('proceso') OR toLower(singleNode.title) CONTAINS toLower('Autoridad') OR toLower(singleNode.title) CONTAINS toLower('aplicaciones') OR toLower(singleNode.title) CONTAINS toLower('DataSnap')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.30.
Definición de query para PG-07

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un sistema de información automatizado para el departamento de internetworking de la empresa, que permita llevar un registro oportuno de las tareas de cotizaciones, pedidos, préstamos, RMA y pre-ventas. |
| Keyword | sistema, registro oportuno, préstamos, empresa, tareas, preventas, departamento, RMA, pedidos, Alpha Systems, cliente |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('registro oportuno') OR toLower(node.title) CONTAINS toLower('préstamos') OR toLower(node.title) CONTAINS toLower('empresa') OR toLower(node.title) CONTAINS toLower('tareas') OR toLower(node.title) CONTAINS toLower('preventas') OR toLower(node.title) CONTAINS toLower('departamento') OR toLower(node.title) CONTAINS toLower('RMA') OR toLower(node.title) CONTAINS toLower('pedidos') OR toLower(node.title) CONTAINS toLower('Alpha Systems') OR toLower(node.title) CONTAINS toLower('cliente')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('registro oportuno') OR toLower(singleNode.title) CONTAINS toLower('préstamos') OR toLower(singleNode.title) CONTAINS toLower('empresa') OR toLower(singleNode.title) CONTAINS toLower('tareas') OR toLower(singleNode.title) CONTAINS toLower('preventas') OR toLower(singleNode.title) CONTAINS toLower('departamento') OR toLower(singleNode.title) CONTAINS toLower('RMA') OR toLower(singleNode.title) CONTAINS toLower('pedidos') OR toLower(singleNode.title) CONTAINS toLower('Alpha Systems') OR toLower(singleNode.title) CONTAINS toLower('cliente')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.31.
Definición de query para PG-08

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar e implementar el sistema Web de atención médica para la caja de salud CORDES en las sucursales La Paz y El Alto. |
| Keyword | salud, CORDES, sistema Web, El Alto, atención, caja, sucursales, La Paz |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('salud') OR toLower(node.title) CONTAINS toLower('CORDES') OR toLower(node.title) CONTAINS toLower('sistema Web') OR toLower(node.title) CONTAINS toLower('El Alto') OR toLower(node.title) CONTAINS toLower('atención') OR toLower(node.title) CONTAINS toLower('caja') OR toLower(node.title) CONTAINS toLower('sucursales') OR toLower(node.title) CONTAINS toLower('La Paz') OR toLower(node.title) CONTAINS toLower('RUP') OR toLower(node.title) CONTAINS toLower('UML')) WITH facets, node</p> <p>OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('salud') OR toLower(singleNode.title) CONTAINS toLower('CORDES') OR toLower(singleNode.title) CONTAINS toLower('sistema Web') OR toLower(singleNode.title) CONTAINS toLower('El Alto') OR toLower(singleNode.title) CONTAINS toLower('atención') OR toLower(singleNode.title) CONTAINS toLower('caja') OR toLower(singleNode.title) CONTAINS toLower('sucursales') OR toLower(singleNode.title) CONTAINS toLower('La Paz') OR toLower(singleNode.title) CONTAINS toLower('RUP') OR toLower(singleNode.title) CONTAINS toLower('UML')) WITH facets, singleNode, node</p> <p>OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.32.**Definición de query para PG-09**

| Atributo | Valor |
|------------------|---|
| Objetivo | Integrar el modelo Balanced Scorecard de SOBOCE con una arquitectura de Business Intelligence mediante la creación de procedimientos y un prototipo de modelo. |
| Keyword | Balanced Scorecard, Business Intelligence, prototipo, creación, SOBOCE, modelo, procedimientos, arquitectura |
| Query resultante | <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('Balanced Scorecard')) OR toLower(node.title) CONTAINS toLower('Business Intelligence') OR toLower(node.title) CONTAINS toLower('prototipo') OR toLower(node.title) CONTAINS toLower('creación') OR toLower(node.title) CONTAINS toLower('SOBOCE') OR toLower(node.title) CONTAINS toLower('modelo') OR toLower(node.title) CONTAINS toLower('procedimientos') OR toLower(node.title) CONTAINS toLower('arquitectura') OR toLower(node.title) CONTAINS toLower('BI') OR toLower(node.title) CONTAINS toLower('BSC') OR toLower(node.title) CONTAINS toLower('DW')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('Balanced Scorecard')) OR toLower(singleNode.title) CONTAINS toLower('Business Intelligence') OR toLower(singleNode.title) CONTAINS toLower('prototipo') OR toLower(singleNode.title) CONTAINS toLower('creación') OR toLower(singleNode.title) CONTAINS toLower('SOBOCE') OR toLower(singleNode.title) CONTAINS toLower('modelo') OR toLower(singleNode.title) CONTAINS toLower('procedimientos') OR toLower(singleNode.title) CONTAINS toLower('arquitectura') OR toLower(singleNode.title) CONTAINS toLower('BI') OR toLower(singleNode.title) CONTAINS toLower('BSC') OR toLower(singleNode.title) CONTAINS toLower('DW')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.33.**Definición de query para PG-10**

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un Portal Web basado en E-Salud con la finalidad de mejorar el servicio médico de los Hospitales, mediante la administración adecuada de la información clínica de los pacientes, un diagnóstico preciso a distancia, la difusión de información orientada al ciudadano y especialista. |
| Keyword | Portal, Portal Web, administración adecuada, información clínica, información orientada, diagnóstico preciso, servicio, Salud, distancia, ciudadano, especialista |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('Portal') OR toLower(node.title) CONTAINS toLower('Portal Web') OR toLower(node.title) CONTAINS toLower('administración adecuada') OR toLower(node.title) CONTAINS toLower('información clínica') OR toLower(node.title) CONTAINS toLower('información orientada') OR toLower(node.title) CONTAINS toLower('diagnóstico preciso') OR toLower(node.title) CONTAINS toLower('servicio') OR toLower(node.title) CONTAINS toLower('Salud') OR toLower(node.title) CONTAINS toLower('distancia') OR toLower(node.title) CONTAINS toLower('ciudadano') OR toLower(node.title) CONTAINS toLower('especialista')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('Portal') OR toLower(singleNode.title) CONTAINS toLower('Portal Web') OR toLower(singleNode.title) CONTAINS toLower('administración adecuada') OR toLower(singleNode.title) CONTAINS toLower('información clínica') OR toLower(singleNode.title) CONTAINS toLower('información orientada') OR toLower(singleNode.title) CONTAINS toLower('diagnóstico preciso') OR toLower(singleNode.title) CONTAINS toLower('servicio') OR toLower(singleNode.title) CONTAINS toLower('Salud') OR toLower(singleNode.title) CONTAINS toLower('distancia') OR toLower(singleNode.title) CONTAINS toLower('ciudadano') OR toLower(singleNode.title) CONTAINS toLower('especialista')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.34.
Definición de query para PG-11

| Atributo | Valor |
|------------------|--|
| Objetivo | Desarrollar un sistema informático integrado de los procesos de Afiliación, Atención Ambulatoria, Fichaje y Farmacia como propuesta para la Caja Petrolera de Salud Regional La Paz a fin de que se logre satisfacer las necesidades de la CPS y de sus asegurados. |
| Keyword | sistema, Atención Ambulatoria, Salud Regional, Caja Petrolera, procesos, Afiliación, Farmacia, necesidades, Fichaje, CPS |
| Query resultante | <p>OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('Atención Ambulatoria') OR toLower(node.title) CONTAINS toLower('Salud Regional') OR toLower(node.title) CONTAINS toLower('Caja Petrolera') OR toLower(node.title) CONTAINS toLower('procesos') OR toLower(node.title) CONTAINS toLower('Afiliación') OR toLower(node.title) CONTAINS toLower('Farmacia') OR toLower(node.title) CONTAINS toLower('necesidades') OR toLower(node.title) CONTAINS toLower('Fichaje') OR toLower(node.title) CONTAINS toLower('CPS') OR toLower(node.title) CONTAINS toLower('software') OR toLower(node.title) CONTAINS toLower('web')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('Atención Ambulatoria') OR toLower(singleNode.title) CONTAINS toLower('Salud Regional') OR toLower(singleNode.title) CONTAINS toLower('Caja Petrolera') OR toLower(singleNode.title) CONTAINS toLower('procesos') OR toLower(singleNode.title) CONTAINS toLower('Afiliación') OR toLower(singleNode.title) CONTAINS toLower('Farmacia') OR toLower(singleNode.title) CONTAINS toLower('necesidades') OR toLower(singleNode.title) CONTAINS toLower('Fichaje') OR toLower(singleNode.title) CONTAINS toLower('CPS') OR toLower(singleNode.title) CONTAINS toLower('software') OR toLower(singleNode.title) CONTAINS toLower('web')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet</p> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.35.**Definición de query para PG-12**

| Atributo | Valor |
|------------------|---|
| Objetivo | Diseñar un sistema de métricas de seguridad con tableros de control integrados para cuantificar los controles de seguridad, su efectividad e identificar posibles acciones de mejora en entidades financieras. |
| Keyword | seguridad, entidades financieras, control, Diseñar, efectividad, tableros, sistema, mejora, acciones, métricas |
| Query resultante | <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('seguridad') OR toLower(node.title) CONTAINS toLower('entidades financieras') OR toLower(node.title) CONTAINS toLower('control') OR toLower(node.title) CONTAINS toLower('Diseñar') OR toLower(node.title) CONTAINS toLower('efectividad') OR toLower(node.title) CONTAINS toLower('tableros') OR toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('mejora') OR toLower(node.title) CONTAINS toLower('acciones') OR toLower(node.title) CONTAINS toLower('métricas')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('seguridad') OR toLower(singleNode.title) CONTAINS toLower('entidades financieras') OR toLower(singleNode.title) CONTAINS toLower('control') OR toLower(singleNode.title) CONTAINS toLower('Diseñar') OR toLower(singleNode.title) CONTAINS toLower('efectividad') OR toLower(singleNode.title) CONTAINS toLower('tableros') OR toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('mejora') OR toLower(singleNode.title) CONTAINS toLower('acciones') OR toLower(singleNode.title) CONTAINS toLower('métricas')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.36.
Definición de query para PG-13

