

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Almacenamiento y manipulación de ontologías que describen
patrones de análisis y su semántica**

presentada por

Lic. Guadalupe Candia Zavaleta

como requisito para la obtención del grado de
Maestra en Ciencias de la Computación

Director de tesis

Dr. Moisés González García

Cuernavaca, Morelos, México. Junio de 2017



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

Cuernavaca, Mor., a 31 de mayo del 2017
OFICIO No. DCC/024/2017

Asunto: Aceptación de documento de tesis

DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **Lic. Guadalupe Candia Zavaleta**, con número de control M14CE056, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Almacenamiento y manipulación de ontologías que describen patrones de análisis y su semántica**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Moisés González García
Doctor en Ciencias en la Especialidad
de Ingeniería Eléctrica
7501724

REVISOR 1

Dra. Olivia Graciela Fragozo Díaz
Doctora en Ciencias en Ciencias de
la Computación
7420199

REVISOR 2

Dr. Joaquín Pérez Ortega
Doctor en Ciencias
Computacionales
4795984

REVISOR 3

Dr. René Santaolaya Salgado
Doctor en Ciencias de la
Computación
4454821

C.p. M.T.J. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

NACS/miz





"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"

Cuernavaca, Mor., 5 de junio de 2017
OFICIO No. SAC/212/2017

Asunto: Autorización de impresión de tesis

**LIC. GUADALUPE CANDIA ZAVALTA
CANDIDATA AL GRADO DE MAESTRA EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**Almacenamiento y manipulación de ontologías que describen patrones de análisis y su semántica**", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

Cp: M.T.I. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr



Dedicatoria

A mi madre:

Martha Zavaleta Honorato

Por el apoyo que me brindo desde el momento que le manifesté mi deseo por estudiar una maestría hasta su culminación. Por todos esos días, que ha tenido que levantarse de la cama de madrugada y las noches de desvelo que ha tenido por estar trabajando, para darnos estudio a mis hermanos y a mí. Por regalarme su cariño, ser mi confidente y por todos los momentos felices que hemos pasado juntas.

Envidio tu fortaleza, coraje, inteligencia y admiro tus aptitudes de emprendedora.

Te amo mamá.

Agradecimientos

A mi padre Tomás Candia González por los momentos felices que hemos pasado en familia....te quiero.

A mis hermanos Josefa, Gabriela, Rafael, Leticia, Virginia, Carmen y Margarita por las adversidades y dichas que hemos vivido juntos, y el apoyo que me han dado cada uno de diferente manera. Espero que al pasar de los años sigamos siendo el equipo que somos hasta este momento. Los quiero a todos.

A mi tía y madrina Ana María Zavaleta Honorato por su apoyo y oraciones.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico recibido, fue esencial para realizar mis estudios de maestría.

A la universidad Leland Stanford Junior por proporcionarnos el código del sistema NCBO BioPortal.

Al área de Ingeniería de Software ubicada en el Departamento de Ciencias Computacionales perteneciente al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por darme la oportunidad de obtener un grado más en mi vida como estudiante.

Al director de mi tema de tesis, el Dr., Moisés González García por su orientación y paciencia durante el proyecto de investigación.

A los revisores de mi tema de tesis: Dra. Olivia Graciela Fragoso Díaz, Dr. Joaquín Pérez Ortega y Dr. René Santaolaya Salgado por las críticas constructivas realizadas durante la investigación y el tiempo que le dedicaron en revisar cada uno de los documentos desarrollados durante la investigación.

A la Licenciada Guadalupe Patricia Armas León por creer en mí, cuando hasta ese momento muchas personas lo dudaban, gracias.

A la Dra. Olivia Fragoso Díaz por conducirme hacia la persona que me ayudo a salir de la crisis emocional por la que atravesaba al inicio de mis estudios de maestría, por su apoyo y por sus palabras de aliento para continuar en la maestría, gracias.

A la Maestra Verónica Sotelo Boyás por escucharme durante un poco más de dos años y motivarme a vencer mis miedos. Fue de gran ayuda durante mi estancia en CENIDET, encontré en ella una amiga y confidente.

Al Dr. René Santaolaya Salgado por su amabilidad y por creer en mi capacidad para realizar actividades referentes al área de computación.

Al Jhon Graybeal, director del programa técnico Repositorio BioPortal en la escuela de medicina de la Universidad de Stanford. Por la disposición, que tuvo al orientarme con la utilización de la aplicación NCBO BioPortal.

Al Dr. José Arturo Olvera López profesor-investigador de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, por la disponibilidad que tuvo para facilitarme la carta de recomendación, que me solicitaron para ingresar a estudiar la maestría.

A la secretaria del Departamento de Ciencias Computacionales Liliana Mérida Zagal por su amabilidad y disponibilidad para orientarme en el proceso de tramites dentro del instituto.

A todos los compañeros que tuve en CENIDET y amigos, que me apoyaron, cada uno de manera diferente. No escribo los nombres por no omitir alguno.

Resumen

Los patrones de análisis son recursos poco utilizados. La razón de esta escasa utilización se relaciona con la falta de conceptos de estabilidad, así como de semántica para su descripción. Lo anterior ocasiona que se presenten dificultades para su aplicación en dominios diferentes.

En esta investigación se propone una estructura de información para almacenar descripciones de patrones de análisis, respetando una jerarquía de ontologías compuesta por tres niveles de abstracción que incluyan los conceptos de estabilidad y semántica, en un repositorio que estará disponible en un servicio web. Con base en la estructura de información propuesta, se especificaron los requerimientos funcionales para la construcción de un software que permita que el usuario final, tenga acceso al repositorio de patrones de análisis a través de una interfaz web.

Se utilizó la aplicación NCBO BioPortal proporcionada por la Universidad de Stanford para adaptar algunos de los requerimientos funcionales especificados en esta investigación. Con el fin de validar dichos requerimientos, se realizaron pruebas aplicando el 829-2008 IEEE Standard for Software and System Test Documentation. Al usar la aplicación NCBO BioPortal se obtuvo un repositorio de descripciones de patrones de análisis.

Disponer de un repositorio de ontologías de patrones de análisis en la web ayudará a los expertos en el proceso de un dominio, así como a personas menos expertas a compartir información. Asimismo, favorecerá la proliferación de patrones de análisis. Ayudará a personas menos expertas a comprender los procesos del dominio que les corresponda, en el caso de que el repositorio cuente con un patrón de análisis correspondiente al dominio requerido.

Abstract

The patterns of analysis are underutilized resources. The reason for this low utilization is related to the lack of concepts of stability and the semantics for its description. This causes difficulties to be applied in different domains.

This research proposes an information structure to store descriptions of patterns of analysis, respecting a hierarchy of ontologies composed of three levels of abstraction that includes the concepts of stability and semantics, in a repository that will be available in a web service. Based on the proposed information structure, the functional requirements were specified for the construction of software that allows the end user to access the repository of analysis patterns through a web interface.

The NCBO BioPortal application provided by Stanford University was used to adapt some of the functional requirements specified in this research. In order to validate these requirements, tests were carried out applying the IEEE 829-2008 Standard for Software and System Test Documentation. Using the NCBO BioPortal application, a repository of analysis pattern descriptions was obtained.

Having an ontology repository of analysis patterns on the web will help experts in the process of a domain and less experienced people to share information. Likewise, it will favor the proliferation of analysis patterns. It will help less experienced people to understand the processes of the domain that corresponds to them in the case that the repository has an analysis pattern corresponding to the required domain.

Contenido

Lista de figuras	iv
Lista de tablas.....	vi
Abreviaturas y conceptos	vii
Capítulo 1. Introducción	1
1.1 Descripción del problema	3
1.2 Objetivo	3
1.3 Justificación	3
1.4 Organización del documento	3
Capítulo 2. Marco Teórico.....	5
2.1 Patrones de análisis	7
2.2 Modelo de Estabilidad de Software (SSM)	9
2.3 Ontología	11
2.4 Lenguaje de Ontologías Web (OWL).....	11
2.5 Notación de requerimientos de Usuario.....	12
Capítulo 3. Estado del Arte.....	15
3.1 Antecedentes	17
3.2 Trabajos relacionados	22
Capítulo 4. Requerimientos y adaptación de los requerimientos con BioPortal	31
4.1 Requerimientos funcionales.....	33
4.1.1 Requerimientos funcionales (diagramas).....	33
4.1.2 Especificación textual de requerimientos funcionales.	43
4.1.3 Requerimientos no funcionales	45
4.2. Adaptación de los requerimientos con Bioportal.....	46
4.2.2 Forma de satisfacer los requerimientos funcionales utilizando la aplicación BioPortal.....	46
Capítulo 5. Descripción del desarrollo	53
5.1 Propuesta de solución	54
5.1.1 Cardinalidad de las relaciones que se observan en la Figura 5.1	55

5.2 Definición de las categorías	56
5.3 Crear cuenta de usuario.....	58
5.4 Agregar una ontología al portal web BioPortal	59
5.5 Agregar una descripción de PA, mediante un individuo en la categoría I-I Descripciones de PA.....	64
5.6 Agregar una ontología o individuo en alguna de las categorías existentes.....	68
5.7 Creación de categorías en BioPortal.....	68
5.7.1 Ingresar a la consola de Ruby en el servidor de BioPortal.....	68
5.7.2 Crear categorías en BioPortal	70
5.8 Eliminar categorías	72
5.9 Crear grupos.....	73
5.10 Nombre y acrónimos de las categorías	74
5.11 Eliminar ontologías.....	74
5.12 Enviar correos electrónicos desde Feedback del portal de BioPortal	77
Capítulo 6. Resultados y pruebas	81
6.1 Resultados generales.....	83
6.1.1 Resultados detallados	83
6.2 Detalles del diseño de la prueba	84
6.2.1 Características a probar	84
6.2.2 Caso de Prueba TC-01.....	86
6.2.3 Caso de Prueba TC-02.....	87
6.2.4 Caso de Prueba TC-03.....	93
6.2.5 Caso de Prueba TC-04.....	95
6.2.6 Caso de Prueba TC-05.....	97
6.2.7 Caso de Prueba TC-06.....	99
6.2.8 Caso de Prueba TC-07.....	106
Capítulo 7. Conclusiones.....	109
5.1 Conclusiones	111
5.2 Trabajos Futuros.....	112
Referencias	113

Anexo A. Descripción técnica de BioPortal.....	117
Anexo B. Instalación de BioPortal	135
Anexo C. Plan de pruebas.....	147
Anexo D. Diseño de pruebas	155
Anexo E. Procedimiento de pruebas	161

Lista de figuras

Figura 2. 1 Arquitectura de Modelos de Estabilidad de Software.	9
Figura 2. 2 Construcciones visuales de GRL en la herramienta JUCMNav v7.0.0.	13
Figura 2. 3 Construcciones visuales de UCM de la herramienta JUCMNav v7.0.0.	13
Figura 2. 4 GRL, UCM y otras piezas de modelado, según (Amyot, 2003).	14
Figura 3. 1 Representación del Sistema de Almacenaje y Utilización de Patrones de Análisis (AUPAs).	20
Figura 3. 2 Infraestructura para la reutilización de patrones de análisis (APRI).	23
Figura 3. 3 Arquitectura de la herramienta ArgoCASEGEO.	24
Figura 3. 4 Enfoque del diseño de bases de datos semántica.	26
Figura 3. 5 Panorama del enfoque propuesto.	27
Figura 4. 1 Diagrama GRL de los requerimientos funcionales del almacén de información.	33
Figura 4. 2 Diagrama UCM meta 1.Crear cuentas de usuario del diagrama GRL.	35
Figura 4. 3 Diagrama UCM meta 2.Representar jerarquía de ontologías del diagrama GRL.	36
Figura 4. 4 Diagramas UCM meta 3. Realizar consultas al repositorio de ontologías del diagrama GRL.	37
Figura 4. 5 Diagrama UCM meta 4.Guardar diagramas XMI del diagrama GRL.	38
Figura 4. 6 Diagrama UCM meta 5.Descargar archivos del diagrama GRL.	39
Figura 4. 7 Diagrama UCM meta 6.Aportaciones de ontologías por usuarios registrados del diagrama GRL.	40
Figura 4. 8 Diagrama UCM meta 7.Actualizar información de cada ontología del diagrama GRL.	41
Figura 4. 9 Diagrama UCM meta 8.Eliminar ontologías del diagrama GRL.	42
Figura 4. 10 Registrar usuarios.	47
Figura 4. 11 Utilizar la sección de categorías para representar la jerarquía de ontologías.	48
Figura 4. 12 Buscar una clase en múltiples ontologías, funcionalidad de BioPortal.	49
Figura 4. 13 Descargar archivos OWL en BioPortal.	50
Figura 4. 14 Agregar una ontología a BioPortal.	50
Figura 4. 15 Actualizar la información de la ontología en BioPortal.	51
Figura 4. 16 Consola de Ruby en servidor CentOS de la aplicación NCBO BioPortal.	52
Figura 5. 1 Estructura de información	55
Figura 5. 2 Categorías en la aplicación BioPortal representando la estructura de información de la Figura 5.1.	57
Figura 5. 3 Leyenda Sing In.	58
Figura 5. 4 Etiqueta Sign Up.	59
Figura 5. 5 Ejemplo de los datos solicitados para crear una cuenta de usuario.	59
Figura 5. 6 Ingresar a la pestaña Browser del portal web BioPortal.	60
Figura 5. 7 Ubicación del botón Submit New Ontology.	61
Figura 5. 8 Ventana para el ingreso de nombre, acrónimo y elegir la categoría a la que pertenecerá la ontología agregar.	61
Figura 5. 9 Ejemplo del ingreso de datos de una ontología.	62
Figura 5. 10 Continuación del ingreso de datos correspondientes a la ontología a ingresar.	63
Figura 5. 11 Ontología ingresada satisfactoriamente.	64
Figura 5. 12 Ingresar a la pestaña Browser del portal web BioPortal.	65
Figura 5. 13 Seleccionar la categoría.	65
Figura 5. 14 Ubicar la sección DOWNLOADS y dar clic sobre el enlace RDF/XML.	66
Figura 5. 15 Archivo de la ontología de la Plantilla abierto en Protegé.	67
Figura 5. 16 Comando para ingresar en la carpeta ncbo_cron.	69
Figura 5. 17 Comando escrito incorrectamente.	69
Figura 5. 18 Comando para entrar a la consola de Ruby.	70
Figura 5. 19 Comando para crear categorías.	70

Figura 5. 20 Asignar nombre a la categoría.....	71
Figura 5. 21 Asignar acrónimo a la categoría.....	71
Figura 5. 22 Guardar nombre y acrónimo de la categoría.....	71
Figura 5. 23 Comando para buscar una categoría.....	72
Figura 5. 24 Comando para eliminar una categoría.....	73
Figura 5. 25 Comando para encontrar una ontología.....	75
Figura 5. 26 Comando para eliminar una ontología.....	76
Figura 6. 1 Datos de entrada para el primer requerimiento funcional crear cuentas de usuario.....	86
Figura 6. 2 Cuenta de usuario con identificador martha2017.....	87
Figura 6. 3 Categorías definidas en la aplicación NCBO BioPortal.....	88
Figura 6. 4 Nivel I de la jerarquía de ontologías representado en la aplicación NCBO BioPortal.....	89
Figura 6. 5 Detalles acerca de la ontología de la plantilla para describir patrones de análisis.....	90
Figura 6. 6 Código de la ontología de la plantilla alojado en servidor de la aplicación NCBO BioPortal.....	91
Figura 6. 7 Código de la ontología de la plantilla representado en editor Protege.....	92
Figura 6. 8 Consulta al repositorio de ontologías.....	94
Figura 6. 9 No existen coincidencias con el nombre de la clase ingresado.....	94
Figura 6. 10 Ingreso de caracteres especiales.....	95
Figura 6. 11 Enlace para la descarga del archivo OWL de la ontología de la plantilla.....	96
Figura 6. 12 Descarga del archivo de la ontología de la plantilla con formato owl.....	96
Figura 6. 13 Parte del código del archivo de la ontología de la plantilla.....	97
Figura 6. 14 Estado de la categoría II-I Individuos de PARE antes de realizar la aportación.....	98
Figura 6. 15 Aportación de la ontología Reserva de Vuelos.....	99
Figura 6. 16 Ejemplo de error al ingresar un archivo sin extensión .owl.....	99
Figura 6. 17 Detalle de la ontología con datos ficticios Reserva de Vuelos.....	101
Figura 6. 18 Primera forma: Editar Información de la Ontología.....	101
Figura 6. 19 Segunda forma: Editar la información presentada.....	102
Figura 6. 20 Tercera forma: Agregar una nueva presentación de la información de la ontología.....	103
Figura 6. 21 Estado del archivo de la ontología antes de ser actualizado.....	103
Figura 6. 22 Salida de la forma uno para actualizar ontologías.....	104
Figura 6. 23 Salida de la forma dos y tres para actualizar ontologías.....	105
Figura 6. 24 Estado de la categoría II-I Individuos de PARE antes de eliminar la ontología Reserva de Vuelos.....	106
Figura 6. 25 Situación de la aplicación después de la eliminación de la ontología Reserva de Vuelos.....	107
Figura A. 1 Pestaña de inicio de BioPortal.....	120
Figura A. 2 Pestaña Browser de BioPortal.....	121
Figura A. 3 Pestaña Search de BioPortal.....	122
Figura A. 4 Pestaña Mappings de BioPortal.....	123
Figura A. 5 Tabla de mapeo de dos ontologías.....	123
Figura A. 6 Pestaña Recommender de BioPortal.....	124
Figura A. 7 Pestaña Annotator de BioPortal.....	125
Figura A. 8 Pestaña Resource Index de BioPortal.....	126
Figura A. 9 Pestaña Projects de BioPortal.....	127
Figura A. 10 Arquitectura de la aplicación NCBO BioPortal.....	129
Figura B. 1 Crear una máquina virtual.....	137
Figura B. 2 Ruta donde se localiza el archivo de la máquina virtual.....	138
Figura B. 3 Forma de instalación del sistema operativo invitado.....	138
Figura B. 4 Seleccionar el sistema operativo a instalar.....	139
Figura B. 5 Asignar nombre a la máquina virtual.....	139

Figura B. 6 Características de los dispositivos de la máquina virtual.....	140
Figura B. 7 Configuración de los dispositivos de la máquina virtual.	141
Figura B. 8 Inicio de la instalación de la aplicación NCBO.	142
Figura B. 9 Crear contraseña para acceder al servidor.	143
Figura B. 10 Ingreso a la aplicación virtual NCBO.....	143
Figura B. 11 Ingresar comando para obtener la IP local.	144
Figura B. 12 Interfaz del portal.	144
Figura B. 13 Editar la información del usuario administrador.	145
Figura B. 14 Registrar nombre y apellido del usuario administrador.	145
Figura B. 15 Cambiar contraseña del usuario administrador.....	145

Lista de tablas

Tabla 2. 1 Características de los niveles del SSM.	10
Tabla 3. 1 Comparación de antecedentes.	21
Tabla 3. 2 Comparación de trabajos relacionados.	28
Tabla 4. 1 Componentes UCM empleados para representar los requerimientos funcionales.....	34
Tabla 6. 1 Resultados obtenidos.	83
Tabla 6. 2 Matriz de trazabilidad.....	84
Tabla 6. 3 Identificador del caso de prueba 01.....	86
Tabla 6. 4 Identificador del caso de prueba 02.....	87
Tabla 6. 5 Identificador del caso de prueba 03.....	93
Tabla 6. 6 Identificador del caso de prueba 04.....	95
Tabla 6. 7 Identificador del caso de prueba 05.....	97
Tabla 6. 8 Identificador del caso de prueba 06.....	100
Tabla 6. 9 Identificador del caso de prueba 07.....	106
Tabla C. 1 Nomenclatura para identificar los documentos generados.....	149
Tabla C. 2 Matriz de trazabilidad de pruebas.	151
Tabla D. 1 Matriz de trazabilidad.....	158
Tabla D. 2 Pruebas a realizar correspondiente a cada actividad de los requerimientos funcionales.	159

Abreviaturas y conceptos

Aportación	Se refiere a las ontologías y/o individuos de ontología que son ingresados a la aplicación NCBO BioPortal por un usuario registrado en la aplicación.
BD	Base de Datos
BioPortal	Interface del portal web de la aplicación NCBO.
GRL	Goal-oriented Requirements Language (Lenguaje de requerimientos orientado a metas).
HTTP	Siglas en inglés de Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
Individuo	Es la instancia de ontología que se obtiene al poblar con datos cada una de las clases de dicha ontología.
IMAP	Siglas en inglés de Internet Message Access Protocol (Protocolo de acceso a mensajes de internet)
MECOPA	Metodología para la Construcción de Ontologías de Patrones de Análisis
NCBO	Centro Nacional de Ontologías Biomédicas.
OPAC	Ontología del Patrón de Análisis de Cuenta.
OPARE	Ontología del Patrón de Análisis para Reservación de Entidades
OWL	Ontology Web Language (Lenguaje web de ontologías)
OWL-DL	OWL Description Logic (Es un tipo de OWL que incorpora funciones de lógica descriptiva).
PA	Patrón de Análisis
PEPA	Plantilla Estable de Patrones de Análisis
POP3	Siglas en inglés de Post Office Protocol (Protocolo de Oficina de Correo o Protocolo de Oficina Postal)
RDF	Siglas en inglés de Resource Description Framework (Marco de Descripción de Recursos)
SDS	Siglas en inglés de Software Design Specification (Especificación de Diseño de Software)
SMTP	Siglas en inglés de Simple Mail Transfer Protocol (Protocolo para Transferencia Simple de Correo)
SRS	Siglas en inglés de Software Requirements Specification (Especificación de Requerimientos de Software).
UCM	Siglas en inglés de Use Case Maps (Mapas de casos de uso)

URN	User Requirements Notation (Notación de requerimientos de usuario)
WSGI	Siglas en inglés de Web Server Gateway Interface (Interfaz de la puerta de enlace del servidor web)
XMI	Siglas en inglés de XML Metadata Interchange (XML de Intercambio de Metadatos) es una especificación para el intercambio de Diagramas.
XML	Siglas en inglés de eXtensible Markup Language (“lenguaje de marcado extensible”)

Capítulo 1. Introducción

En esta investigación se analiza el problema de la falta de conceptos de estabilidad, así como de la falta de semántica para la descripción de los patrones de análisis. Su objetivo es encontrar la manera de almacenar descripciones de patrones de análisis siguiendo una jerarquía de ontologías y sus individuos.

El uso de patrones de análisis es importante para describir procesos de dominio y evitar errores al momento de desarrollar una solución dentro de un dominio específico. Debido a que un patrón de análisis se enfoca en los procesos que ocurren en ese dominio y no en la implementación del software.

Hasta ahora, los patrones de análisis almacenados existentes no incluyen una semántica y tampoco los conceptos de estabilidad de software. Estos patrones están descritos solamente en lenguaje RDF accesibles solo para cierto número de personas expertas. Sin embargo, personas menos expertas no podían utilizarlos. Por esta razón se propuso diseñar un repositorio de patrones de análisis que esté disponible en la web con el fin de que sea utilizado tanto por personas expertas como inexpertas.

Tener al alcance de todo tipo de analistas un repositorio de descripciones de patrones de análisis que incluyan su semántica, les facilitará la comprensión de los procesos de un dominio específico.

Las descripciones de patrones de análisis se almacenan siguiendo una estructura de información que contempla tres niveles de abstracción. Actualmente se cuenta con dos descripciones de patrones de análisis (PARE y PAC) y sus respectivas ontologías. Las ontologías están representadas en lenguaje OWL-DL.

1.1 Descripción del problema

La falta de conceptos de estabilidad, así como la falta de semántica en la descripción de los patrones de análisis ocasiona dificultad para su utilización cuando se aplican en dominios diferentes.

1.2 Objetivo

Almacenar descripciones de patrones de análisis que incluyan su semántica, mediante una jerarquía de ontologías y sus individuos.

1.3 Justificación

Debido a que en la actualidad no existe una herramienta que contenga un repositorio de patrones de análisis que incluya su semántica (elementos, atributos, relaciones entre elementos y restricciones o reglas que los gobiernan), sólo existen modelos, arquitecturas e ideas que plantean cómo podría integrarse un catálogo de patrones de análisis en una herramienta o sistema (da Matta Vegi, Alves Peixoto, Santos Soares, Lisboa-Filho, & de Paiva Oliveira, 2012).

Al incorporar un catálogo o almacén de patrones de análisis en una herramienta o sistema, permitirá al usuario encontrar descripciones de problemas, y a partir de ellas, las soluciones que ya han sido utilizadas en problemas similares. La disponibilidad de las descripciones de los patrones de análisis permitirá utilizar las mejores descripciones de problemas y habilitará la descripción de problemas complejos, mediante la composición de varios patrones de análisis. Esto repercutirá en ahorro de tiempo en la fase del análisis y mayor calidad en la descripción de problemas.

1.4 Organización del documento

Este documento se organiza de la siguiente manera:

Capítulo 2. Marco Teórico. En este capítulo se presenta los conceptos fundamentales (Ej. Entender que es un patrón de análisis, que es una ontología y la notación que se utilizó para describir de forma visual los requerimientos funcionales) que determinan el ámbito contextual en el cual se posiciona la investigación.

Capítulo 3. Estado del Arte. En este capítulo se exhiben los resultados del proceso de búsqueda de información relevante con respecto a antecedentes (investigaciones realizadas en CENIDET) y trabajos relacionados (realizados fuera de CENIDET) que sirven como punto de referencia y comparación con la investigación propuesta.

Capítulo 4. Requerimientos y adaptación de los requerimientos con BioPortal. En este capítulo se describen los requerimientos que se consideran necesarios para la realización del almacén de ontologías de patrones de análisis. Se explica, además, la manera de adaptación de los requerimientos utilizando la aplicación *NCBO BioPortal*.

Capítulo 5. Descripción del Desarrollo. En este capítulo se muestra el proceso realizado para almacenar las ontologías de descripciones de patrones de análisis que incluyen su semántica, utilizando la aplicación *NCBO BioPortal*.

Capítulo 6. Resultados y Pruebas. En este capítulo se exponen los resultados obtenidos de la investigación y los casos de pruebas siguiendo el estándar de la IEEE (IEEE Computer Society, 2008) para documentación de pruebas de software y sistemas.

Capítulo 7. Conclusiones y Trabajos Futuros. En este capítulo se comentan las conclusiones a las que se llegó, del análisis de los resultados de la investigación, con respecto al objetivo propuesto. Se presentan, los trabajos futuros que se visualizan para complementar esta investigación.

Anexo A. Descripción Técnica de *BioPortal*. En este anexo se describen las funcionales que ofrece el portal *web BioPortal* de la aplicación *NCBO*. *BioPortal* tiene un repositorio de ontologías biomédicas y es el más completo en el mundo de ontologías biomédicas.