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un modelo de Integración de Sistemas basados en SOA, para integrar funciones o servicios, con el propósito de obtener un mejor acceso a la información a través de una arquitectura de información normalizada. |
| Keyword | información normalizada, Sistema, SOA, servicio, Integración, acceso, modelo, información, propósito, función |
| Query resultante | <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('información normalizada') OR toLower(node.title) CONTAINS toLower('Sistema') OR toLower(node.title) CONTAINS toLower('SOA') OR toLower(node.title) CONTAINS toLower('servicio') OR toLower(node.title) CONTAINS toLower('Integración') OR toLower(node.title) CONTAINS toLower('acceso') OR toLower(node.title) CONTAINS toLower('modelo') OR toLower(node.title) CONTAINS toLower('información') OR toLower(node.title) CONTAINS toLower('propósito') OR toLower(node.title) CONTAINS toLower('función')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('información normalizada') OR toLower(singleNode.title) CONTAINS toLower('Sistema') OR toLower(singleNode.title) CONTAINS toLower('SOA') OR toLower(singleNode.title) CONTAINS toLower('servicio') OR toLower(singleNode.title) CONTAINS toLower('Integración') OR toLower(singleNode.title) CONTAINS toLower('acceso') OR toLower(singleNode.title) CONTAINS toLower('modelo') OR toLower(singleNode.title) CONTAINS toLower('información') OR toLower(singleNode.title) CONTAINS toLower('propósito') OR toLower(singleNode.title) CONTAINS toLower('función')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.37.**Definición de query para PG-14**

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un sistema basado en lógica difusa que determine mediante los signos y síntomas de las infecciones de Transmisión Sexual, posibles soluciones que apoyen el diagnóstico y tratamiento médico. |
| Keyword | sistema, Transmisión, Sexual, diagnóstico, infecciones, soluciones, tratamiento, signos |
| Query resultante | <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('sistema') OR toLower(node.title) CONTAINS toLower('Transmisión') OR toLower(node.title) CONTAINS toLower('Sexual') OR toLower(node.title) CONTAINS toLower('diagnóstico') OR toLower(node.title) CONTAINS toLower('infecciones') OR toLower(node.title) CONTAINS toLower('soluciones') OR toLower(node.title) CONTAINS toLower('tratamiento') OR toLower(node.title) CONTAINS toLower('signos')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('sistema') OR toLower(singleNode.title) CONTAINS toLower('Transmisión') OR toLower(singleNode.title) CONTAINS toLower('Sexual') OR toLower(singleNode.title) CONTAINS toLower('diagnóstico') OR toLower(singleNode.title) CONTAINS toLower('infecciones') OR toLower(singleNode.title) CONTAINS toLower('soluciones') OR toLower(singleNode.title) CONTAINS toLower('tratamiento') OR toLower(singleNode.title) CONTAINS toLower('signos')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

Cuadro N° 3.38.
Definición de query para PG-15

| Atributo | Valor |
|------------------|---|
| Objetivo | Desarrollar un sistema para la automatización de Identificación de Huellas Dactilares para el estudio Dactiloscópico de la Unidad Especializada “Fuerza Especial de Lucha Contra el Crimen (FELCC)”, para reducir el tiempo en la identificación y control de las personas. |
| Keyword | Identificación, estudio Dactiloscópico, Fuerza Especial, Unidad Especializada, Huellas Dactilares, control, personas, FELCC, tiempo, Crimen |
| Query resultante | <pre> OPTIONAL MATCH facets = (initialNode)-[*2]-(node:`entity:node`) WHERE (initialNode.vid = 'social_tags' OR initialNode.vid = 'topic_tags' OR initialNode.vid = 'industry_tags' OR initialNode.vid = 'markup_tags') OR (toLower(node.title) CONTAINS toLower('Identificación') OR toLower(node.title) CONTAINS toLower('estudio') OR toLower(node.title) CONTAINS toLower('Dactiloscopico') OR toLower(node.title) CONTAINS toLower('dactiloscopia') OR toLower(node.title) CONTAINS toLower('dactilar') OR toLower(node.title) CONTAINS toLower('Fuerza Especial') OR toLower(node.title) CONTAINS toLower('Unidad') OR toLower(node.title) CONTAINS toLower('Huella') OR toLower(node.title) CONTAINS toLower('Dactilar') OR toLower(node.title) CONTAINS toLower('control') OR toLower(node.title) CONTAINS toLower('persona') OR toLower(node.title) CONTAINS toLower('FELCC') OR toLower(node.title) CONTAINS toLower('tiempo') OR toLower(node.title) CONTAINS toLower('Crimen') OR toLower(node.title) CONTAINS toLower('organizacion') OR toLower(node.title) CONTAINS toLower('sensor')) WITH facets, node OPTIONAL MATCH (singleNode:`entity:node`) WHERE (toLower(singleNode.title) CONTAINS toLower('Identificación') OR toLower(singleNode.title) CONTAINS toLower('estudio') OR toLower(singleNode.title) CONTAINS toLower('Dactiloscopico') OR toLower(singleNode.title) CONTAINS toLower('dactiloscopia') OR toLower(singleNode.title) CONTAINS toLower('dactilar') OR toLower(singleNode.title) CONTAINS toLower('Fuerza Especial') OR toLower(singleNode.title) CONTAINS toLower('Unidad') OR toLower(singleNode.title) CONTAINS toLower('Huella') OR toLower(singleNode.title) CONTAINS toLower('Dactilar') OR toLower(singleNode.title) CONTAINS toLower('control') OR toLower(singleNode.title) CONTAINS toLower('persona') OR toLower(singleNode.title) CONTAINS toLower('FELCC') OR toLower(singleNode.title) CONTAINS toLower('tiempo') OR toLower(singleNode.title) CONTAINS toLower('Crimen') OR toLower(singleNode.title) CONTAINS toLower('organizacion') OR toLower(singleNode.title) CONTAINS toLower('sensor')) WITH facets, singleNode, node OPTIONAL MATCH (singleNode)-[relationSingle]->(tags:`entity:taxonomy_term`) OPTIONAL MATCH (node)-[relationFacet]->(tags:`entity:taxonomy_term`) RETURN facets, tags, singleNode, relationSingle, relationFacet </pre> |

Fuente: Elaboración propia, 2017.

3.4.2.2. Obtención de resultados

Si se trata de un sistema que realiza la búsqueda en una base de datos estática, como es el caso de las presentes ontologías, no es necesario guardar los resultados, ya que no variarán dentro del período de prueba. En ese sentido, durante esta etapa se ejecutaron las consultas y se analizaron en etapas posteriores.

3.4.2.3. Presentación de resultados

La presentación de resultados implica el análisis de patrones de presentación de los contenidos al usuario. En el presente proyecto se omitió la presente sección por el límite de tiempo para el desarrollo del mismo. Sin embargo, la presentación de los contenidos en grafos, representa una técnica actual para la representación de conocimiento (Malhotra, 2015), mediante la evolución desde los mapas conceptuales, mapas cognitivos difusos, la web semántica (sin el uso de grafos) y FreeBase.

3.4.2.4. Valoración de resultados

En esta sección, se definieron los elementos de la Matriz de confusión en base a los resultados obtenidos con la selección y ejecución de consultas en los apartados anteriores. A continuación se muestran las Matrices de confusión resultantes.

Tabla N° 3.1.
Matriz de confusión para PG-01

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 37 | 0 |
| No recuperado | 25 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.2.**Matriz de confusión para PG-02**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 41 | 0 |
| No recuperado | 21 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.3.**Matriz de confusión para PG-03**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 42 | 0 |
| No recuperado | 16 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.4.**Matriz de confusión para PG-04**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 24 | 0 |
| No recuperado | 1 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.5.**Matriz de confusión para PG-05**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 35 | 0 |
| No recuperado | 25 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.6.**Matriz de confusión para PG-06**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 19 | 0 |
| No recuperado | 30 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.7.**Matriz de confusión para PG-07**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 26 | 0 |
| No recuperado | 9 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.8.**Matriz de confusión para PG-08**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 24 | 0 |
| No recuperado | 22 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.9.**Matriz de confusión para PG-09**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 22 | 0 |
| No recuperado | 38 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.10.**Matriz de confusión para PG-10**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 17 | 0 |
| No recuperado | 0 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.11.**Matriz de confusión para PG-11**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 14 | 0 |
| No recuperado | 18 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.12.**Matriz de confusión para PG-12**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 36 | 0 |
| No recuperado | 24 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.13.**Matriz de confusión para PG-13**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 15 | 0 |
| No recuperado | 2 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.14.**Matriz de confusión para PG-14**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 25 | 0 |
| No recuperado | 25 | 0 |

Fuente: Elaboración propia, 2017.

Tabla N° 3.15.**Matriz de confusión para PG-15**

| | Relevante | No relevante |
|----------------------|------------------|---------------------|
| Recuperado | 27 | 0 |
| No recuperado | 38 | 0 |

Fuente: Elaboración propia, 2017.

3.4.2.5. Análisis de datos

En esta sección se discuten las formas en que los datos, recopilados en las pruebas de eficacia de recuperación, pueden ser analizados, junto con las principales medidas utilizadas en dichas pruebas. Inicialmente, se deben obtener los valores de precisión y sensibilidad mediante las siguientes fórmulas:

- Fórmula de precisión:

$$P = \frac{TP}{TP + FP}$$

- Fórmula de sensibilidad:

$$R = \frac{TP}{TP + FN}$$

De esa forma, se obtienen los siguientes resultados para cada Proyecto de grado considerado. Es necesario entender que la Precisión de la totalidad de los *queries* ejecutados fue igual al 100% debido a que se consideró que todos los nodos que forman parte de las ontologías, eran relevantes para el desarrollo de los proyectos que representan.

Tabla N° 3.16.
Resultados de precisión y sensibilidad

| Proyecto de grado | Precisión | sensibilidad |
|--------------------------|------------------|---------------------|
| PG-01 | 1 | 0.597 |
| PG-02 | 1 | 0.661 |
| PG-03 | 1 | 0.724 |
| PG-04 | 1 | 0.960 |
| PG-05 | 1 | 0.583 |
| PG-06 | 1 | 0.388 |
| PG-07 | 1 | 0.743 |
| PG-08 | 1 | 0.522 |
| PG-09 | 1 | 0.367 |
| PG-10 | 1 | 1 |
| PG-11 | 1 | 0.438 |
| PG-12 | 1 | 0.600 |
| PG-13 | 1 | 0.882 |
| PG-14 | 1 | 0.500 |
| PG-15 | 1 | 0.415 |

Fuente: Elaboración propia, 2017.

Así, aplicando la fórmula de Valor-F

$$F = \frac{2PR}{P + R}$$

Se obtienen los siguientes resultados para cada Proyecto de grado:

Tabla N° 3.17.**Valor-F para cada Proyecto de grado**

| Proyecto de Grado | Valor-F |
|--------------------------|----------------|
| PG-01 | 0.748 |
| PG-02 | 0.796 |
| PG-03 | 0.840 |
| PG-04 | 0.980 |
| PG-05 | 0.736 |
| PG-06 | 0.559 |
| PG-07 | 0.852 |
| PG-08 | 0.686 |
| PG-09 | 0.537 |
| PG-10 | 1 |
| PG-11 | 0.609 |
| PG-12 | 0.750 |
| PG-13 | 0.937 |
| PG-14 | 0.667 |
| PG-15 | 0.586 |

Fuente: Elaboración propia, 2017.