Anexo B. Instalación de BioPortal. En este anexo se describe el proceso de instalación de la aplicación *NCBO BioPortal*.

Anexo C. Plan de Pruebas. En este anexo se describe el plan de pruebas a cubrir para verificar la funcionalidad y cumplimiento de los requerimientos funcionales que se especificaron para la implementación del almacén de ontologías de descripciones de patrones de análisis, siguiendo la jerarquía de ontologías propuesta. Siguiendo el estándar de la IEEE para la documentación de pruebas de software y sistemas.

Anexo D. Diseño de Pruebas. En este anexo se identifican las características a probar, se determina la estrategia para realizar las pruebas y el criterio de aceptación. Siguiendo el estándar de la IEEE para la documentación de pruebas de software y sistemas.

Anexo E. Procedimiento de Pruebas. En este anexo se describe el procedimiento para realizar cada uno de los casos de pruebas. Siguiendo el estándar de la IEEE para la documentación de pruebas de software y sistemas.

Capítulo 2. Marco Teórico

En el presente capítulo se explican los conceptos fundamentales que determinan el ámbito contextual en el cual se posiciona la investigación. Se detallan los conceptos de: patrones de análisis, ontologías y los lenguajes GRL y UCM, utilizados en esta investigación para describir los requerimientos funcionales de forma gráfica.

2.1 Patrones de análisis

Existen muchas definiciones de patrones de análisis, una de éstas es: “son aquellos patrones cuyo propósito es construir un modelo de análisis, el cual se enfocará en los procesos de negocio en lugar de la implementación del software”. (Pantoquilha, Raminhos, & Araujo, 2003) . Sin embargo, no nos podemos quedar con esta definición. Para entender a los patrones de análisis es necesario remontarnos un poco a los orígenes e ir viendo cómo se fue acuñando este término con el paso del tiempo.

Christopher Alexander en su libro publicado en 1977 introduce por primera vez el concepto de patrón, en el contexto de la arquitectura y lo define como: “cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que se pueda usar esa solución un millón de veces, sin hacerla dos veces de la misma manera” (Alexander, Ishikawa , & Silverstein, 1977). Posteriormente en 1994, Erick Gamma escribe un libro acerca de patrones de diseño, basándose en el concepto de Alexander, pero aplicado esta vez al software, cambiando puertas y paredes, por términos de objetos e interfaces, pero manteniendo la esencia, de que un patrón de diseño, “es una solución a un problema recurrente, en un contexto y sistema de fuerzas en particular” (Gamma, Helm, Johnson, & Vlissides, 1994).

Es hasta 1997 cuando Martin Fowler introduce el concepto de patrón de análisis. Según Fowler: “un patrón de análisis es una idea que ha sido utilizada en un contexto práctico y que probablemente se utilizará en otros” (Fowler, 1997). Además Fowler, en su libro *Analysis Patterns: Reusable Object Models*, propone los primeros patrones de análisis (PAs). Estos son considerados avanzados, complicados y pobremente descritos (Bobkawska & Grabowski, 2009).

Un año más tarde, Eduardo Fernández define al patrón de análisis como “el conjunto de clases y asociaciones que tienen alguna razón de ser en el contexto de una aplicación; esto es, un modelo conceptual de una parte de la aplicación” (Fernandez , 1998). Es una descripción del problema (sus clases y asociaciones). Fernández propone tres patrones de análisis más, utilizando una forma más clara para describirlos. Ese mismo año, otros autores presentan más patrones de análisis, tal es el caso de Rosana Braga (Braga, Germano, & Masiero, 1998), describiéndolos de manera distinta a Fowler y a Fernández, y así con el paso de los años más autores fueron generando sus propios patrones de análisis, cada uno con un formato distinto de presentarlos.

Como un intento para unificar las plantillas de patrones de análisis, Ricardo Raminhos propuso una plantilla que combina las características comunes de las plantillas ya existentes y añade nuevas características que no habían sido consideradas (Raminhos, Pantoquilha, Araújo, & Moreira, 2006).

La plantilla de Raminhos consta los 18 apartados, descritos a continuación:

1. Nombre: Identificador del patrón.
2. También conocido como: Otros nombres que hacen referencia a este patrón.
3. Historial: Registro cronológico de las versiones anteriores.
4. Ajustes estructurales: Definición de extensiones u omisiones de campos de la plantilla.
5. Problema: Breve descripción del problema que se resuelve a través del patrón.
6. Motivación: Descripción de las fuerzas involucradas y una situación representativa que motive el uso del patrón.
7. Contexto: Descripción detallada del contexto en el que el problema y la solución se repita y para el cual la solución es deseable.
8. Aplicabilidad: Descripción de las condiciones bajo las cuales el patrón puede ser aplicado.
9. Requerimientos: Listado de requerimientos funcionales, no funcionales, dependencias y contribuciones, identificación de conflictos y guía de soluciones, prioridades y participantes.
10. Modelado: Uso de modelos para representar procesos expresados en un nivel alto de abstracción.
11. Contexto resultante: Configuración del sistema después de aplicar el patrón.
12. Consecuencias: Ventajas y desventajas del uso del patrón.
13. Trampas anti-patrones: Las trampas más comunes que pueden originarse al aplicar el patrón.
14. Ejemplos: Uno o más ejemplos de aplicaciones que ilustren el uso del patrón.
15. Patrones relacionados: Lista de patrones similares que describan problemas o soluciones similares.
16. Patrones de diseño: Patrones de diseño o de arquitectura que pueden utilizarse para un mayor refinamiento.
17. Directrices de diseño: Consejos sobre la aplicabilidad del patrón, sin llegar a detalles específicos.
18. Usos conocidos: Ocurrencias conocidas de la aplicación del patrón en sistemas existentes. Se deben incluir al menos tres sistemas diferentes.

Los patrones de análisis se clasifican en cuatro tipos de acuerdo a su proceso de construcción o interpretación (Hamza & Fayad, 2002):

- Los patrones contruidos de acuerdo a la experiencia de su autor
- Los patrones contruidos a partir de analogías.
- Los patrones contruidos tomando como referencia a los patrones de diseño y su estructura.
- Los patrones contruidos usando el enfoque de estabilidad de software.

La plantilla de Raminhos abarca los tres primeros tipos de patrones de análisis, sin embargo no contempla aquellos patrones contruidos tomando como referencia el modelo de estabilidad de software (el cual se explica a continuación).

2.2 Modelo de Estabilidad de Software (SSM)

Introducido en (Fayad & Altman, 2001), presenta tres niveles conceptuales clasificados de acuerdo a su naturaleza (Figura 2.1)

Temas de negocios perdurables (EBTs): son las clases invariantes que presentan los conceptos básicos y perdurables del sistema. Por lo tanto, son extremadamente estables y forman el núcleo del SSM. Describen los principales objetivos del sistema.

Objetos de negocios (BOs): son las clases que mapean las EBTs del sistema en objetos concretos. Los BOs son tangibles y externamente estables, pero son internamente adaptables.

Objetos industriales (IOs): son las clases que mapean las BOs del sistema en objetos físicos.

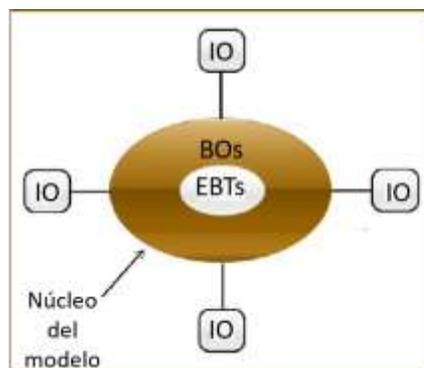


Figura 2. 1 Arquitectura de Modelos de Estabilidad de Software.

En la Tabla 2.3 se muestra las características (columnas) de los componentes de cada nivel (filas), del SSM. A continuación se describen los símbolos, usados en las columnas:

- Comportamiento (2da. columna). Define la manera de portarse del componente en le modelo:

- Estable: El componente no cambia de conducta externa e internamente, todo el tiempo.
- Estable-adaptable: El componente no cambia de conducta externamente y es adaptable (desempeña funciones distintas) internamente.
- Inestable: El componente puede reemplazarse, agregarse o removerse del sistema, sin afectar el núcleo del sistema.
- Percepción (3ra. columna). Define la naturaleza del componente a través de los sentidos.
 - Intangible: El componente representa conceptos (no de sustancia física), del dominio.
 - Semi-tangible: El componente es tangible (se puede percibir por un sentido). Sin embargo aún no es un objeto concreto (específico).
 - Tangible: El componente es completamente un objeto concreto. Usualmente representa objetos reales del mundo.
- Especificidad (4ta. columna). Establecer el grado de similitud del componente con el dominio.
 - General: El componente es común a todas las aplicaciones del dominio donde está identificado.
 - Particular: El componente puede ser completamente diferente de una aplicación a otra, incluso en el mismo dominio.

Tabla 2. 1 Características de los niveles del SSM.

Nivel de Estabilidad	Comportamiento	Percepción	Especificidad
EBTs	Estable	Intangible	General
BOs	Estable-adaptable	Semi-tangible	General
IOs	Inestable	Tangible	Particular

Tomando en cuenta este modelo surge la plantilla estable de patrones de análisis (González, Martínez, Villamil, & López, 2015). La Plantilla estable de patrones de análisis, considera como una modificación a la plantilla de Raminhos (Raminhos, Pantoquilho, Araújo, & Moreira, 2006), pero agregando el modelo de estabilidad expuesto en el artículo *An Introduction to Software Stability* (Fayad & Altman, 2001). Dicha modificación consiste básicamente en añadir aspectos dinámicos y estructurales. Para su implementación se optó por utilizar ontologías que permitieran clasificar los elementos de la plantilla en cada uno de los niveles del SSM y definir sus características, además de aprovechar todas las prestaciones que una ontología ofrece: formalización, consenso, compartición, semántica, etc. En la siguiente sección se explica más a detalle el concepto de ontología y sus características.

2.3 Ontología

Según Studer (Studer, Benjamins, & Fensel, 1998), las dos definiciones que expresan de una mejor manera, el concepto de ontología son las de Gruber (Gruber, 1993): “Una ontología es una especificación explícita de una conceptualización” y la de Borst (Borst, 1997): “Una ontología es una especificación formal de una conceptualización compartida”; y propone una nueva definición haciendo una mezcla de ambas: “Una ontología es una especificación formal y explícita de una conceptualización compartida”.

Los elementos que componen esta definición, los explica el autor de la siguiente forma:

a) Compartida: refleja la noción de que una ontología captura conocimiento consensual, que no es objeto de un sólo individuo, sino aceptado por un grupo.

a) Conceptualización: se refiere a un modelo abstracto de algún fenómeno en el mundo proveniente de haber identificado los conceptos relevantes de dicho fenómeno.

b) Explícita: significa que el tipo de conceptos utilizados y las restricciones de su uso se definen explícitamente.

d) Formal: se refiere al hecho de que la ontología debe ser legible por una computadora, lo cual se consigue con la utilización de un lenguaje de programación de ontologías. Uno de los más utilizados es el OWL, que se presenta a continuación.

2.4 Lenguaje de Ontologías Web (OWL)

El Lenguaje de Ontologías Web (OWL) (McGuinness & Harmelen, 2004)) está diseñado para usarse en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interoperabilidad de contenido Web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S), proporcionando un vocabulario adicional junto con una semántica formal. OWL tiene tres sub-lenguajes, con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full.

OWL proporciona más vocabulario que RDF para describir propiedades y clases tales como: relaciones entre clases (por ejemplo “clases disjuntas”), cardinalidad (por ejemplo “exactamente uno”) e igualdad. OWL también proporciona más tipos para las propiedades características (por ejemplo “simetría”), y clases enumeradas.

- OWL Lite está diseñado para aquellos usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples.

- OWL DL está diseñado para aquellos usuarios que requieren la máxima expresividad conservando completitud computacional (se garantiza que todas las conclusiones sean computables), y decidibilidad (todos los cálculos acaban en un tiempo finito). OWL DL se denomina de esta forma debido a su correspondencia con la lógica de descripción (Description Logics, en inglés), un campo de investigación que estudia la lógica que compone la base formal de OWL.
- OWL Full está diseñado a usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales. Por ejemplo, en OWL Full una clase puede ser considerada simultáneamente como una colección de clases individuales y como una clase individual propiamente dicha.

2.5 Notación de requerimientos de Usuario

Para la especificación de requerimientos se utilizan dos lenguajes de la Notación de Requerimientos de Usuario URN (por sus siglas en inglés *User Requirements Notation*), siguiendo la recomendación de (ITU-T, 2012). URN permite realizar licitación, análisis, especificación y validación de los requerimientos. Los lenguajes incluidos en la notación URN son:

- GRL (Goal-oriented Requirements Language)
- UCM (Use Case Maps)

La recomendación ITU-T Z.151 define al lenguaje GRL como un lenguaje para soportar modelado orientado a metas y razonamiento sobre requerimientos, especialmente requerimientos no funcionales. En la Figura 2.2 se muestran construcciones visuales para expresar varios tipos de conceptos que surgen durante el proceso de especificación de requerimientos.

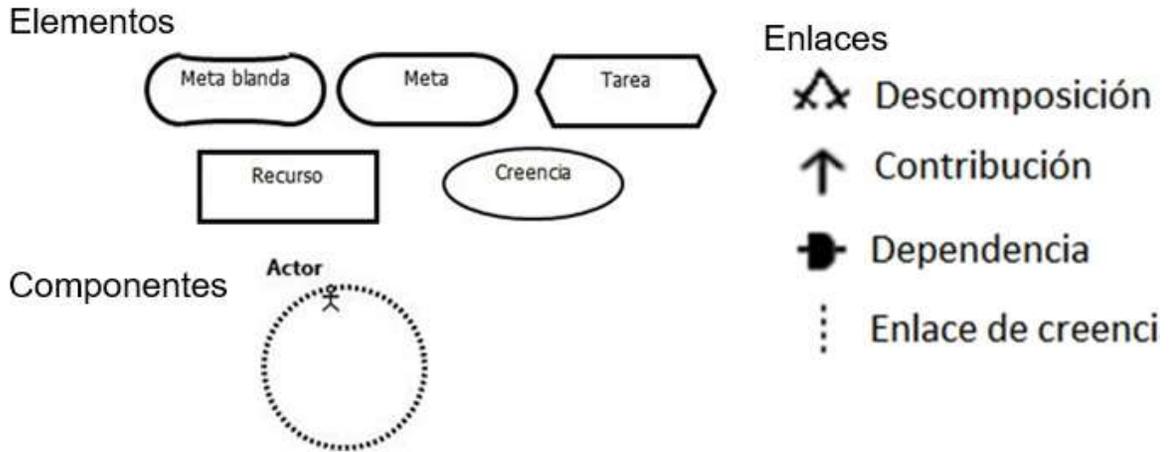


Figura 2. 2 Construcciones visuales de GRL en la herramienta JUCMNav v7.0.0.

La recomendación ITU-T Z.151 también menciona que UCM es un lenguaje de especificación, que debido a su naturaleza visual, simple e intuitiva permite ser utilizado por modeladores, no especialistas y al mismo tiempo ofrece suficiente rigurosidad para los desarrolladores. Las especificaciones UCM emplean rutas de escenarios para ilustrar las relaciones causales entre las responsabilidades, proporcionando una visión integrada de conducta y estructura por lo que permite la superposición de trayectorias de escenarios en una estructura de componentes abstractos. En la Figura 2.3 se exponen elementos para la representación de los diagramas UCM.

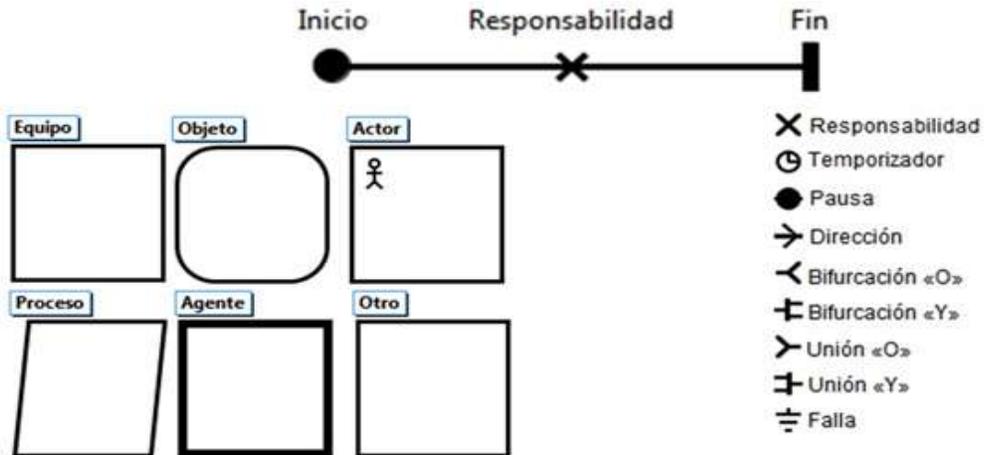


Figura 2. 3 Construcciones visuales de UCM de la herramienta JUCMNav v7.0.0.

En (Amyot, 2003), se dice que el lenguaje UCM permite describir la funcionalidad, la estructura y el comportamiento del sistema. URN representa la pieza faltante del rompecabezas de modelado ver Figura 2.4 que conecta la brecha entre requerimientos y diseño.



Figura 2. 4 GRL, UCM y otras piezas de modelado, según (Amyot, 2003).

En este capítulo se incluyen conceptos que permiten sustentar la investigación. Se incluye información acerca de patrones de análisis porque esta investigación se enfoca al almacenamiento y uso de patrones de análisis. Se expone información acerca de ontologías porque son adecuadas para representar la semántica de los patrones de análisis. Se describe los dos lenguajes de la Notación de Requerimientos de Usuario que se utilizan en el capítulo 4 para la especificación de requerimientos de manera gráfica en el Capítulo 5.

En el siguiente capítulo se presenta las investigaciones que preceden a esta investigación y las investigaciones que utilizan patrones de análisis o bases de datos que sirven de comparativa con respecto a esta investigación.

Capítulo 3. Estado del Arte

En este capítulo se presentan los resultados del proceso de búsqueda de información relevante con respecto a antecedentes (investigaciones realizadas en CENIDET) y trabajos relacionados (realizados fuera de CENIDET) que sirven como punto de referencia y comparación con la investigación de esta tesis.

3.1 Antecedentes

Respecto al uso de patrones, Rosanigo [2000] afirma que la captura del conocimiento, la experiencia de los desarrolladores expertos que toman buenas decisiones y encuentran buenas soluciones en un dominio específico, se pueden documentar en forma de patrones, explicando el problema y la razón de cada decisión. Será útil el uso de patrones cuando el mismo problema se presente en otra situación, permitiendo que los desarrolladores, analistas e ingenieros menos expertos o principiantes puedan comprender el problema y las razones para adoptar una solución en particular (Rosanigo, Maximizando reuso en software para Ingeniería Estructural, Modelos y Patrones, 2000).

En el estudio de antecedentes y trabajos relacionados al planteamiento anterior, encontramos que ayudar a los desarrolladores, analistas e ingenieros menos expertos a representar un problema, que después se resolverá mediante una solución similar que ya ha sido utilizada en otro dominio, mejorará la productividad del desarrollo de aplicaciones. Pero, para poder utilizar los patrones de análisis se requiere documentarlos y almacenarlos para consultarlos.

Para documentar patrones de análisis en investigaciones anteriores de nuestro grupo de investigación, se propuso una plantilla para la descripción de patrones de análisis estables (Martínez Jiménez, 2013) y una metodología (López Ramírez, 2015) para describir los patrones de análisis descritos con lenguaje natural y diagramas no estandarizados, mediante ontologías que incluyan la semántica de patrones de análisis.

El problema que se resuelve en la investigación que se describe en la sección 1.1 y el objetivo se describe en la sección 1.2. En este documento se describe el proceso para almacenar ontologías de patrones de análisis siguiendo la jerarquía que se observa en la Figura 5.1 y colocarlos al alcance de los usuarios a través de un portal web a manera de promover los patrones de análisis y facilitar el alcance para su utilización. Se pretende que este almacén permita que los usuarios expertos en cierto dominio aporten descripciones de patrones de análisis a los usuarios menos expertos.

Para la creación de la aplicación que almacena patrones de análisis se analizó el código del archivo owl de la ontología de la plantilla para describir patrones de análisis y el código de dos individuos de la plantilla que describen al patrón de análisis para reservación de entidades (PARE) (López Ramírez, 2015) y el patrón de análisis de cuentas (PAC) (Júarez Hernández, 2016). Mediante este análisis se pretendió obtener la estructura de una base de datos relacional para almacenar las ontologías de los patrones de análisis.

No se continuó con la estructura de la base de datos relacional de la plantilla porque cuando se terminó de analizar el código del archivo owl del individuo patrón de análisis de cuenta, se detectó que para la creación de cada ontología de un patrón de análisis, se tendría que diseñar

e implementar una base de datos relacional diferente, porque es una estructura diferente para cada ontología de patrón de análisis. Y cada una de estas bases de datos diferente se tendría que conectar a la base de datos de la ontología de la plantilla.

Debido a que se tiene contemplado que en un futuro se almacenarán más ontologías de patrones de análisis, como consecuencia la estructura de la base de datos estaría sufriendo cambios continuamente debido a que cada ontología de un patrón de análisis tiene una estructura diferente y se tiene que crear una base de datos acorde a su estructura y agregarla a la base de datos principal. En el artículo *A study on ontology storage based on relational database* se menciona que crear y eliminar tablas sería una sobre carga de recursos y la eficiencia de almacenamiento sería baja (Zhou & Yongkang, 2013). Por esta circunstancia se descartó el uso de bases de datos relacionales.

Se tuvo, en consecuencia, la necesidad de buscar otra alternativa para almacenar ontologías. Encontramos en la literatura el artículo *BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF* (Salvadores, Alexander, Musen, & Noy, 2013), en el cual se refiere un repositorio de ontologías biomédicas, accediendo al repositorio a través de un portal. La idea que se presenta en el artículo es similar a la nuestra, la diferencia en nuestro caso es que se quiere almacenar ontologías de patrones de análisis. En este artículo se menciona que el código de la aplicación es de código abierto, por lo que nos pusimos en contacto con *John Graybeal* director del programa del repositorio *BioPortal*, de la escuela de medicina de la Universidad de Stanford.

Después de enviar varios correos a *John Graybeal*, él nos proporcionó un enlace para acceder a la descarga de la aplicación de la máquina virtual *NCBO* (Centro Nacional de Ontologías Biomédicas) *BioPortal* (Whetzel PL, y otros, 2011). Al obtener una copia de la máquina virtual *NCBO*, nos dimos a la tarea de investigar el proceso de instalación debido a que la información que se nos proporcionó por parte de los encargados del programa *NCBO* no fue suficiente para lograr instalar y hacer funcionar la aplicación *NCBO BioPortal*.

La aplicación *NCBO BioPortal* ofrece al público información acerca de ontologías biomédicas, utiliza categorías para clasificar las ontologías acorde a un tipo, por ejemplo la categoría *Anatomy* que engloba las ontologías que se refieren a anatomía y la categoría *Neurologic Disease* que comprende las ontologías de enfermedades neurológicas, por citar algunos ejemplos. En nuestro caso utilizamos la creación de categorías para la representación de la estructura de información que se presenta en la Figura 5.1, del capítulo 5, sección 5.1 Propuesta de solución. Cada categoría representará un nivel dentro de la estructura de información. Se establecieron cinco categorías en la aplicación *NCBO BioPortal* para adaptar la estructura de información y almacenar las ontologías dependiendo del nivel que corresponden.

La estructura de información, mencionada en el párrafo anterior, tiene la finalidad de refinar patrones de análisis pasando por tres niveles de abstracción, desde las ontologías almacenadas en el primer nivel: ontología de la plantilla de patrones de análisis estables, individuos de la plantilla de análisis (descripciones de los patrones de análisis) y archivos XMI de los diagramas referenciados en los individuos; hasta las ontologías almacenadas en el tercer nivel: las ontologías que representan las diferencias estructurales y dinámicas que refleje el proceso del negocio de un caso específico (Ej. De una empresa o institución en particular).

Con la creación de las diferentes categorías en la aplicación *NCBO BioPortal*, se consiguió almacenar la ontología de la plantilla, los patrones de análisis y la semántica de los patrones de análisis. Siguiendo un orden para que el usuario ubique el nivel en el que requiera, desde descargar el archivo de la ontología de la plantilla para describir un patrón de análisis, consultar la información de un patrón de análisis para tener una mejor comprensión del proceso del dominio que desconozca, aportar una ontología de un patrón de análisis existente en el repositorio hasta utilizar la semántica de cada patrón de análisis almacenado en el repositorio para combinar patrones de análisis.

A continuación se describen de manera general las investigaciones realizadas en el departamento de Ciencias Computacionales en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), que preceden a esta investigación.

3.1.1 Plantilla estable para Patrones de Análisis Estables

En los resultados de tesis de maestría: “Plantilla estable de patrones de análisis” de Martínez Jiménez Rubí Celia (González, Martínez, Villamil, & López, 2015). El objetivo que se plantea es generar una plantilla de patrones de análisis que contenga elementos del modelo de estabilidad de software (se explica en la sección 2.2) y la representación del conocimiento mediante ontologías. Cuyo producto resultado es una ontología para una plantilla de patrones de análisis, la cual sirve de base para describir patrones de análisis.

3.2.2 Almacenamiento y uso de patrones de análisis pegados a su semántica

López Ramírez Ramiro Mar, en (López, 2015) propone facilitar la utilización de patrones de análisis como apoyo en la elaboración de software. Incorporando semántica a la descripción de patrones de análisis, a partir de individuos de la ontología de la plantilla estable de patrones de análisis. El resultado de la investigación de Ramiro es la especificación de requerimientos y el diseño de alto nivel del sistema AUPAS (Almacenamiento y Uso de Patrones de Análisis pegados a su Semántica). La representación del Sistema de Almacenaje y Utilización de Patrones de Análisis se muestra en la Figura 3.1.



Figura 3. 1 Representación del Sistema de Almacenaje y Utilización de Patrones de Análisis (AUPAs).

Los resultados de investigación mencionados son antecedentes directos a esta investigación, porque en (González, Martínez, Villamil, & López, 2015) se genera una plantilla estable para patrones de análisis, la cual se incluye en el almacén de información planteado en esta propuesta de tesis. La tesis de (López, 2015) es continuación de la investigación de (Martínez, 2013), pues en esta investigación se toma un individuo de la plantilla estable definida en (Martínez, 2013) y se construye una ontología del patrón de análisis que representa su semántica. Dos resultados de la tesis de (López, 2015) son: 1) la ontología del patrón de análisis de reservación de entidades reusables, y 2) la definición de los requerimientos y el diseño de alto nivel del sistema AUPAS. En la Tabla 3.1 se presenta una comparativa de los trabajos relacionados señalados en este apartado. A continuación se describen los símbolos y columnas de la tabla mencionada.