Mientras el valor esperado del valor-F es 1, para mantener un equilibrio entre precisión y sensibilidad, contar con un valor que represente un valor mayor al 51% implica que la recuperación de conocimiento realizada mediante la ejecución de consultas sirve de soporte para el proceso de gestión de conocimiento personal aplicado a la resolución de problemas mediante proyectos de software, y permite una precisión (relación entre documentos relevantes recuperados y número total de elementos relevantes) y sensibilidad (la habilidad para encontrar el conjunto completo de resultados relevantes de una colección de documentos) suficiente para la

obtención de notas de aprobación en el desarrollo de Proyectos de Grado.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

PRIMERA, se determinaron satisfactoriamente las variables para la gestión de conocimiento personal, mediante el análisis del modelo de Völkel y su impacto en el proceso. Debido a la forma de definición del modelo de Völkel se realizó el ajuste de los procesos del mismo para su utilización con tecnologías semánticas y la herramienta del presente proyecto. Si bien el modelo incluye un proceso secuencial para la gestión de conocimiento personal, se tuvo que definir la manera de indexación de las Señales de conocimiento basados en diferentes estudios y la parametrización del servicio web Open Calais.

SEGUNDA, se diseñó exitosamente el modelo ontológico de la herramienta, mediante la implementación de una arquitectura basada en la web semántica y el uso de ontologías facetadas ligeras. El proceso de selección de un tipo de ontología que se ajuste al modelamiento de conocimiento de diferentes dominios, requirió una amplia revisión documental previa a la selección de ontologías facetadas ligeras. Se consideraron varias opciones para la construcción de la arquitectura semántica, como las ontologías de dominio, lenguajes como OWL y sublenguajes del mismo, servicios para el análisis semántico (que fueron obviados por su falta de soporte en el idioma español), bases de datos orientadas a grafos no nativas y módulos de Drupal con diferentes enfoques y dependencias para su funcionamiento. El patrón de diseño ontológico definido, fue resultado del análisis de las ontologías facetadas ligeras y su aplicación en la herramienta del presente proyecto. Se realizó la búsqueda de alternativas al mismo, que se ajusten a los requerimientos de un enfoque independiente de dominio y finalmente se decidió la construcción de uno propio debido a las escasas opciones proporcionadas por la metodología NeOn.

TERCERA, se modeló el método de búsqueda semántica para la selección de conocimiento, mediante la implementación de interfaces de usuario que permiten la interacción con servicios web para el análisis semántico de los contenidos y mediante el uso de interfaces para la conexión con bases de datos orientadas a grafos. La creación de una interfaz capaz de interactuar con las capas de la arquitectura semántica utilizada requirió del uso de módulos de Drupal en etapa *alpha* y *beta* que ofrecen la funcionalidad mínima requerida por el sistema. Algunos módulos, como es el caso de *Relation* requieren de soluciones alternativas para la indexación de sus campos de

referencia con el módulo *Search API* debido a la falta de soporte de campos de referencia dinámicos, lo que conllevó a la inclusión de campos equivalentes de referenciación con actual soporte de indexación. La búsqueda de temas para la interfaz gráfica resultó en la creación de código extra para su inclusión en los archivos de dependencias. Finalmente, para la selección semántica de conocimiento se adaptó el proceso del sistema denominado *QueryGen* mediante el uso de palabras clave.

CUARTA, Se midió la eficacia de la aplicación de conocimiento personal, mediante el uso de Proyectos de Grado de estudiantes de la carrera de Ingeniería de Sistemas Informáticos de la Universidad Privada del Valle para la creación de ontologías de prueba. La definición de una población para la medición de la eficacia requirió de la búsqueda documental de Proyectos de Grado aprobados y disponibles en la Biblioteca de la Universidad. La falta de versiones digitales de los Proyectos seleccionados implicó la transcripción manual de los contenidos requeridos para las pruebas.

RECOMENDACIONES

PRIMERA, considerar la implementación de la herramienta en la Universidad Privada del Valle para el soporte en las materias de Taller de Programación I, II y Taller de Sistemas Informáticos I, II y III a fin de validar los resultados obtenidos.

SEGUNDA, realizar la actualización de la herramienta cuando se liberen nuevas versiones estables del núcleo de Drupal y los módulos utilizados, con el objetivo de corregir errores e implementar nuevas funcionalidades.

TERCERA, implementar una ruta de actualización desde la implementación de la primera versión beta.

CUARTA, extender el estudio del uso de las ontologías facetadas ligeras para su aplicación en la gestión de conocimiento personal.

QUINTA, simplificar la asignación de variables para la interacción con el servicio web Open Calais y la base de datos Neo4j.

SEXTA, considerar el uso de autómatas para la verificación del contenido ingresado por el usuario, a fin de asegurar gramática y sintaxis.

SÉPTIMA, empaquetar el sistema para proporcionar toda la funcionalidad con una sola instalación.

OCTAVA, considerar la implementación de la herramienta como SaaS (*Software as a service*).

NOVENA, simplificar la creación de *queries* mediante la implementación de búsquedas con lenguaje natural.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

- Ambrose, S., Bridges, M., Lovett, M., DiPietro, M., Norman, M. (2010). *How Learning Works: Seven Research-Based Principles for Smart Teaching* [Cómo funciona el aprendizaje: Siete principios basados en la investigación para una enseñanza inteligente]. (1a Ed.) San Francisco CA, Estados Unidos: Jossey-Bass.
- Antoniou, G., & Van Harmelen, F. (2008). *A semantic web primer, second edition* [Un manual de la web semántica, segunda edición]. MIT press.
- Blanchard, B., Fabrycky, W., & Verma, D. (1994). *Application of the system engineering process to define requirements for computer-based design tools* [Aplicación del proceso de ingeniería de sistemas para definir los requisitos de las herramientas de diseño por ordenador]. Monografía, Society of Logistics Engineers, 8100.
- Eriksson, H. E., Penker, M., Lyons, B., & Fado, D. (2003). *UML 2 toolkit* [Kit de herramientas UML 2]. John Wiley & Sons.
- Gómez-Pérez, A., Motta, E., & Suárez-Figueroa, M. C. (2010). *NeOn Methodology in a Nutshell* [Metodología NeOn en pocas palabras].
- Kapurch, S. J. (2010). *NASA systems engineering handbook* [Manual de ingeniería de sistemas de la NASA]. Diane Publishing.
- Pellegrino, J. W., & Hilton, M. L. (2012). *Education for life and work: developing transferable knowledge and skills in the 21st century* [Educación para la vida y el trabajo: desarrollo de conocimientos y competencias transferibles en el siglo XXI]. Washington, D.C.: The National Academies Press.
- Rosenberg, D., & Scott, K. (2001). *Applying Use Case Driven Object Modeling with UML: An Annotated E-commerce Example* [Aplicación de Use Case Driven Object Modeling con UML: Un ejemplo de comercio electrónico anotado]. Addison-Wesley Professional.
- Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases: new opportunities for connected data* [Bases de datos gráficas: nuevas oportunidades para los datos conectados]. O'Reilly Media, Inc.
- Rosenberg, D., & Stephens, M. (2007). *Use case driven object modeling with UML* [Caso de uso

de modelado basado en objetos con UML]. APress, Berkeley, USA.

Sommerville, I. (2011). *Software engineering* [Ingeniería de Software]. Boston: Pearson.

Van Bruggen, R. (2014). *Learning Neo4j* [Aprendiendo Neo4j]. Packt Publishing Ltd.

Van Vliet, H. (2007). *Software engineering: principles and practice* (Vol. 3) [Ingeniería de software: principios y práctica]. New York: Wiley.

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., & Shortell, T. M. (2015). *INCOSE Systems Engineering Handbook: A guide for system life cycle processes and activities* [Manual de Ingeniería de Sistemas INCOSE: Una guía para los procesos y actividades del ciclo de vida del sistema]. Hoboken, Estados Unidos: Wiley.

DOCUMENTOS

Ackoff, R. L. (1989). *From data to wisdom*. Journal of applied systems analysis, 16(1), 3-9.

Agnihotri, R., & Troutt, M. D. (2009). *The effective use of technology in personal knowledge management: A framework of skills, tools and user context* [El uso efectivo de la tecnología en la gestión del conocimiento personal: Un marco de habilidades, herramientas y contexto de usuario]. Online Information Review, 33(2), 329-342.

Alavi, M. & Leidner, D. (1999). *Knowledge management systems: issues, challenges, and benefits* [Sistemas de gestión del conocimiento: problemas, desafíos y beneficios]. Communications of the AIS, 1(2es), 1.

Amavizca, L., García, A., Jiménez, E., Duarte, G., & Vásquez, J. (2014). *Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software*. Latin American and Caribbean Consortium of Engineering Institutions, México.

Ariñez, J. (2010). *Sistema de promoción de fondos de inversión utilizando dispositivos móviles. Caso: Fortaleza SAFI S.A.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Asociación Española de Normas y Certificación (2003). *Los recursos humanos en un sistema de gestión de la calidad*. Gestión de las competencias. UNE, 66173, 2003.

Babekr, S. T., Fouad, K. M., & Arshad, N. (2013). *Personalized semantic retrieval and summarization of web based documents* [Recuperación semántica personalizada y resumen de documentos basados en la web]. IJACSA) International Journal of Advanced Computer Science

and Applications, 4(1), 177-86.

Barceló, M., Sánchez, G., & Pérez, A. (2006). *La Web Semántica como apoyo a la Gestión del Conocimiento y al Modelado Organizacional*. Revista De Ingeniería Informática, 12.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). *The semantic web* [La web semántica]. Scientific american, 284(5), 28-37.

Bobed, C., & Mena, E. (2016). *QueryGen: Semantic interpretation of keyword queries over heterogeneous information systems* [QueryGen: Interpretación semántica de consultas de palabras clave sobre sistemas de información heterogéneos]. Information Sciences, 329, 412-433.

Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. (1999). *El lenguaje unificado de modelado* (Vol. 1). Addison-Wesley.

Carreón, S. (2011). *Sistema experto para el apoyo de diagnóstico y tratamiento de Infecciones de Transmisión Sexual*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Cerruto, R. (2012). *Sistema para la identificación de huellas dactilares para el estudio dactiloscópico de la unidad especializada "Fuera Especial de Lucha Contra el Crimen (FELCC)"*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Cheong, K. F. (2011). *The roles and values of personal knowledge management* [Las funciones y los valores de la gestión personal del conocimiento]. Southern Cross University.

Cheong, R. & Tsui, E. (2010). *The roles and values of personal knowledge management: an exploratory study* [Los roles y valores de la gestión personal del conocimiento: un estudio exploratorio]. Vine, 40(2), 204-227.

Davenport, T. & Prusak, L. (2000). *Working Knowledge: How Organizations Manage What They Know* [Conocimientos prácticos: Cómo gestionan las organizaciones lo que saben]. HBS Press, Boston.

Ding, Y. (2001). *Ontology: The enabler for the semantic web* [Ontología: El facilitador de la web semántica]. Journal of Information Science, 27(6), 377-384.

Durán, D. (2010). *Sistema de métricas de seguridad con tableros de control integrados para medir la efectividad de los controles de entidades financieras. Caso de Estudio: Banco de Crédito de Bolivia S.A.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Feroz, M. (2010). *Faceted lightweight ontologies: a formalization and some experiments*

[Ontologías facetadas ligeras: una formalización y algunos experimentos]. Doctoral dissertation, University of Trento.