Símbolos empleados:

- PA: Patrón de análisis
- OWL: Lenguaje de Ontologías para la Web, tiene como objetivo facilitar un modelo de marcado (mediante etiquetas) construido sobre RDF (Marco de Descripción de Recursos, por sus siglas en ingles) y codificado en XML (Lenguaje de Marcas Extensible, por sus siglas en ingles).
- OWL-DL: Lenguaje de Ontologías para la Web para Lógica Descriptiva, incluye todas las construcciones del lenguaje OWL, además de incluir restricciones expresadas en lógica descriptiva.

Columnas:

- Ontología (1ra columna): Utilización de ontología para resolución del problema de estudio.
- Herramienta (2da columna): Herramientas de apoyo para el desarrollo del trabajo de investigación.
- Metodología o Lenguaje (3ra columna): Contiene los distingos métodos y lenguajes para la formalización y creación de ontologías.
- Perspectiva del negocio (4rta columna): Define la perspectiva empleada para definir el negocio.
- Producto resultado (5ta columna): Contiene los productos resultantes del trabajo de investigación, los cuales pueden ser, herramientas, modelos, componentes, arquitecturas, entre otros.

Tabla 3. 1 Comparación de antecedentes.

Antecedentes	Ontología	Herramienta	Metodología o lenguaje	Perspectiva del negocio	Producto resultado
Plantilla Estable de Patrones de Análisis Estable (Martínez, 2013)	Representa la semántica de la descripción de los PAs	Protége	OWL	PAs	<ul style="list-style-type: none"> • Ontología de Plantilla Estable para PAs Estables. • Descripción de tres PAs.
Almacenamiento y uso de patrones de análisis apegados a su semántica (López, 2015)	Describe la semántica del PA para Reservación de Entidades (PARE).	Protége	OWL	PA y su semántica	<ul style="list-style-type: none"> • Diseño de alto nivel del sistema AUPAS. • Ontología de PARE.
Esta investigación	Almacenar descripciones de PAs que incluyan su semántica.	NCBO BioPortal	OWL, OWL-DL	PAs	<ul style="list-style-type: none"> • Almacén de información que permite almacenar descripciones de PAs y su semántica. • Jerarquía de ontologías que ayuda al refinamiento de patrones de análisis.

En la Tabla 3.1 se describe lo que se tiene hasta ahora: una manera de representar permanentemente la semántica de la descripción de los patrones de análisis (Martínez, 2013) y la descripción de la semántica de los patrones de análisis utilizando ontologías (López, 2015), bajo estos antecedentes se propone almacenar descripciones de patrones de análisis y la semántica de éstos. Con el objetivo de colocar las descripciones de patrones de análisis y su semántica, al alcance de los usuario a través de un portal web a manera de promover los patrones de análisis y facilitando el alcance para su utilización

3.2 Trabajos relacionados

En este apartado se analizan los aspectos más relevantes que proponen diversos autores con respecto a sus resultados de investigación en temas de patrones de análisis, bases de datos, repositorios y ontologías que tienen una relación con la investigación que se reporta en este documento.

3.2.1 An infrastructure oriented for cataloging services and reuse of analysis patterns

En (da Matta Vegi, Alves Peixoto, Santos Soares, Lisboa-Filho, & de Paiva Oliveira, 2012) se propone una infraestructura de servicios web para la reutilización de patrones de análisis (APRI, por sus siglas en inglés), integrado por servicios Web, estructurados en una arquitectura SOA (Arquitectura Orientada a Servicios). En la Figura 3.2 se muestra APRI, la cual consta de los siguientes componentes:

- Portal de Patrones: contiene un conjunto de sitios enfocados en la obtención de los patrones de análisis, herramientas y servicios que proporcionan el descubrimiento; la catalogación y la reutilización de ellos.
- Repositorio de metadatos: es un depósito que contienen metadatos en XML (Lenguaje de marcas extensibles) para la especificación de los patrones de análisis y los servicios de contenido en el APRI.
- Repositorio Patrones de Análisis: es un repositorio que contienen los patrones de análisis en formato XMI (XML intercambio de metadatos), lo que permite su uso para los servicios de visualización y colaboración.
- Portal del servicio: es un servicio que apoya la visualización de patrones de análisis en APRI.
- Catálogo de Servicio: este servicio permite descubrir el uso de modelos y servicios de análisis en APRI, basándose en sus metadatos.
- Servicio de acceso: este servicio permite acceder y descargar los patrones de análisis.
- Servicios de colaboración: este servicio permite al diseñador con experiencia contribuir al mejoramiento de los patrones de análisis.

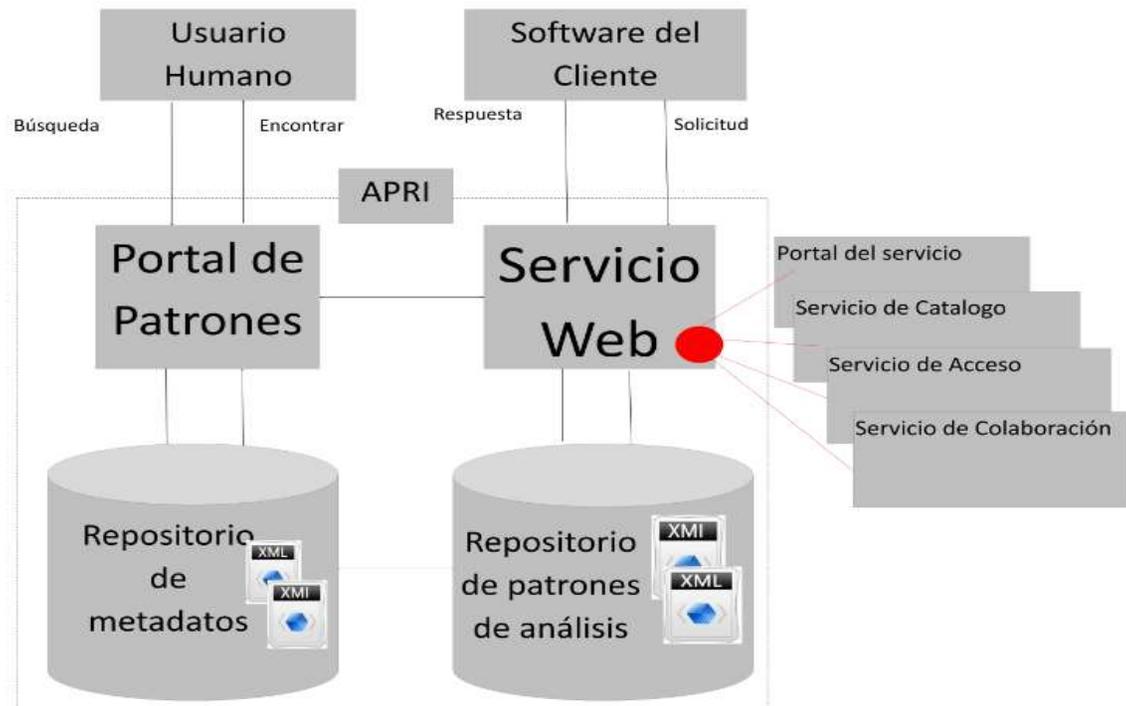


Figura 3. 2 Infraestructura para la reutilización de patrones de análisis (APRI).

En (da Matta Vegi, Alves Peixoto, Santos Soares, Lisboa-Filho, & de Paiva Oliveira, 2012) brinda una visión, sobre cómo otros investigadores perciben, la manera en que debe estar compuesto una infraestructura orientada a servicios para apoyar el almacenamiento y reutilización de patrones de análisis.

3.2.2 Improving productivity and quality of GIS databases desing using an analysis pattern catalog

En (de Freitas Sondré, Lisboa F., Vilela, & Alvim Andrade, 2005) se propone la idea de conectar un catálogo de patrones de análisis en una herramienta *CASE* (ver Figura 3.3), la cual tiene por objetivo ayudar al diseñador de bases de datos a encontrar soluciones similares que ya han sido utilizadas en aplicaciones GIS (Sistema de información geográfica, por sus siglas en ingles). Esto mejora la productividad del desarrollo de aplicaciones GIS.

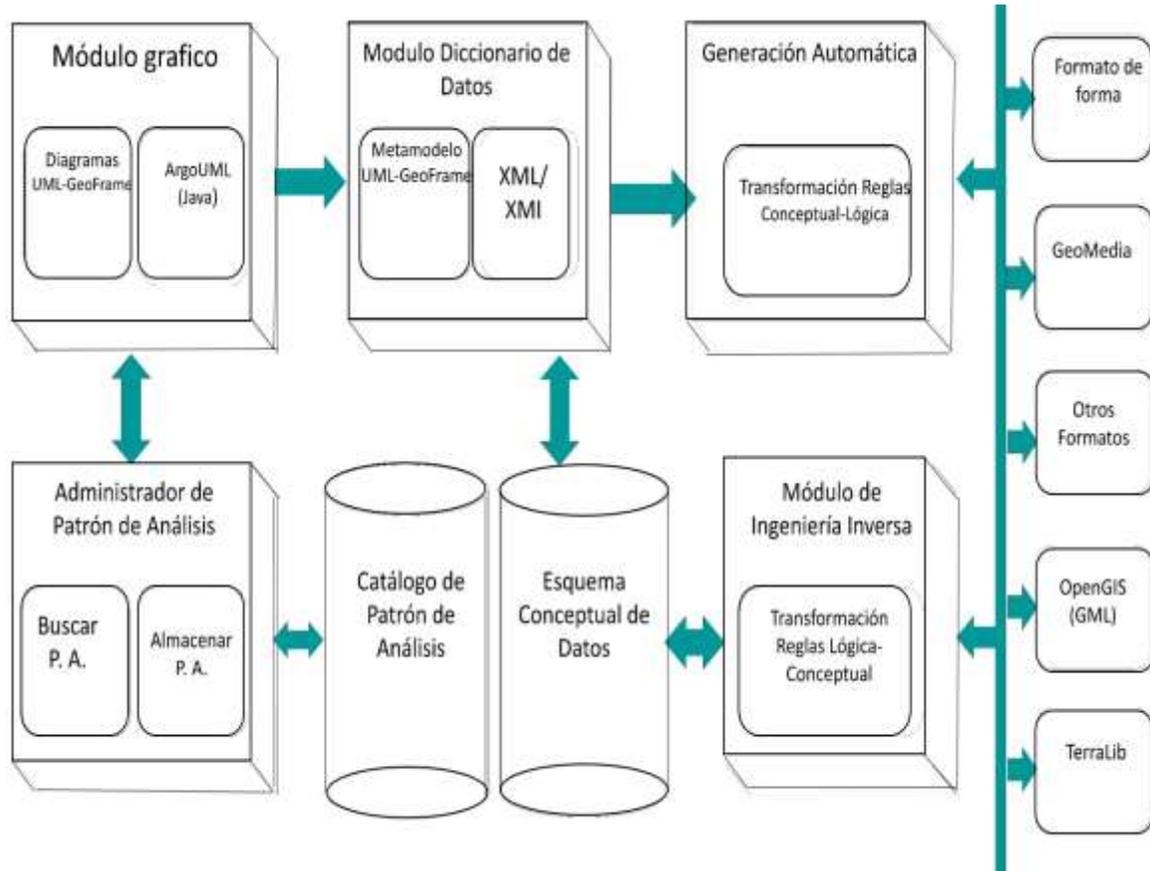


Figura 3. 3 Arquitectura de la herramienta ArgoCASEGEO.

En (de Freitas Sondré, Lisboa F., Vilela, & Alvim Andrade, 2005) nos da razones para trabajar en el almacenamiento de descripciones de patrones de análisis, porque es conveniente para ayudar al analista a encontrar soluciones, que ya han sido utilizadas en aplicaciones similares.

3.2.3 Ontodbench: novel benchmarking system for ontology-based databases

En (Jean, Bellatreche, Fokou, Baron, & Khouri, 2012) se propone un sistema de evaluación comparativa llamado OntoDBench. Para evaluar el rendimiento y escalabilidad de modelos de almacenamiento disponibles para datos ontológicos. La forma comparativa de evaluación de OntoDBench es:

- 1.- Evaluar las características relevantes de un conjunto de datos reales.
- 2.-Clasificar el conjunto de datos siguiendo los modelos de almacenamiento existentes.
- 3.-Expresar consultas de carga de trabajo basadas en estos modelos.
- 4.- Evaluar el rendimiento de la consulta.

En (Jean, Bellatreche, Fokou, Baron, & Khouri, 2012) nos muestra que la explosión de los datos basados en ontologías en la Web impulsa a los sistemas manejadores de bases de datos comerciales y académicos a ampliar sus funciones para apoyar este nuevo tipo de datos.

3.2.4 Be careful when designing semantic databases: data and concepts redundancy

Inspirándose en el diseño de bases de datos relacionales. En (Chakroun, Bellatreche, Ait-Ameur, Berkani, & Jean, 2013) los autores proponen incorporar relaciones de dependencia en el proceso de diseño de bases de datos semánticas con el fin de reducir la redundancia mediante la generación de un modelo lógico normalizado. Para llevar a cabo lo anterior ellos proponen una metodología para el diseño de bases de datos semánticas basadas en ontologías. En la Figura 3.4 se exhibe el enfoque del diseño de bases de datos semántica. La metodología está compuesta por las siguientes etapas:

- 1.- Definición del modelo conceptual.
- 2.- Identificación de los conceptos canónicos.
- 3.- Generación de un esquema lógico normalizado.
- 4.- Ocultación de la implementación física.
- 5.- Soporte al acceso de las clases no canónicas.
- 6.- Desplegar.

Para comprobar este trabajo los autores utilizan un fragmento amplio de la ontología de la universidad de Lehigh, el proceso de validación contiene dos pasos principales:

- 1.- La etapa del diseño
- 2.- La etapa del despliegue donde se propone una herramienta basada en servicios case.

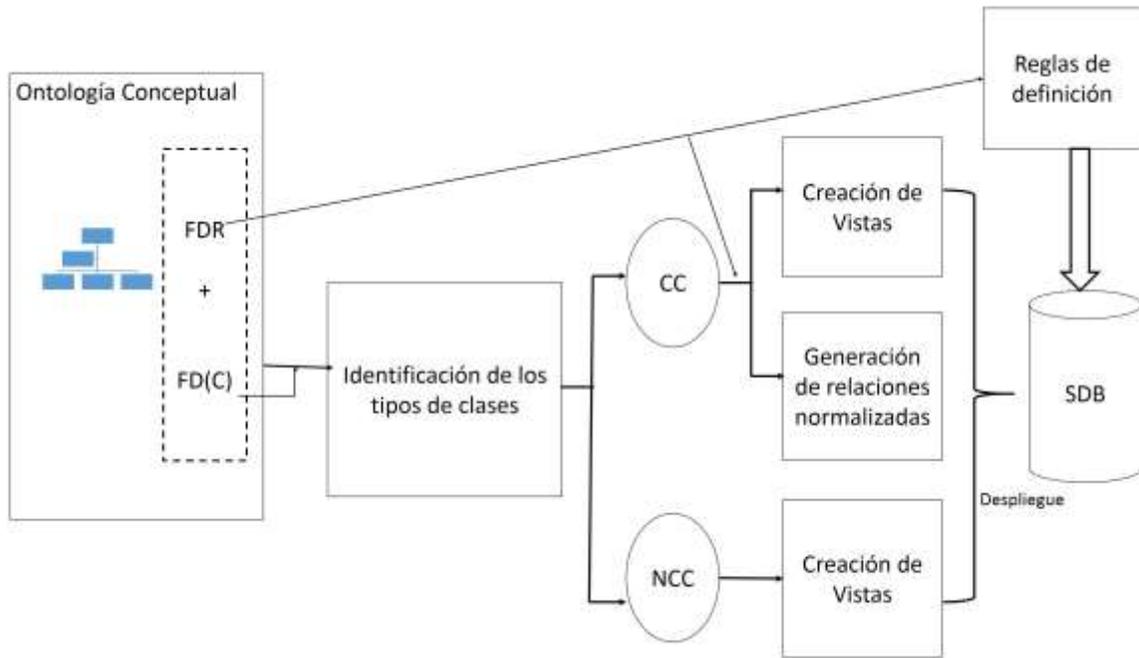


Figura 3. 4 Enfoque del diseño de bases de datos semántica.

En (Chakroun, Bellatreche, Ait-Ameur, Berkani, & Jean, 2013) se proporciona un nuevo tipo de base de datos, llamada base de datos semántica, para almacenar ontologías. Y la importancia de hacer uso de las ontologías para mantener las relaciones entre los conceptos que pertenecen a un dominio.

3.2.5 Toward an ontology based approach for data warehousing

En (Elamin & Feki, 2014) se realiza un estudio exhaustivo de veinticinco trabajos. Con el fin de comparar el método que se aplica en el diseño de un almacenamiento de datos, basado en ontologías que se propone en cada trabajo. Tomando en cuenta el resultado de este estudio, los autores sugieren un enfoque híbrido basado en ontologías para el diseño de almacenamiento de datos, el cual se observa en la Figura 3.5. Para mejorar, cada paso planteado, se realiza una ontología para representar el conocimiento, para ayudar aliviar los problemas semánticos. El enfoque híbrido propuesto consiste en tres pasos principales:

- 1.- Construcción de los esquemas principales en base a negocios-requisitos.
- 2.- Construcción de los esquemas desde una fuente de datos operacional.
- 3.- Combinación de dos conjuntos de esquemas para producir un modelo de almacenamiento de datos compatibles con el usuario y la fuente de datos al mismo tiempo.

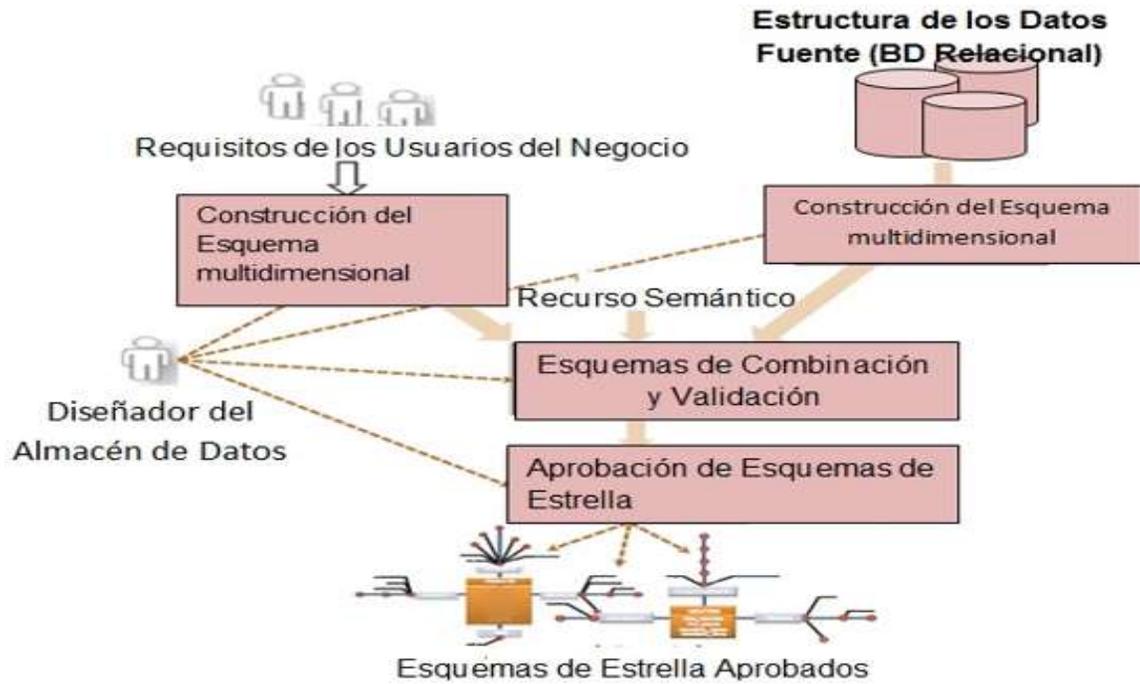


Figura 3. 5 Panorama del enfoque propuesto.

En (Elamin & Feki, 2014) se muestra la tendencia en el uso de ontologías como una forma de realizar el diseño de almacenamiento de dato. Sin embargo, las investigaciones hasta ahora centran el uso de ontologías como un paso aislado del proceso del diseño de almacenamiento de datos

En la Tabla 3.2 se presenta una comparativa de los trabajos relacionados señalados en este apartado. A continuación se describen las columnas de la tabla mencionada.

Columnas:

- Almacenan (1ra columna): Tipo de datos o archivos que guardan en un repositorio.
- Propósito (2da columna): Finalidad para lo cual se requiere realizar el trabajo.
- Perspectiva del negocio (3ra columna): Define la perspectiva empleada para definir el negocio.
- Utilizan (4ta columna): Medio de apoyo para el desarrollo del trabajo de investigación.
- Producto resultado (5ta columna): Contiene los productos resultantes del trabajo de investigación, los cuales pueden ser, herramientas, modelos, componentes, arquitecturas, entre otros.

Capítulo 3. Estado del Arte

Tabla 3. 2 Comparación de trabajos relacionados.

Proyecto	Almacenan	Propósito	Perspectiva del negocio	Utilizan	Resultado obtenido
(da Matta Vegi, Alves Peixoto, Santos Soares, Lisboa-Filho, & de Paiva Oliveira, 2012)	Archivos en formato RDF (Marco de descripción de recursos).	Apoyar la catalogación y reutilización de patrones de análisis.	Reutilizar patrones de análisis	Repositorio de patrones de análisis.	Infraestructura conceptual para la reutilización de patrones de análisis (APRI)
(de Freitas Sondré, Lisboa F., Vilela, & Alvim Andrade, 2005)	Modelos conceptuales de bases de datos geográfica en formato XML	Ayudar al diseñador de bases de datos a encontrar soluciones similares en aplicaciones SIG.	Reutilizar patrones de análisis	Catálogo de patrones de análisis.	Arquitectura conceptual de la herramienta ArgoCASEGEO incorporando un catálogo de patrones de análisis
(Jean, Bellatreche, Fokou, Baron, & Khouri, 2012)	Datos ontológicos	Comparar arquitecturas de bases de datos basadas en ontologías.	Evaluar el rendimiento y escalabilidad de los modelos de almacenamientos disponibles para datos ontológicos.	Modelos de almacenamientos para datos ontológicos.	Sistema de evaluación comparativa llamado OntoDBench
(Chakroun, Bellatreche, Ait-Ameur, Berkani, & Jean, 2013)	Reglas definidas	Reducir la redundancia de datos.	Incorporar relaciones de dependencia en el proceso de diseño de bases de datos semánticas	Relaciones de dependencia.	Una metodología para diseñar bases de datos semánticas.
(Elamin & Feki, 2014)	Datos basados en ontologías	Destacar la falta del uso de ontologías en el proceso del diseño del almacén de datos.	Realizar un estudio exhaustivo de veinticinco trabajos.	Diferentes métodos para el diseño de almacenamiento de datos.	Un enfoque híbrido para el diseño de almacenamiento de datos basados en ontologías.
En esta investigación.	Ontologías de descripciones de patrones de análisis.	Almacenar patrones de análisis	Almacenar la semántica de las descripciones de patrones de análisis	Herramienta NCBO BioPortal	Estructura de información que ayuda al refinamiento de patrones de análisis.

En la Tabla 3.2 se describe lo que se ha realizado hasta ahora respecto al uso de patrones de análisis y uso de ontologías: Una página web que apoya la catalogación y reutilización de patrones de análisis (da Matta Vegi, Alves Peixoto, Santos Soares, Lisboa-Filho, & de Paiva Oliveira, 2012). La idea de agregarle un repositorio de patrones de análisis a la herramienta ArgoCASEGEO para ayudar al analista a encontrar soluciones, en aplicaciones del sistema de información geográfica (de Freitas Sondré, Lisboa F., Vilela, & Alvim Andrade, 2005). El sistema OntoDBench que indica el tipo de modelo de bases de datos (respecto a tres modelos) disponible, que se adecua para el almacenamiento de datos ontológicos (Jean, Bellatreche, Fokou, Baron, & Khouri, 2012). Un nuevo tipo de base de datos, llamada base de datos semántica, para almacenar ontologías, para reducir la redundancia de datos (Chakroun, Bellatreche, Ait-Ameur, Berkani, & Jean, 2013). La tendencia en el uso de ontologías como una forma de realizar el diseño de almacenamiento de datos (Elamin & Feki, 2014). Tomando en cuenta estos trabajos en esta tesis se propone almacenar descripciones de patrones de análisis y la semántica de estos con formato owl. Con el objetivo de colocar las descripciones de patrones de análisis y su semántica, al alcance de los usuario a través de un portal web a manera de promover los patrones de análisis y facilitando el alcance para su utilización.

En este capítulo se realizó una comparativa de los principales antecedentes y trabajos relacionados previos a este trabajo, se explicaron las similitudes y las diferencias con la presente investigación, además de servir como punto de referencia para ver la importancia de esta aportación con respecto de las demás. De acuerdo a los trabajos presentados podemos afirmar que no existe un almacén de patrones de análisis que apliquen ontologías para su mejor descripción. En el capítulo 4, se describen los requerimientos funcionales para almacenar descripciones de patrones de análisis, que incluyan su semántica, mediante una estructura de información.

Capítulo 4. Requerimientos y adaptación de los requerimientos con BioPortal

En este capítulo se describe la especificación de los requerimientos de software para resolver el problema de almacenar descripciones de patrones de análisis, siguiendo una jerarquía de ontologías que consta de tres niveles de abstracción, en un repositorio que esté disponible al público. Se describe cada funcionalidad de la aplicación NCBO BioPortal, que se utilizó para satisfacer los requerimientos funcionales presentados.

4.1 Requerimientos funcionales

4.1.1 Requerimientos funcionales (diagramas)

El diagrama que se observa en la Figura 4.1 muestra los requerimientos que debe cumplir la infraestructura de almacenamiento de ontologías de patrones de análisis.