Frاند, J., & Hixon, C. (1999). *Personal Knowledge Management: Who, What, Why, When, Where, How?* [Gestión del Conocimiento Personal: ¿Quién?, ¿qué?, ¿por qué?, ¿cuándo?, ¿dónde?, ¿cómo?].

Gangemi, A., & Presutti, V. (2009). *Ontology design patterns* [Patrones de diseño ontológicos]. En *Handbook on ontologies* (p. 221-243). Springer Berlin Heidelberg.

Giunchiglia, F., Dutta, B., & Maltese, V. (2009). *Faceted lightweight ontologies* [Ontologías facetadas ligeras]. *Conceptual Modeling: Foundations and Applications*, 36-51.

Gómez, A., & Suárez, M. (2009). *NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology* [Metodología NeOn para la creación de redes de ontología: una metodología basada en escenarios].

González, R. (2008). *Aplicación de redes convergentes para grupos empresariales. Caso de estudio: Grupo empresarial Kantutani S.A.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

González, R. (2008). *Guía de implementación de portales colaborativos basados en la administración de contenidos empresariales. Caso de estudio: Pymes de telecomunicaciones.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Grau, X. F., & Segura, M. I. S. (2001). *Desarrollo orientado a objetos con UML*. Facultad de Informática–UPM.

Gruber, T. (1993). *A translation approach to portable ontology specifications* [Un enfoque de traducción de las especificaciones de ontología portátil]. *Knowledge acquisition*, 5(2), 199-220.

Gupta, B., Iyer, L., & Aronson, J. (2000). *Knowledge management: practices and challenges. Industrial management & data systems* [Gestión del conocimiento: prácticas y desafíos. Gestión industrial y sistemas de datos]. 100(1), 17-21.

Gutierrez, A. (2010). *Optimización de una red frame relay integrando voz y datos. Caso de estudio: RUAT.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Hidalgo, Y., & Rodríguez, R. (2013). *La web semántica: una breve revisión*. *Revista Cubana de Ciencias Informáticas*, 7(1), 76-85.

- Hripesak, G., & Rothschild, A. S. (2005). *Agreement, the f-measure, and reliability in information retrieval* [Acuerdo, el valor-f y fiabilidad en la recuperación de información]. *Journal of the American Medical Informatics Association*, 12(3), 296-298.
- Jain, P. (2011). *Personal knowledge management: The foundation of organizational knowledge management* [Gestión personal del conocimiento: La base de la gestión del conocimiento organizacional]. *South African Journal of Libraries and Information Science*, Nro. 77, 1-14.
- Jefferson, T. (2006). *Taking it personally: personal knowledge management* [Tomándolo como algo personal: la gestión del conocimiento personal]. *VINE*, 36(1), 35-37.
- Khan, L., McLeod, D., & Hovy, E. (2004). *Retrieval effectiveness of an ontology-based model for information selection* [Eficacia de recuperación de un modelo basado en ontologías para la selección de información]. *The VLDB Journal—The International Journal on Very Large Data Bases*, 13(1), 71-85.
- King, W. (2009). *Knowledge management and organizational learning* [Gestión del conocimiento y aprendizaje organizacional]. En *Knowledge management and organizational learning* (p. 3-13). Springer US.
- Kupetiene, O. (2015). *KNOWLEDGE MANAGEMENT Course Handbook* [Gestión de conocimiento, manual del curso]. (1a Ed.) Lithuania: SMK University of Applied Social Sciences.
- Lewandowski, D. (2012). *A framework for evaluating the retrieval effectiveness of search engines* [Un marco para evaluar la eficacia de recuperación de los motores de búsqueda]. *Next generation search engines: Advanced models for information retrieval*, 456-479.
- Lewandowski, D. (2015). *Evaluating the retrieval effectiveness of Web search engines using a representative query sample* [Evaluar la eficacia de recuperación de los motores de búsqueda Web utilizando una muestra representativa de consulta]. *Journal of the Association for Information Science and Technology*, 66(9), 1763-1775.
- Lipa, A. (2010). *Gobierno electrónico para el seguimiento de los procesos de saneamiento de tierras. Caso: Instituto Nacional de Reforma Agraria INRA*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.
- Llano, C. (2012). *Aplicación de DataSnap Multi-Tier para el desarrollo de un sistema*

informático que permita realizar el registro de datos del proceso de reclutamiento y selección de personal. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Maedche, A., & Staab, S. (2001). *Ontology learning for the semantic web* [Aprendizaje ontológico para la web semántica]. IEEE Intelligent systems, 16(2), 72-79.

Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999). *Performance measures for information extraction* [Medidas de la ejecución para la extracción de información]. In Proceedings of DARPA broadcast news workshop (pp. 249-252).

Malhotra, M., & Nair, T. G. (2015). *Evolution of knowledge representation and retrieval techniques* [Evolución de las técnicas de representación y recuperación de conocimientos]. International Journal of Intelligent Systems and Applications, 7(7), 18.

Mea, V. D., & Mizzaro, S. (2004). *Measuring retrieval effectiveness: A new proposal and a first experimental validation* [Medición de la efectividad de la recuperación: Una nueva propuesta y una primera validación experimental]. Journal of the Association for Information Science and Technology, 55(6), 530-543.

Mendieta, O. (2005). *Sistema de información de seguimiento de equipos para el departamento de internetworking de la empresa Alpha Systems S.R.L.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Mertins, K., Heisig, P. & Vorbeck, J. 2003, *Knowledge Management: Concepts and Best Practices* [Gestión del Conocimiento: Conceptos y Buenas Prácticas]. 2da Edición, Springer, New York.

Mora, M., & Segarra, V. (2016). *Modelo ontológico para la representación de datos académicos y su publicación con tecnología semántica.* Opción, 32(10).

Nagypál, G. (2005, October). *Improving information retrieval effectiveness by using domain knowledge stored in ontologies* [Mejorando la eficacia de la recuperación de información mediante el uso del conocimiento de dominio almacenado en ontologías]. En OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 780-789). Springer, Berlin, Heidelberg.

Nonaka, I. (2008). *The knowledge-creating company* [La empresa creadora de conocimiento]. Harvard Business Review Press.

- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation* [La empresa creadora de conocimiento: Cómo las empresas japonesas crean la dinámica de la innovación]. Oxford university press.
- Ñauta, F., & Escandón, M. (2011). *Estudio, diseño e implementación de un buscador semántico*. Tesis de grado.
- Owens, D. M., & Khazanchi, D. (2009). *Software quality assurance* [Aseguramiento de la calidad del software]. En Handbook of Research on Technology Project Management, Planning, and Operations (p. 242-260). IGI Global.
- Pauleen, D. (2009). *Personal knowledge management: putting the "person" back into the knowledge equation* [Gestión personal del conocimiento: volver a poner a la "persona" en la ecuación del conocimiento]. Online Information Review, 33(2), 221-224.
- Pauleen, D. & Gorman, G. (2011). *Personal knowledge management: Individual, organizational and social perspectives* [Gestión personal del conocimiento: Perspectivas individuales, organizativas y sociales]. Gower Publishing, Ltd.
- Pereyra, B. (2010). *Sistema de atención médica para la caja de salud CORDES*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.
- Radatz, J., Geraci, A., & Katki, F. (1990). *IEEE standard glossary of software engineering terminology* [Glosario de terminología de ingeniería de software estándar IEEE]. IEEE Std, 610121990 (121990).
- Razmerita, L., Kirchner, K., Sudzina, F. (2009). *Personal knowledge management. The role of Web 2.0 tools for managing knowledge at individual and organisational levels* [Gestión personal del conocimiento. El papel de las herramientas Web 2.0 en la gestión del conocimiento a nivel individual y organizativo]. Online Information Review, Vol. 33, 1021-1039.
- Roussey, C., Pinet, F., Kang, M., & Corcho, O. (2011). *An introduction to ontologies and ontology engineering* [Una introducción a las ontologías y a la ingeniería ontológica]. En Ontologies in Urban Development Projects (pp. 9-38). Springer London.
- Różewski, P., Jankowski, J., Brodka, P. & Michalski, R. (2015). *Knowledge workers' collaborative learning behavior modeling in an organizational social network* [Modelado del comportamiento de aprendizaje colaborativo de los trabajadores del conocimiento en una red

- social organizacional]. *Computers in Human Behavior*, 51, 1248-1260.
- Sánchez, D. (2010). *Integración de una arquitectura Business Intelligence con el modelo Balanced ScoreCard para el control y seguimiento de la planificación estratégica. Caso: SOBOCE S.A.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.
- Serrano, R. (2011). *Desarrollo de un portal web basado en E-salud.* (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.
- Shawar, B. & Al-Sadi, J. (2010). *Learning management systems: Are they knowledge management tools?* [Sistemas de gestión del aprendizaje: ¿Son herramientas de gestión del conocimiento?]. *International Journal of Emerging Technologies in Learning (iJET)*, 5(1), 4-10.
- Suárez, M., Gómez, A., & Fernández, M. (2012). *The NeOn methodology for ontology engineering* [La metodología NeOn para la ingeniería ontológica]. In *Ontology engineering in a networked world* (p. 9-34). Springer Berlin Heidelberg.
- Sutton, M. (2008). *Knowledge citizenship for active informed citizenship* [Conocimiento y ciudadanía para una ciudadanía activa e informada]. *SA Journal of Information Management*, 10(4).
- Takeuchi, H., & Nonaka, I. (2004). *Knowledge creation and dialectics* [Creación de conocimiento y dialéctica]. *Hitotsubashi on knowledge management*, 1-27.
- Thayer, R. H. (2002). *Software system engineering: a tutorial* [Ingeniería de sistemas de software: un tutorial]. 68-73.
- Torres, T. (2003). *¿La mejor estrategia docente? La gestión del conocimiento.* *Educación*, Nro. 32, 9-24.
- Tsui, E. (2002). *Technologies for personal and peer-to-peer (p2p) knowledge management* [Tecnologías para la gestión del conocimiento personal y entre pares (p2p)]. In *CSC Leading Edge Forum Technology Grant Report*.
- Ulrich, D. (1998). *Intellectual capital = competence x commitment* [Capital intelectual = competencia x compromiso]. *Sloan management review*, 39(2), 15.
- Van, D. & Spijkervet, A. (1997). *Knowledge Management: dealing intelligently with knowledge* [Gestión del Conocimiento: el manejo inteligente del conocimiento]. En J. Leibowitz & LC. Wilcox (eds), *Knowledge Management and its integrative element*, CRC Press, Washington pp.

31-59.

Vargas, R. (2008). *Modelo basado en SOA para la integración de Sistemas Informáticos que permita implementar una arquitectura de información*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Vásquez, J. (2010). *Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz*. (Proyecto de grado). Universidad del Valle, La Paz, Bolivia.

Völkel M. (2010). *Personal knowledge models with semantic technologies* [Modelos de conocimiento personal con tecnologías semánticas]. Ph.D. thesis University of the State of Baden-Württemberg, KIT.