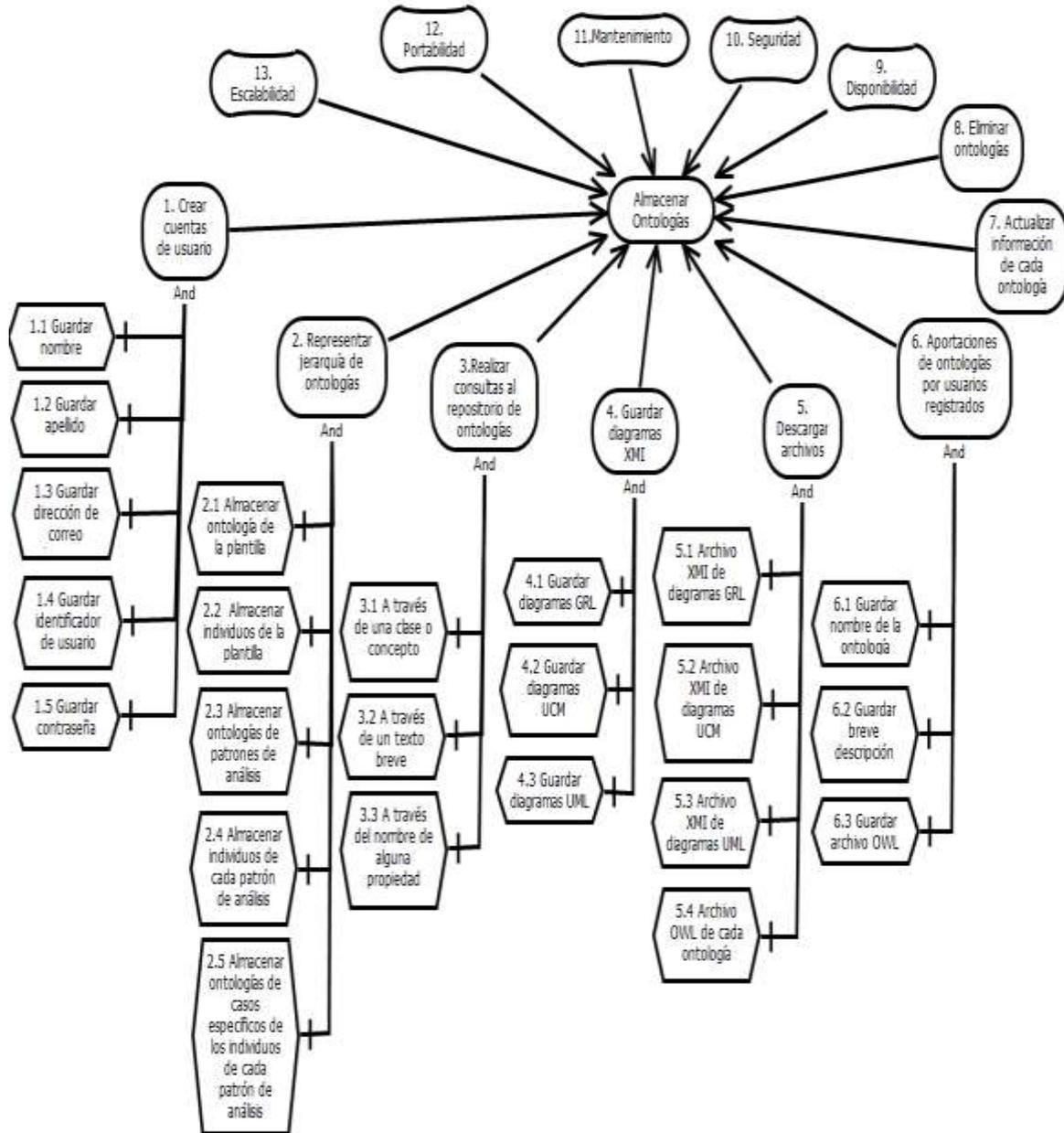
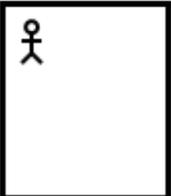
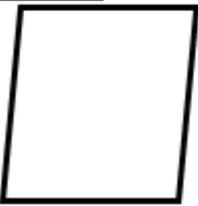


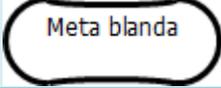
Figura 4. 1 Diagrama GRL de los requerimientos funcionales del almacén de información.

En la Figura 4.1 se muestra el diagrama GRL, que plasma las metas, metas blandas y tareas que intervienen para el cumplimiento del objetivo de almacenar descripciones de patrones de análisis. El diagrama representa, de manera gráfica, los requerimientos funcionales y no funcionales, para la elaboración de un software que almacene descripciones de patrones de análisis y la semántica de estos.

En la Tabla 4.1 se describen, los componentes de la notación UCM, empleados para describir de manera visual las trayectorias de los escenarios y combinarlos con las metas representadas en los diagramas GRL, para expresar y razonar acerca de los requerimientos funcionales.

Tabla 4. 1 Componentes UCM empleados para representar los requerimientos funcionales.

Componente UCM	Descripción
<p>Actor</p> 	<p>Se utiliza para especificar los Usuarios y el Usuario Administrador.</p>
<p>Equipo</p> 	<p>Se emplea para establecer la Interfaz Web que se considera que se observa en un navegador web el cual se encuentra alojado en un equipo de cómputo. También se dispone para representar el servidor de la aplicación que se localizará en un equipo de cómputo.</p>
<p>Proceso</p> 	<p>Se aplica para figurar Base de Datos, Repositorio de Ontologías y Repositorio de Diagramas.</p>
	<p>Esta línea es el camino que se sigue para efectuar el requerimiento señalado, en el cual el círculo relleno de color negro indica el punto de partida de un requerimiento, el símbolo con forma de la letra equis significa una tarea o responsabilidad para llevar a cabo el requerimiento y la barra vertical indica que el requerimiento ha sido realizado.</p>
	<p>Se emplea para definir el objetivo del sistema y cada uno de los requerimientos funcionales que nos llevan a obtener el objetivo principal.</p>

Componente UCM	Descripción
 Tarea	Se emplea para representar cada tarea a realizar, para el cumplimiento de cada requerimiento funcional.
 Meta blanda	Se utiliza en para representar los requerimientos no funcionales.

A continuación se muestran los diagramas UCM de cada una de las metas que se señalan en la Figura 4.1. Se puede apreciar en la Figura 4.2, un escenario para la meta 1, en el que el usuario entrará en la interfaz web y creará una cuenta. En la interfaz web se requiere que el usuario ingrese su nombre, primer apellido, dirección de correo electrónico, un nombre para su cuenta de usuario y una contraseña, posteriormente estos datos se guardaran en una base de datos que se localice en el servidor de la aplicación.

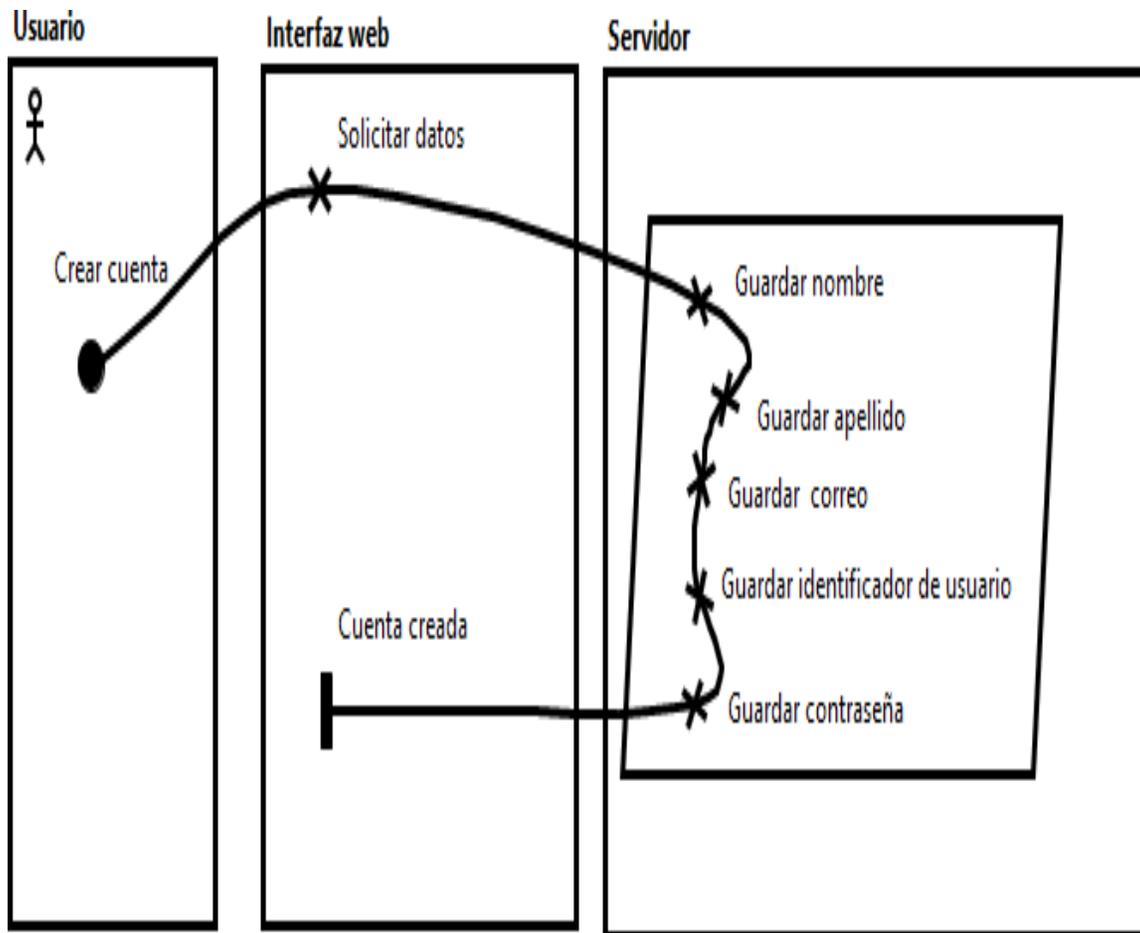


Figura 4. 2 Diagrama UCM meta 1.Crear cuentas de usuario del diagrama GRL.

Se expone en la Figura 4.3, un escenario para la meta 2, donde se presenta la trayectoria para la creación de la jerarquía de ontologías. En Primer lugar se debe de crear la jerarquía de acuerdo a la estructura que se expone en la Figura 5.1 del capítulo 4, subsiguiente se almacenan las ontologías siguiendo el orden convenido.

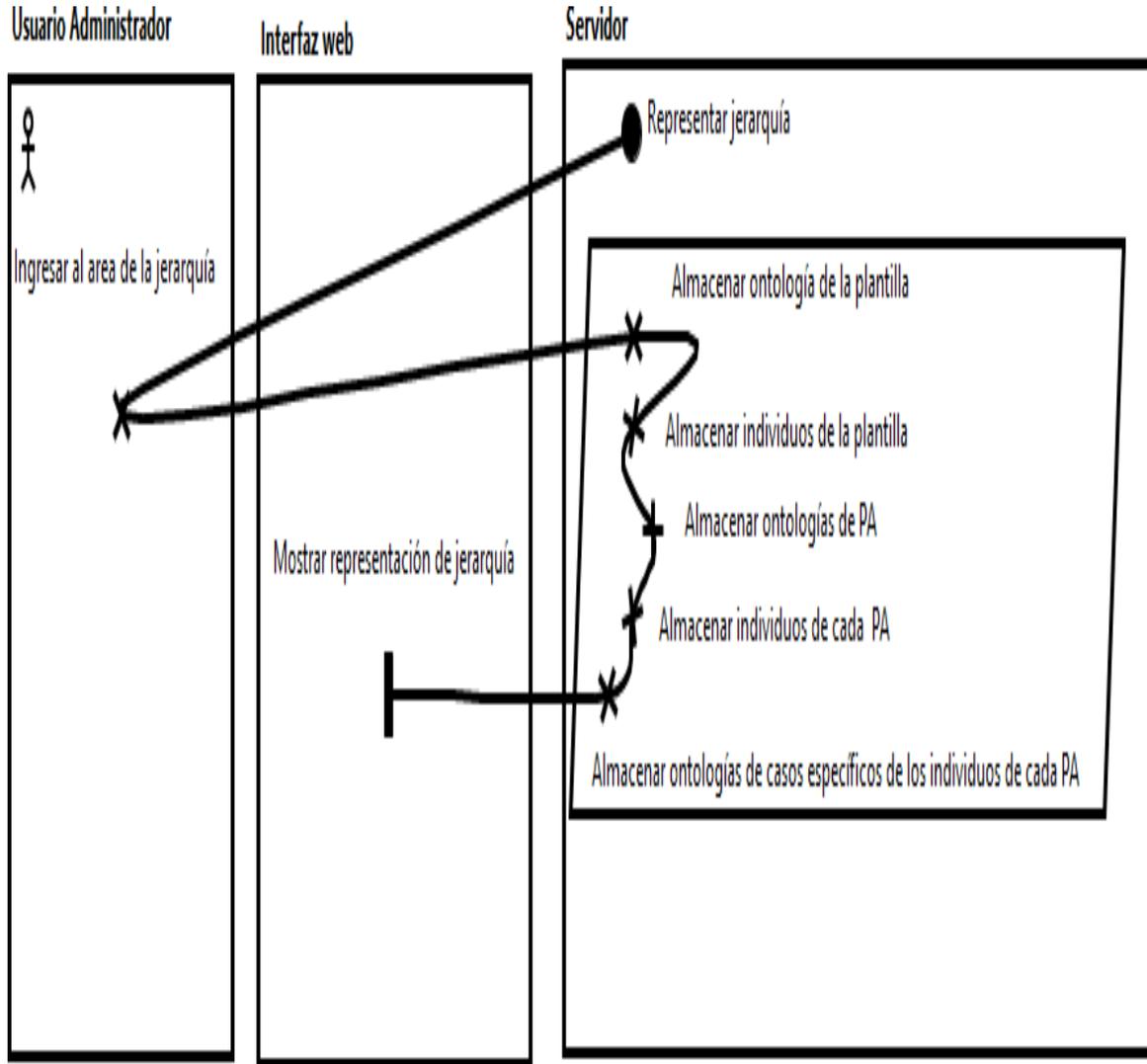


Figura 4. 3 Diagrama UCM meta 2. Representar jerarquía de ontologías del diagrama GRL.