Wiig, K. (1997). *Integrating intellectual capital and knowledge management* [Integrar el capital intelectual y la gestión del conocimiento]. Long range planning, 30(3), 399-405.

Wiig, K. (1997). *Knowledge management: Where did it come from and where will it go?* [Gestión del conocimiento: ¿De dónde vino y adónde irá?]. Expert systems with applications, 13(1), 1-14.

Xu, Z., Fu, Y., Mao, J., & Su, D. (2006). *Towards the semantic web: Collaborative tag suggestions* [Hacia la web semántica: Sugerencias de etiquetas colaborativas]. En Collaborative web tagging workshop at WWW2006, Edinburgh, Scotland.

WEB

Belavkin, R. V. (2013) *Lecture 8: Ontologies* [Conferencia 8: Ontologías]. Recuperado el 30 de Septiembre de 2017, de <http://www.eis.mdx.ac.uk/staffpages/rvb/teaching/BIS4410/hand08.pdf>.

Chris, S. (2017). *Knowledge Work & Knowledge Worker* [Trabajo de Conocimiento y Trabajador de Conocimiento]. Recuperado el 13 de Abril de 2017 , de <http://www.managementexchange.com/story/knowledge-work-knowledge-worker>.

Dev.mysql.com (2017). *What is MySQL?* [¿Qué es MySQL?]. Recuperado el 16 de Abril de 2017, de <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>.

Dev.mysql.com (2017). *The Main Features of MySQL* [Las características principales de MySQL]. Recuperado el 16 de Abril de 2017, de

<https://dev.mysql.com/doc/refman/5.7/en/features.html>.

Drupal.org (2017). *Drupal as a Content Management System* [Drupal como sistema de gestión de contenidos]. Recuperado el 15 de Abril de 2017, de https://www.drupal.org/docs/user_guide/en/understanding-drupal.html.

Drupal.org (2017). *Estándares*. Recuperado el 15 de Abril de 2017, de <https://www.drupal.org/8/standards>.

Drupal.org (2017). *Introduction to distributions*. Recuperado el 15 de Abril de 2017, de <https://www.drupal.org/docs/7/distributions/introduction-to-distributions>.

Drupal.org (2017). *Concept: Distributions*. Recuperado el 15 de Abril de 2017, de https://www.drupal.org/docs/user_guide/en/understanding-distributions.html.

EDUCAUSE (2015). *The Next Generation Digital Learning Environment: A report on research* [La Próxima Generación de Entornos de Aprendizaje Digital: Un informe sobre la investigación]. Recuperado el 16 de Abril de 2017, de <https://library.educause.edu/~media/files/library/2016/6/eli3037.pdf>.

Hart, J. (2017). *Top down implementation of social learning doesn't work* [La implementación vertical del aprendizaje social no funciona]. Recuperado el 15 de Abril de 2017, de <http://modernworkplacelearning.com/magazine/top-down-implementation-of-social-learning-doesnt-work/>.

Hartley, D. (2017). *Social Learning: Top Down Versus Bottom Up* [Aprendizaje social: De arriba hacia abajo vs. De abajo hacia arriba]. Recuperado el 15 de Abril de 2017, de <http://www.clomedia.com/2012/09/16/social-learning-top-down-versus-bottom-up/>.

Jarche, H. (2017). *Network Learning: Working Smarter with PKM* [Aprendizaje en red: Trabajar de forma más inteligente con PKM]. Recuperado el 15 de Abril de 2017, de <http://jarche.com/2010/10/network-learning-working-smarter/>.

Jarche, H. (2011). *Personal Knowledge Mastery* [Dominio del Conocimiento Personal]. Recuperado el 10 de Abril de 2017, de <http://jarche.com/pkm>.

Kirsch, D. (2006). *Web 2.0 & Personal Knowledge Management* [Web 2.0 y Gestión del Conocimiento Personal]. Recuperado el 30 de octubre de 2017, de: <http://blogs.ittoolbox.com/km/dr-dan/archives/web-20-personal-knowledge-management-10747>.

- Lanier, D. (2014). *Getting Started with Behat* [Comenzando con Behat]. Recuperado el 30 de Junio de 2017, de: <https://thinkshout.com/blog/2014/10/getting-started-with-behat/>.
- Martin, J. (2006). *The basis of corporate and institutional knowledge management* [La base de la gestión del conocimiento corporativo e institucional]. Recuperado el 13 de Mayo de 2017, de: <http://www.spottedcowpress.ca/KnowledgeManagement/pdfs/06MartinJ.pdf>.
- Neo4j.com. (2017). *The Neo4j Graph Platform* [La plataforma gráfica Neo4j]. Recuperado el 10 de Septiembre de 2017, de <https://neo4j.com/product/>
- Neo4j.com. (2017). *What is a Graph Database?* [¿Qué es una base de datos orientada a grafos?]. Recuperado el 10 de Septiembre de 2017, de <https://neo4j.com/developer/graph-database>
- Obitko, M. (2007). *Semantic Web Architecture* [Arquitectura de la web semántica]. Recuperado el 15 de noviembre de 2017, de: <https://www.obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html>
- OCDE (2005). *The definition and selection of key competencies* [La definición y selección de las competencias clave]. Executive summary. OCDE-USAID. Recuperado el 15 de Agosto de 2017, de: <http://www.deseco.admin.ch/bfs/deseco/en/index/02.html>.
- Php.net (2017). *PHP: What is PHP?*. Recuperado el 17 de Abril de 2017, de <http://php.net/manual/en/intro-what-is.php>.
- Php.net (2017). *PHP: What can PHP do?*. Recuperado el 15 de Abril de 2017, de <http://php.net/manual/en/intro-what-can-do.php>.
- Thomson Reuters. (2017). *Open Calais: API User guide* [Abre Calais: Guía del usuario de la API]. Recuperado el 20 de Septiembre de 2017, de http://www.opencalais.com/wp-content/uploads/folder/ThomsonReutersOpenCalaisAPIUserGuideR11_2.pdf

APÉNDICES

APÉNDICES

APÉNDICE N° 1. ENCUESTA DE DOCENTES

Llene las siguiente preguntas escogiendo sólo una opción a menos que se indique lo contrario. Tome en cuenta su experiencia como docente y trate de escoger la respuesta que más se ajuste a lo percibido por su persona.

1. En su opinión, ¿cuál es la opción que describe mejor el cumplimiento de objetivos de los estudiantes al finalizar el desarrollo de su proyecto de taller?
 - a. Objetivos superados
 - b. Objetivos cumplidos plenamente
 - c. Objetivos parcialmente cumplidos
 - d. Objetivos no cumplidos

2. En las diferentes materias de taller de programación y taller de sistemas informáticos ¿Cuán claros son los alumnos para generar objetivos, definir problemas, alcances y límites?
 - a. Muy buenos
 - b. Buenos
 - c. Regulares
 - d. Malos

3. Cuando se desarrolla un proyecto de taller ¿Utilizan los alumnos las metodologías de desarrollo de software correctamente?
 - a. Completamente
 - b. Parcialmente
 - c. Metodología escogida no se adapta al problema

4. ¿Cuáles cree usted que son las mayores obstáculos, para un estudiante, al realizar un proyecto de taller? Puede escogerse más de una opción.
 - a. Falta de conocimiento de metodologías de desarrollo y conceptos
 - b. Redacción o recolección de información
 - c. Falta de tiempo necesario para el desarrollo del proyecto
 - d. Falta de acceso a computadoras
 - e. Falta de organización
 - f. Todas las anteriores

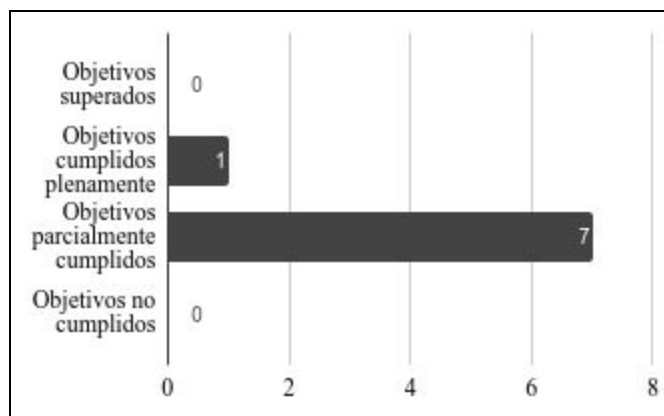
5. ¿Cuál es el grado de influencia que tiene la mala definición de proyectos de taller, por parte de los estudiantes, en la elaboración de proyecto de grado?

- a. Muy alto
 - b. Alto
 - c. Regular
 - d. Ninguno
6. En su opinión ¿Qué grado de importancia tienen las materias de taller de programación y taller de sistemas informáticos en el desarrollo académico de los estudiantes?
- a. Muy alto
 - b. Alto
 - c. Regular
 - d. Ninguno
7. ¿Cree usted que la falta de organización y planificación es un problema habitual entre los estudiantes?
- a. Si
 - b. No
8. ¿Cree usted que todo el conocimiento previo adquirido por los estudiantes es suficiente para una correcta elaboración de Proyectos de Grado?
- a. Si
 - b. No
 - c. Parcialmente
9. ¿Cree usted que los estudiantes son capaces de crear conexiones entre los conceptos aprendidos en las materias cursadas para aplicarlos al análisis y desarrollo de Proyectos de Grado?
- a. Si
 - b. No
 - c. Parcialmente
10. ¿Cree usted que la falta de conocimiento para aplicar conceptos metodológicos y temáticos respecto a un proyecto es un problema recurrente entre los estudiantes?
- a. Si
 - b. No
11. ¿Estaría usted de acuerdo con proporcionar a los estudiantes con herramientas que les permitan gestionar su conocimiento personal?
- a. Si
 - b. No

APÉNDICE N° 2. RESULTADOS DE ENCUESTA

Los resultados de la encuesta anterior se presentan a continuación. Se consultó a ocho docentes de la Carrera de Ingeniería de Sistemas Informáticos de la Universidad Privada del Valle.

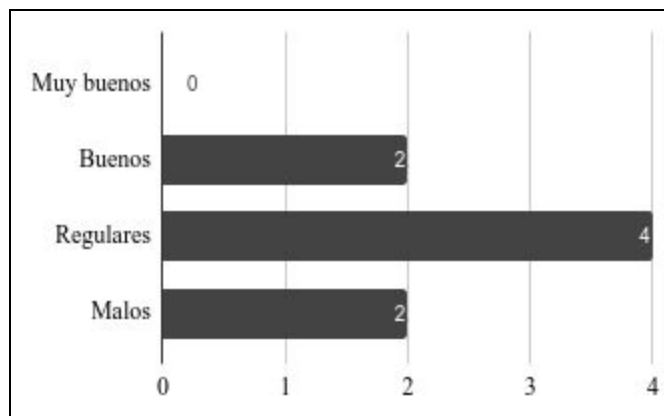
Encuesta Docentes: Resultado pregunta 1



Fuente: Elaboración propia, 2017.

Los resultados de la pregunta 1 denota que los docentes piensan, en su mayoría, que los estudiantes cumplen los objetivos de sus proyectos parcialmente. Además, un docente afirma que los estudiantes cumplen sus objetivos plenamente. Sin embargo, no existen docentes que piensen que los objetivos son superados o no cumplidos.

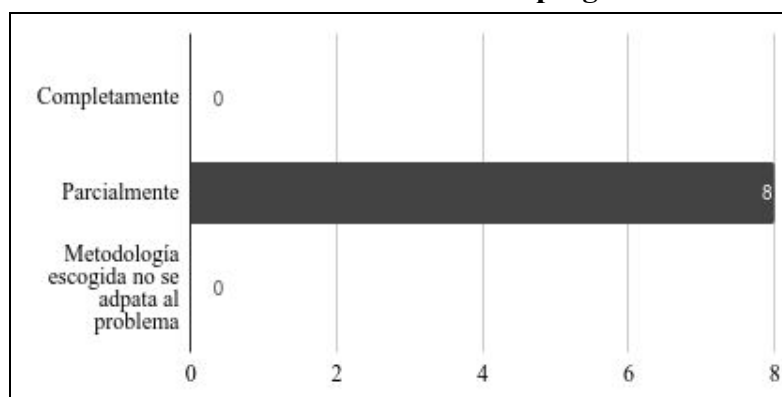
Encuesta Docentes: Resultado pregunta 2



Fuente: Elaboración propia, 2017.

De acuerdo con los resultados de la pregunta 2, se pudo evidenciar que un la mitad de los docentes encuestados creen que los estudiantes son “Regulares” para generar objetivos, definir problemas, alcances y límites. Al mismo tiempo existen una cantidad igual, 2 por cada opción que creen que los estudiantes son buenos para generar los puntos mencionados previamente y que son malos para realizar dicha actividad.

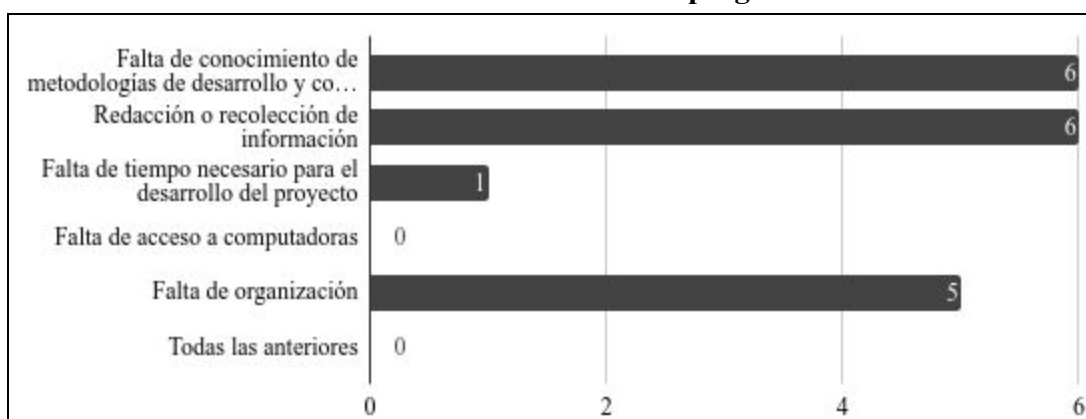
Encuesta Docentes: Resultado pregunta 3



Fuente: Elaboración propia, 2017.

La gráfica anterior indica que todos los docentes encuestados concuerdan en que los estudiantes utilizan parcialmente las diferentes metodologías de desarrollo de software.

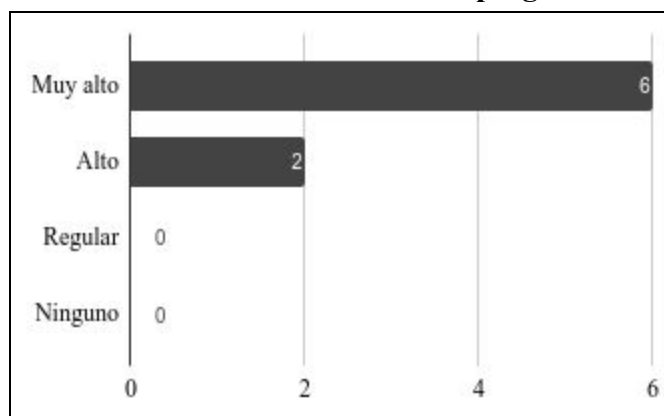
Encuesta Docentes: Resultado pregunta 4



Fuente: Elaboración propia, 2017.

De acuerdo a los resultados de la gráfica anterior se pudo observar que los docentes identifican, como mayores obstáculos, la falta de conocimiento de metodologías de desarrollo, redacción y recolección de información y falta de organización por parte de los estudiantes. Se permitió la selección de múltiples opciones en esta pregunta.

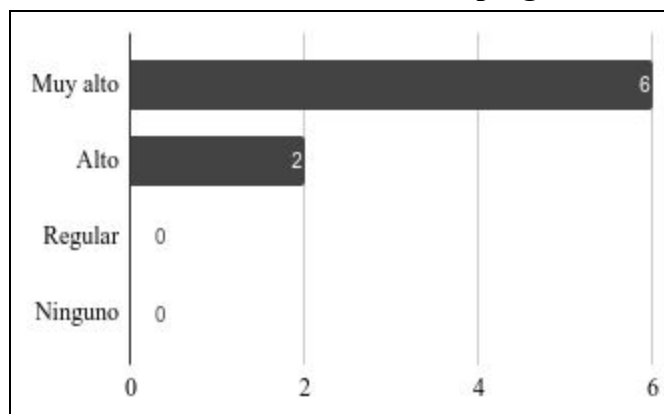
Encuesta Docentes: Resultado pregunta 5



Fuente: Elaboración propia, 2017.

La pregunta anterior hace referencia a la influencia del desarrollo de proyectos de taller en el desarrollo de proyectos de grado, donde se muestra que los docentes indican como “muy alto” y “alto” la influencia de los mismos.

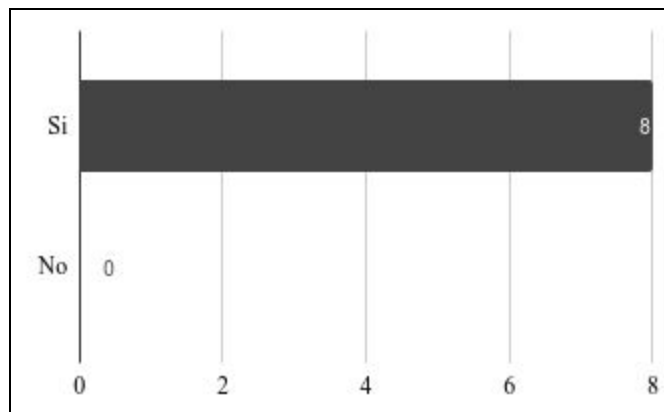
Encuesta Docentes: Resultado pregunta 6



Fuente: Elaboración propia, 2017.

Los docentes, mediante la pregunta anterior, indican que el desarrollo de proyectos de taller tiene una “muy alta” y “alta” importancia en el desarrollo académico de los estudiantes.

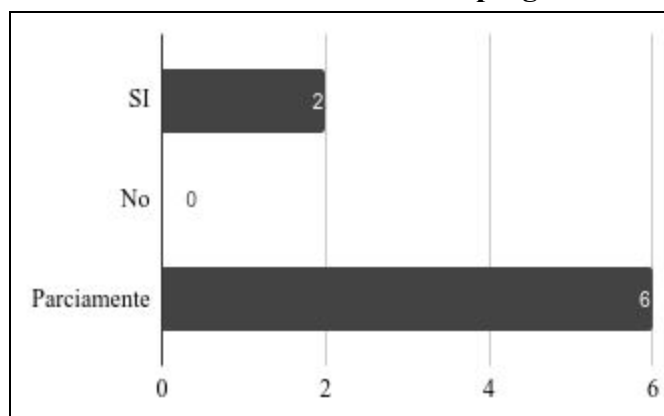
Encuesta Docentes: Resultado pregunta 7



Fuente: Elaboración propia, 2017.

En la pregunta anterior, todos los docentes encuestados concuerdan en afirmar que la falta de organización es un problema recurrente en los estudiantes, el cual, también fue identificado como uno de los mayores problemas para el desarrollo de proyectos de taller.

Encuesta Docentes: Resultado pregunta 8

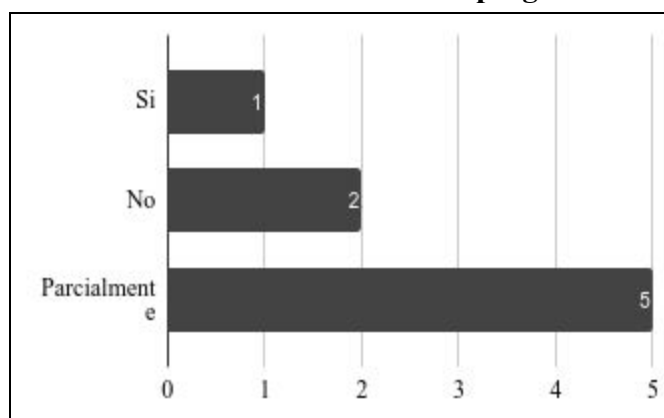


Fuente: Elaboración propia, 2017.

La pregunta anterior denota el grado de suficiencia del conocimiento previo de los estudiantes para la correcta creación de proyectos de grado donde los docentes afirman que es “parcial” y en

menor medida que “sí, es suficiente”.

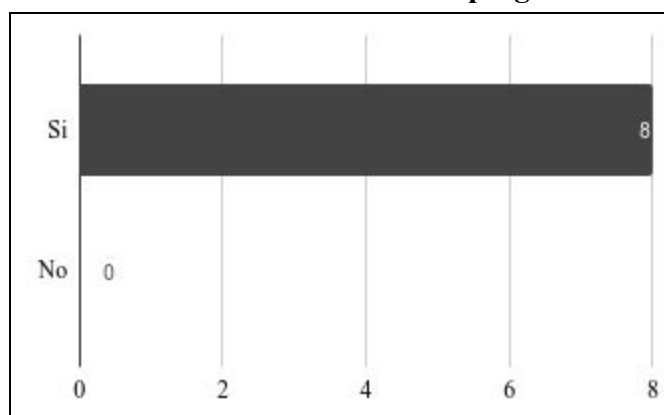
Encuesta Docentes: Resultado pregunta 9



Fuente: Elaboración propia, 2017.

La mayoría de los docentes entrevistados creen que los estudiantes son “parcialmente” capaces de relacionar los conceptos de conocimiento que poseen para ser aplicados en el análisis y desarrollo de proyectos de programación.

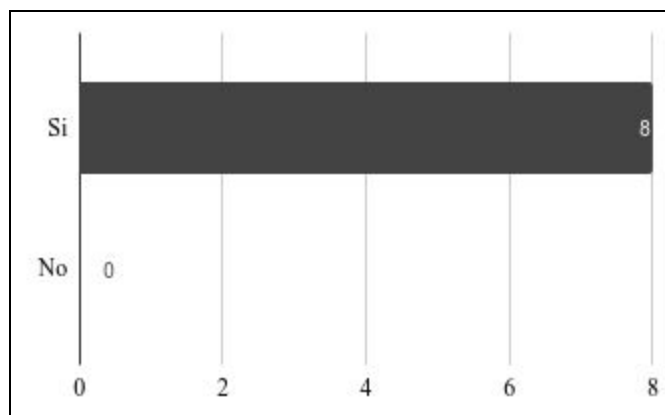
Encuesta Docentes: Resultado pregunta 10



Fuente: Elaboración propia, 2017.

La pregunta anterior demuestra que los docentes piensan que la falta de conocimiento para aplicar conceptos metodológicos y temáticos respecto a un proyecto es un problema recurrente entre los estudiantes.

Encuesta Docentes: Resultado pregunta 11



Fuente: Elaboración propia, 2017.

Con la pregunta anterior los docentes indicaron su respaldo para la creación de una herramienta de gestión de conocimiento personal para los estudiantes con el fin de servir de soporte para la creación de proyectos de taller de programación, taller de Sistemas y proyectos de grado.

APÉNDICE Nº 3. ARCHIVO STUDENT.INFO.YML

```
name: Drupal PKM
type: profile
description: 'Drupal PKM installation profile.'
version: 8.x-1.x-dev
core: 8.x
distribution:
  name: 'Drupal PKM'
dependencies:
  - automated_cron
  - block
  - block_content
  - breakpoint
  - ckeditor
  - color
  - config
  - contextual
  - datetime
  - dblog
  - diff
  - dp_base
  - dp_neo4j_visualizer
  - dynamic_page_cache
  - editor
  - entity_reference_revisions
  - field_ui
  - fieldable_path
  - file
  - help
  - hide_revision_field
  - history
  - image
  - menu_link_content
  - menu_ui
  - neo4j_connector
  - node
  - opencalais_ui
  - options
  - page_cache
  - path
  - pathauto
  - rdf
  - relation
  - selection_note
  - shortcut
  - similarterms
  - simplify
  - taxonomy
  - text
```

- tour
- user
- views
- views_ui
- better_exposed_filters
- ds
- views_infinite_scroll

themes:

- bootstrap

APÉNDICE Nº 4. ARCHIVO STUDENT.LINKS.MENU.YML

```
drupal_pkm.search:
  title: 'Search'
  menu_name: main
  expanded: true
  route_name: system.admin
  weight: 100

drupal_pkm.search_graph:
  title: 'In graph'
  parent: drupal_pkm.search
  route_name: neo4j_visualizer.graph

drupal_pkm.create:
  title: 'Create'
  menu_name: main
  route_name: system.admin_content
  expanded: true
  weight: 101

drupal_pkm.create_knowledge:
  title: 'Knowledge'
  parent: drupal_pkm.create
  route_name: node.add
  route_parameters: { node_type: 'note' }

drupal_pkm.create_taxonomy:
  title: 'Taxonomy term'
  parent: drupal_pkm.create
  route_name: entity.taxonomy_term.add_form
  route_parameters: { taxonomy_vocabulary: 'markup_tags' }

drupal_pkm.create_relation:
  title: 'Relation'
  parent: drupal_pkm.create
  route_name: entity.relation.add_page
  route_parameters: { relation_type: 'intersection' }

drupal_pkm.manage:
  title: 'Manage'
  menu_name: main
  route_name: system.admin
  expanded: true
  weight: 102

drupal_pkm.manage_knowledge:
  title: 'Knowledge'
  parent: drupal_pkm.manage
  route_name: system.admin_content
```

```
drupal_pkm.manage_taxonomy:  
  title: 'Taxonomy term'  
  parent: drupal_pkm.manage  
  route_name: entity.taxonomy_vocabulary.collection
```

```
drupal_pkm.manage_relation:  
  title: 'Relation'  
  parent: drupal_pkm.manage  
  route_name: entity.relation.collection
```

```
drupal_pkm.config:  
  title: 'Configuration'  
  menu_name: main  
  route_name: system.admin  
  expanded: true  
  weight: 103
```

```
drupal_pkm.manage_open_calais:  
  title: 'Open Calais settings'  
  parent: drupal_pkm.config  
  route_name: opencalais_ui.general_settings
```

```
drupal_pkm.manage_neo4j:  
  title: 'Neo4j settings'  
  parent: drupal_pkm.config  
  route_name: neo4j_connector.settings
```

```
drupal_pkm.admin:  
  title: ''  
  description: 'Administration'  
  menu_name: main  
  route_name: system.admin  
  weight: 104  
  options:  
    attributes:  
      data-toggle: 'tooltip'  
      class: 'glyphicon glyphicon-cog'
```

APÉNDICE Nº 5. ARCHIVO SEARCH_APL.INDEX.NEO4J_INDEX.YML

```
status: true
dependencies:
  config:
    - field.storage.node.field_tags
    - field.storage.node.field_description
    - field.storage.relation.field_endpoints
    - field.storage.taxonomy_term.subclassof
    - search_api.server.neo4j
  module:
    - search_api
    - neo4j_connector
    - node
    - relation
    - taxonomy
id: neo4j_index
name: 'Neo4j index'
description: ''
read_only: false
field_settings:
  title:
    label: Title
    datasource_id: 'entity:node'
    property_path: title
    type: string
  nid:
    label: ID
    datasource_id: 'entity:node'
    property_path: nid
    type: integer
  content_created:
    label: 'Authored on'
    datasource_id: 'entity:node'
    property_path: created
    type: date
  field_tags:
    label: Tags
    datasource_id: 'entity:node'
    property_path: field_tags
    type: integer
    dependencies:
      config:
        - field.storage.node.field_tags
description:
  label: 'Description » Processed text'
  datasource_id: 'entity:node'
  property_path: 'field_description:processed'
  type: string
  dependencies:
```

```

    config:
      - field.storage.node.field_description
  relation_type:
    label: 'Relation type'
    datasource_id: 'entity:relation'
    property_path: relation_type
    type: string
  relation_id:
    label: ID
    datasource_id: 'entity:relation'
    property_path: relation_id
    type: integer
  field_endpoints:
    label: Endpoints
    datasource_id: 'entity:relation'
    property_path: field_endpoints
    type: integer
    dependencies:
      config:
        - field.storage.relation.field_endpoints
  relation_created:
    label: Created
    datasource_id: 'entity:relation'
    property_path: created
    type: date
  vid:
    label: Vocabulary
    datasource_id: 'entity:taxonomy_term'
    property_path: vid
    type: string
  tid:
    label: 'Term ID'
    datasource_id: 'entity:taxonomy_term'
    property_path: tid
    type: integer
  subclassof:
    label: subclassOf
    datasource_id: 'entity:taxonomy_term'
    property_path: subclassof
    type: integer
    dependencies:
      config:
        - field.storage.taxonomy_term.subclassof
  name:
    label: Name
    datasource_id: 'entity:taxonomy_term'
    property_path: name
    type: string
  datasource_settings:
    'entity:node':

```

```
  bundles:
    default: true
    selected: { }
  languages:
    default: true
    selected: { }
  'entity:relation':
    bundles:
      default: true
      selected: { }
  'entity:taxonomy_term':
    bundles:
      default: true
      selected: { }
    languages:
      default: true
      selected: { }
processor_settings:
  neo4j_connector_mapping_processor:
    field_mapping:
      title: ''
      nid: ''
      content_created: ''
      field_tags: taxonomy_term
      description: ''
      relation_type: ''
      relation_id: ''
      field_endpoints: node
      relation_created: ''
      vid: ''
      tid: ''
      subclassof: taxonomy_term
      name: ''
    fields: { }
    weights:
      preprocess_index: 0
  add_url: { }
  aggregated_field: { }
  rendered_item: { }
tracker_settings:
  default: { }
options:
  index_directly: true
  cron_limit: 50
server: neo4j
```

APÉNDICE Nº 6. ARCHIVO COMPOSER.JSON

```
{
  "name": "drupal-composer/drupal-project",
  "description": "Project template for Drupal 8 projects with composer",
  "type": "project",
  "license": "GPL-2.0+",
  "authors": [
    {
      "name": "",
      "role": ""
    }
  ],
  "repositories": [
    {
      "type": "composer",
      "url": "https://packages.drupal.org/8"
    }
  ],
  "require": {
    "composer/installers": "^1.2",
    "cweagans/composer-patches": "^1.6",
    "drupal-composer/drupal-scaffold": "^2.2",
    "drupal/better_exposed_filters": "^3.0@alpha",
    "drupal/bootstrap": "^3.9",
    "drupal/console": "~1.0",
    "drupal/core": "^8.0",
    "drupal/diff": "1.x-dev",
    "drupal/drupal-extension": "^3.2",
    "drupal/ds": "^3.1",
    "drupal/entity_reference_revisions": "^1.3",
    "drupal/fieldable_path": "1.x-dev",
    "drupal/hide_revision_field": "^1.1",
    "drupal/neo4j_connector": "2.x-dev",
    "drupal/opencalais_ui": "^1.0",
    "drupal/pathauto": "^1.0",
    "drupal/relation": "2.x-dev",
    "drupal/search_api": "^1.4",
    "drupal/selection_note": "^1.0",
    "drupal/similarterms": "1.3",
    "drupal/simplify": "^1.0",
    "drupal/views_infinite_scroll": "^1.5",
    "drush/drush": "~8.0",
    "webflo/drupal-finder": "^1.0",
    "webmozart/path-util": "^2.3"
  },
  "require-dev": {
    "behat/mink": "~1.7",
    "behat/mink-goutte-driver": "~1.2",
    "jcalderonzumba/gastonjs": "~1.0.2",
  }
}
```

```

        "jcalderonzumba/mink-phantomjs-driver": "~0.3.1",
        "mikey179/vfsstream": "~1.2",
        "phpunit/phpunit": ">=4.8.28 <5",
        "symfony/css-selector": "~2.8"
    },
    "conflict": {
        "drupal/drupal": "*"
    },
    "minimum-stability": "dev",
    "prefer-stable": true,
    "config": {
        "sort-packages": true
    },
    "autoload": {
        "classmap": [
            "scripts/composer/ScriptHandler.php"
        ]
    },
    "scripts": {
        "drupal-scaffold": "DrupalComposer\\DrupalScaffold\\Plugin::scaffold",
        "pre-install-cmd": [
            "DrupalProject\\composer\\ScriptHandler::checkComposerVersion"
        ],
        "pre-update-cmd": [
            "DrupalProject\\composer\\ScriptHandler::checkComposerVersion"
        ],
        "post-install-cmd": [
            "DrupalProject\\composer\\ScriptHandler::createRequiredFiles"
        ],
        "post-update-cmd": [
            "DrupalProject\\composer\\ScriptHandler::createRequiredFiles"
        ]
    },
    "extra": {
        "installer-paths": {
            "web/core": ["type:drupal-core"],
            "web/libraries/{$name}": ["type:drupal-library"],
            "web/modules/contrib/{$name}": ["type:drupal-module"],
            "web/profiles/contrib/{$name}": ["type:drupal-profile"],
            "web/themes/contrib/{$name}": ["type:drupal-theme"],
            "drush/contrib/{$name}": ["type:drupal-drush"]
        }
    }
}

```

APÉNDICE N° 7. DOCUMENTOS DE ESPECIFICACIÓN DE REQUERIMIENTOS ONTOLÓGICOS

**Sistema de promoción de fondos de inversión utilizando dispositivos móviles. Caso:
Fortaleza SAFI S.A.**

| Documento de especificación de requisitos ontológicos | |
|---|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Sistema de promoción de fondos de inversión utilizando dispositivos móviles. Caso: Fortaleza SAFI S.A. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema de promoción de fondos de inversión utilizando dispositivos móviles. Caso: Fortaleza SAFI S.A. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none">- La ontología será desarrollada como un ontología liviana de dominio.- La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Diseñar y desarrollar un sistema de promoción de Fondos de Inversión utilizando dispositivos móviles, destinado a optimizar el desempeño de los Promotores de Inversión de la empresa FORTALEZA SAFI S.A. |

Fuente: Elaboración propia, basado en Ariñez, 2010.

Aplicación de redes convergentes para grupos empresariales. Caso de estudio: Grupo empresarial Kantutani S.A.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Aplicación de redes convergentes para grupos empresariales. Caso de estudio: Grupo empresarial Kantutani S.A. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Aplicación de redes convergentes para grupos empresariales. Caso de estudio: Grupo empresarial Kantutani S.A. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Diseñar una red convergente para grupos empresariales multi-servicios utilizando la tecnología de comunicaciones existente en Bolivia, para integrar los servicios de voz, datos y video, tomando en cuenta la calidad de servicio y seguridad para lograr una red escalable y flexible para reducir el costo administrativo de la red. |

Fuente: Elaboración propia, basado en González, 2008.

Guía de implementación de portales colaborativos basados en la administración de contenidos empresariales. Caso de estudio: Pymes de telecomunicaciones.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Guía de implementación de portales colaborativos basados en la administración de contenidos empresariales. Caso de estudio: Pymes de telecomunicaciones. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Guía de implementación de portales colaborativos basados en la administración de contenidos empresariales. Caso de estudio: Pymes de telecomunicaciones. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Formalizar y elaborar una guía de implementación de portales colaborativos basados en la adquisición administración de contenidos empresariales para contribuir a integrar los servicios que un ambiente colaborativo requiere en Pymes de telecomunicaciones. |

Fuente: Elaboración propia, basado en González, 2008.

Optimización de una red Frame Relay integrando voz y datos. Caso de estudio: RUAT

| Documento de especificación de requisitos ontológicos | |
|---|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Optimización de una red frame relay integrando voz y datos. Caso de estudio: RUAT. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Optimización de una red frame relay integrando voz y datos. Caso de estudio: RUAT. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none">- La ontología será desarrollada como un ontología liviana de dominio.- La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Realiza la optimización y ampliación de servicios de la red Frame Relay del RUAT haciendo énfasis en la tecnología de VoFR para lograr satisfacer las necesidades de comunicación de voz y datos de la organización bajo los criterios de seguridad, fiabilidad y costos. |

Fuente: Elaboración propia, basado en Gutierrez, 2010.

**Gobierno electrónico para el seguimiento de los procesos de saneamiento de tierras. Caso:
Instituto Nacional de Reforma Agraria INRA**

| Documento de especificación de requisitos ontológicos | |
|--|--|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Gobierno electrónico para el seguimiento de los procesos de saneamiento de tierras. Caso: Instituto Nacional de Reforma Agraria INRA |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Gobierno electrónico para el seguimiento de los procesos de saneamiento de tierras. Caso: Instituto Nacional de Reforma Agraria INRA |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar e implementar una aplicación de Gobierno Electrónico para un control adecuado, ágil y segura tanto geográfica y alfanumérica, para coadyuvar a la toma de decisiones en la institución, las organizaciones sociales o sectoriales y quienes sean partes interesadas. |

Fuente: Elaboración propia, basado en Lipa, 2010.

Aplicación de DataSnap Multi - Tier para el desarrollo de un sistema informático que permita realizar el registro de datos del proceso de reclutamiento y selección de personal.

| Documento de especificación de requisitos ontológicos | |
|--|--|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Aplicación de DataSnap Multi - Tier para el desarrollo de un sistema informático que permita realizar el registro de datos del proceso de reclutamiento y selección de personal. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Aplicación de DataSnap Multi - Tier para el desarrollo de un sistema informático que permita realizar el registro de datos del proceso de reclutamiento y selección de personal. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un sistema informático para el registro de datos personales del funcionario de acuerdo al proceso de reclutamiento y selección de personal administrado por el encargado de RR.HH. de la Autoridad de Agua Potable y Saneamiento Básico, aplicando DataSnap como tecnología Multicapa (Multi-Tier) para el desarrollo de aplicaciones de base de datos de gestión. |

Fuente: Elaboración propia, basado en Llano, 2012.

**Sistema de información de seguimiento de equipos para el departamento de
internetworking de la empresa Alpha Systems S.R.L.**

| Documento de especificación de requisitos ontológicos | |
|--|--|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Sistema de información de seguimiento de equipos para el departamento de internetworking de la empresa Alpha Systems S.R.L. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema de información de seguimiento de equipos para el departamento de internetworking de la empresa Alpha Systems S.R.L. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un sistema de información automatizado para el departamento de internetworking de la empresa, que permita llevar un registro oportuno de las tareas de cotizaciones, pedidos, préstamos, RMA y pre-ventas. |

Fuente: Elaboración propia, basado en Mendieta, 2005.

Sistema de atención médica para la caja de salud CORDES.

| Documento de especificación de requisitos ontológicos | |
|---|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Sistema de atención médica para la caja de salud CORDES. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema de atención médica para la caja de salud CORDES. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none">- La ontología será desarrollada como un ontología liviana de dominio.- La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar e implementar el sistema Web de atención médica para la caja de salud CORDES en las sucursales La Paz y El Alto. |

Fuente: Elaboración propia, basado en Pereyra, 2010.

Desarrollo de un portal web basado en E-salud.

| Documento de especificación de requisitos ontológicos | |
|---|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Desarrollo de un portal web basado en E-salud. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Desarrollo de un portal web basado en E-salud. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none">- La ontología será desarrollada como un ontología liviana de dominio.- La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un Portal Web basado en E-Salud con la finalidad de mejorar el servicio médico de los Hospitales, mediante la administración adecuada de la información clínica de los pacientes, un diagnóstico preciso a distancia, la difusión de información orientada al ciudadano y especialista. |

Fuente: Elaboración propia, basado en Serrano, 2011.

Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un sistema informático integrado de los procesos de Afiliación, Atención Ambulatoria, Fichaje y Farmacia como propuesta para la Caja Petrolera de Salud Regional La Paz a fin de que se logre satisfacer las necesidades de la CPS y de sus asegurados. |

Fuente: Elaboración propia, basado en Vásquez, 2010.

Sistema de métricas de seguridad con tableros de control integrados para medir la efectividad de los controles de entidades financieras. Caso: Banco de Crédito de Bolivia S.A.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema de métricas de seguridad con tableros de control integrados para medir la efectividad de los controles de entidades financieras. Caso: Banco de Crédito de Bolivia S.A. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Diseñar un sistema de métricas de seguridad con tableros de control integrados para cuantificar los controles de seguridad, su efectividad e identificar posibles acciones de mejora en entidades financieras. |

Fuente: Elaboración propia, basado en Durán, 2010.

Modelo basado en SOA para la integración de Sistemas Informáticos que permita implementar una arquitectura de información.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Modelo basado en SOA para la integración de Sistemas Informáticos que permita implementar una arquitectura de información. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un modelo de Integración de Sistemas basados en SOA, para integrar funciones o servicios, con el propósito de obtener un mejor acceso a la información a través de una arquitectura de información normalizada. |

Fuente: Elaboración propia, basado en Vargas, 2008.

Sistema experto para el apoyo de diagnóstico y tratamiento de Infecciones de Transmisión Sexual.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema experto para el apoyo de diagnóstico y tratamiento de infecciones de transmisión sexual. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un sistema basado en lógica difusa que determine mediante los signos y síntomas de las infecciones de Transmisión Sexual, posibles soluciones que apoyen el diagnóstico y tratamiento médico. |

Fuente: Elaboración propia, basado en Carreón, 2011.

Sistema para la identificación de huellas dactilares para el estudio dactiloscópico de la unidad especializada “Fuera Especial de Lucha Contra el Crimen (FELCC)”.

| Documento de especificación de requisitos ontológicos | |
|--|---|
| 1 | Propósito |
| | Representa el conocimiento referente al área de Ingeniería de Sistemas para su aplicación en el Proyecto de Grado: Propuesta de un sistema informático integrado de los procesos de afiliación, atención ambulatoria, farmacia y fichaje. Caso: Caja Petrolera de Salud, Regional La Paz. |
| 2 | Alcance |
| | La ontología incluirá el contenido definido por el autor del proyecto, en los capítulos: Marco Teórico y Marco Referencial. |
| 3 | Lenguaje de implementación |
| | La ontología seguirá las líneas de diseño de las ontologías facetadas ligeras. |
| 4 | Usuarios finales previstos |
| | Autor del Proyecto: Sistema para la identificación de huellas dactilares para el estudio dactiloscópico de la unidad especializada “Fuera Especial de Lucha Contra el Crimen (FELCC)”. |
| 5 | Usos previstos |
| | Pruebas de eficacia para la medición de selección de conocimiento de la herramienta propuesta en el presente proyecto. |
| 6 | Requisitos |
| | a. Requisitos no funcionales |
| | <ul style="list-style-type: none"> - La ontología será desarrollada como un ontología liviana de dominio. - La ontología podrá contener elementos en los idiomas Español e Inglés. |
| | b. Requisitos funcionales |
| | Desarrollar un sistema para la automatización de Identificación de Huellas Dactilares para el estudio Dactiloscópico de la Unidad Especializada “Fuerza Especial de Lucha Contra el Crimen (FELCC)”, para reducir el tiempo en la identificación y control de las personas. |

Fuente: Elaboración propia, basado en Cerruto, 2012.

ANEXOS

ANEXOS

ANEXO N° 1. ENTIDADES OPEN CALAIS

| | | |
|-------------------------|--------------------|---------------------|
| Anniversary | MedicalCondition | ProgrammingLanguage |
| City | MedicalTreatment | ProvinceOrState |
| Company | Movie | PublishedMedium |
| Continent | MusicAlbum | RadioProgram |
| Country | MusicGroup | RadioStation |
| Editor | NaturalFeature | Region |
| EmailAddress | OperatingSystem | SportsEvent |
| EntertainmentAwardEvent | Organization | SportsGame |
| Facility | Person | SportsLeague |
| FaxNumber | PharmaceuticalDrug | Technology |
| Holiday | PhoneNumber | TVShow |
| IndustryTerm | PoliticalEvent | TVStation |
| Journalist | Position | URL |
| MarketIndex | Product | |