El procedimiento para lograr la meta 3, para hacer consultas al repositorio de ontologías se presenta en la Figura 4.4. Se busca que las consultas se realicen a través del identificador de una clase, de un texto breve, o del valor de alguna propiedad y como resultado se presenten ontologías, que coincidan de manera parcial o total con el texto o valor proporcionados.

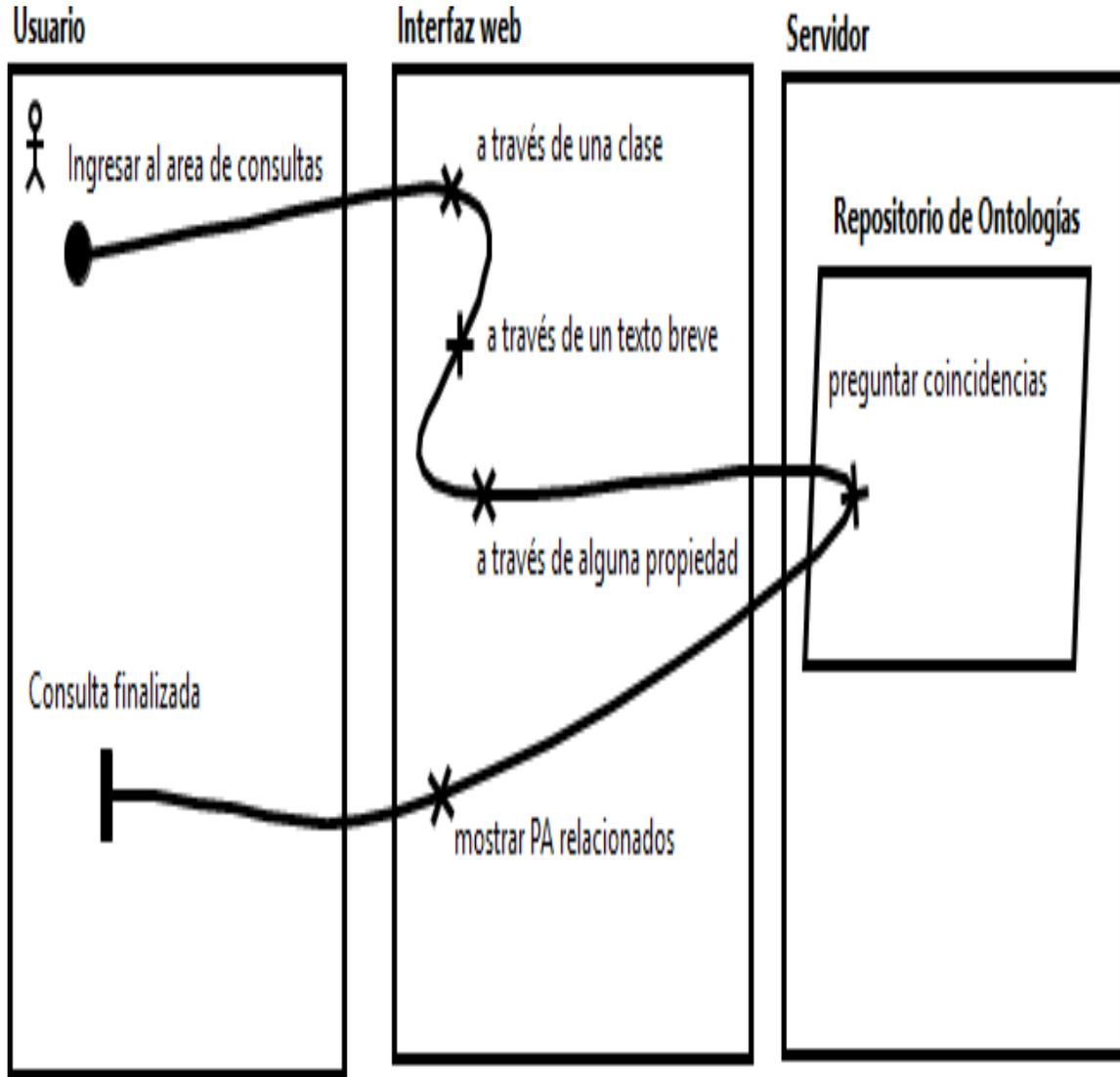


Figura 4. 4 Diagramas UCM meta 3. Realizar consultas al repositorio de ontologías del diagrama GRL.

Los diagramas que se refiere como diagramas *XMI* son diagramas *UML*, *GRL* y *UCM*, se representa en *XMI* porque éste es el formato que se requiere para almacenarlos en el repositorio. Se puede observar en la Figura 4.5, un escenario para la meta 4, donde un usuario desea agregar diagramas al repositorio. Para esto se requiere que se disponga de una cuenta de usuario, y que sea el responsable de haber ingresado la ontología del patrón de análisis. Solo las ontologías de descripciones de patrones de análisis cuentan con diagramas *XMI*.

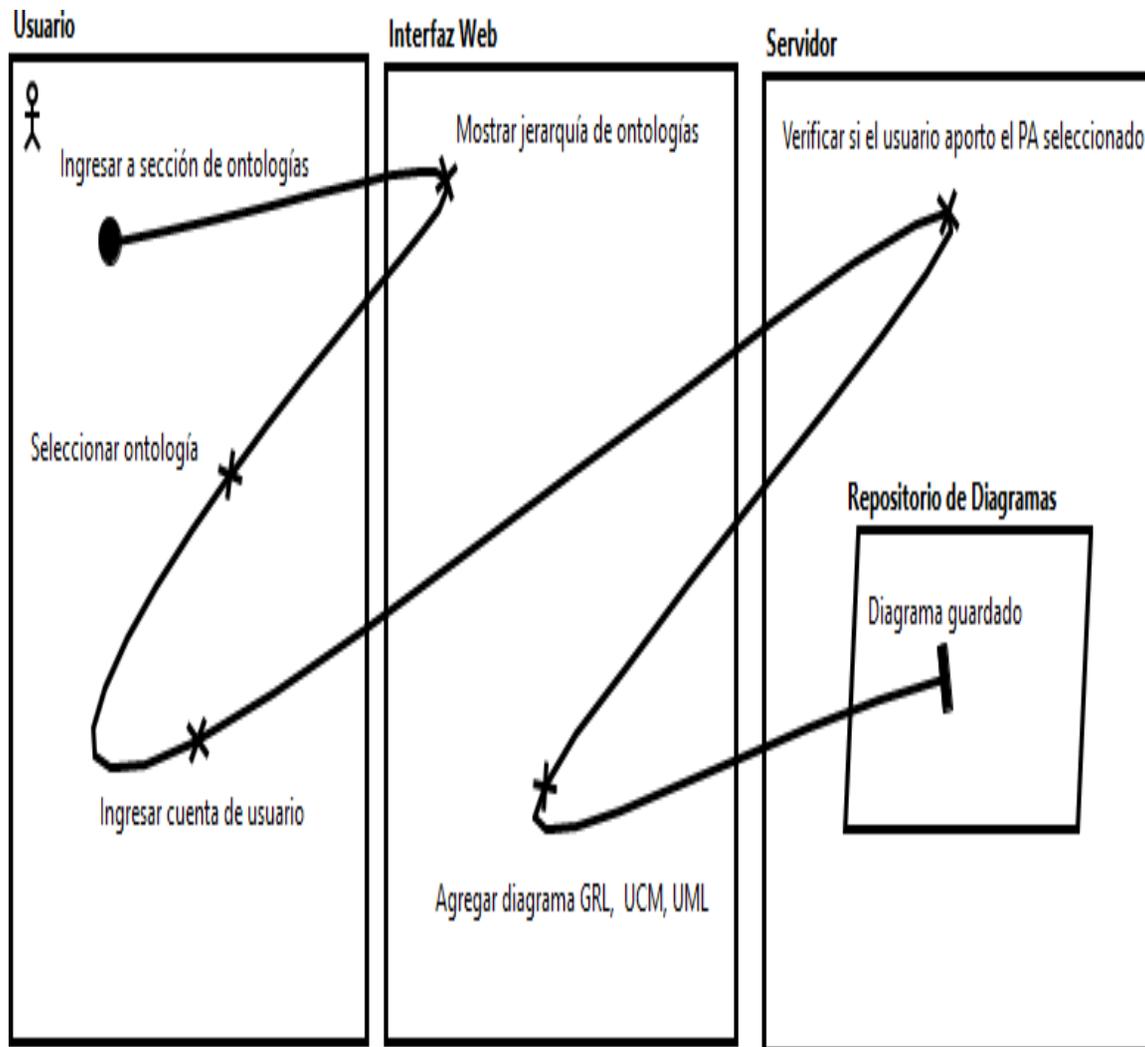


Figura 4. 5 Diagrama UCM meta 4. Guardar diagramas XMI del diagrama GRL.

Se contempla descargar los archivos de los diagramas *GRL*, *UCM*, *UML* en formato *XMI* para que el usuario visualice los diagramas en un editor que soporte este tipo de formato. Y el archivo de la ontología en formato *OWL*, porque la estructura de los archivos sirven para describir otros patrones o bien la información que contienen ayudan a la construcción de otro tipo de ontologías de patrones de análisis. El usuario entrará a la interfaz, en la sección de ontologías y seleccionará el archivo de la ontología que desea descargar, para una mejor comprensión visualizar la Figura 4.6.

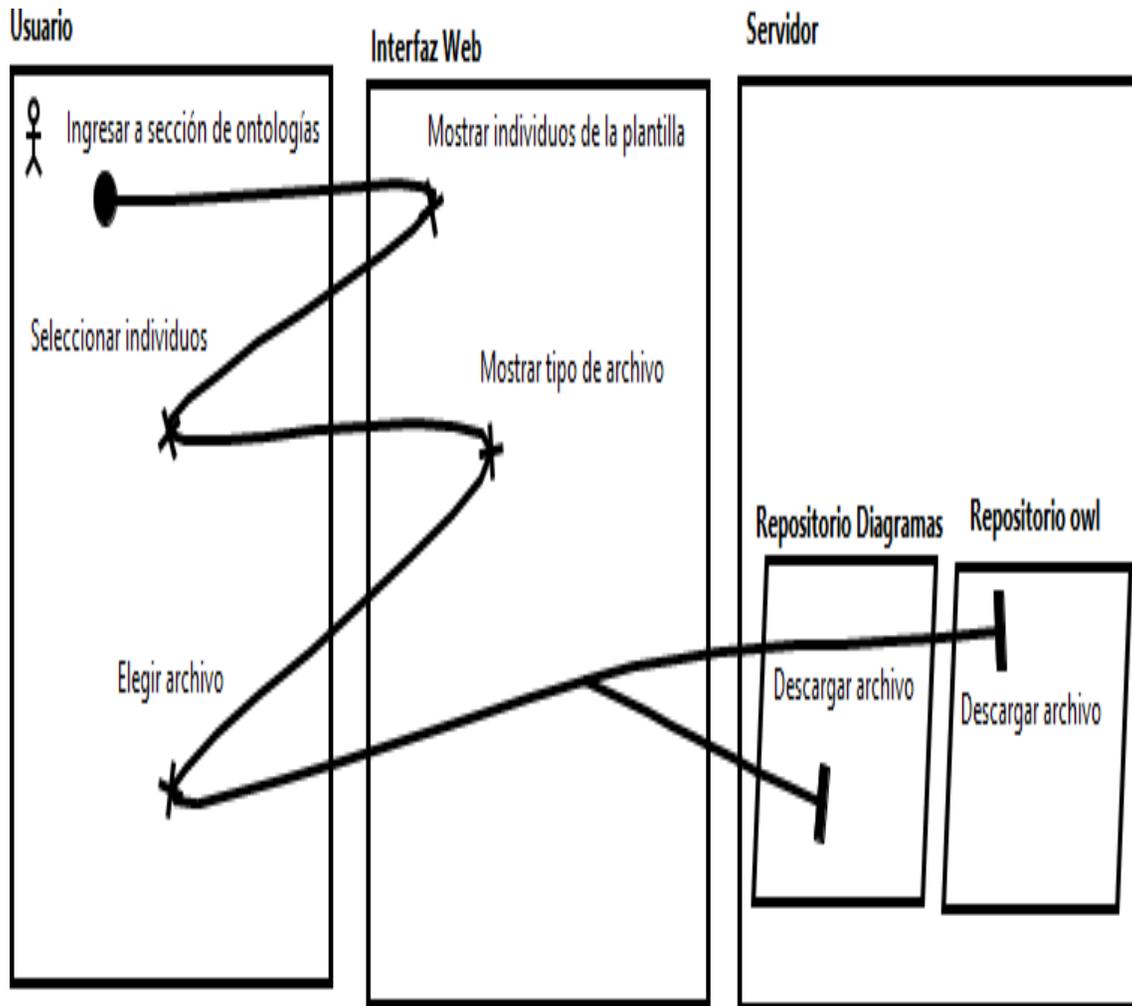


Figura 4. 6 Diagrama UCM meta 5.Descargar archivos del diagrama GRL.

Es importante que los usuarios realicen aportaciones de ontologías de patrones de análisis para enriquecer el repositorio. El usuario interesado en brindar aportaciones de ontologías, sigue el escenario para la meta 6, para esto debe de contar con una cuenta de usuario de la aplicación, posteriormente ubicarse en el nivel acorde a la aportación a realizar, seguir con la descripción hasta terminar la trayectoria, ver la Figura 4.7.

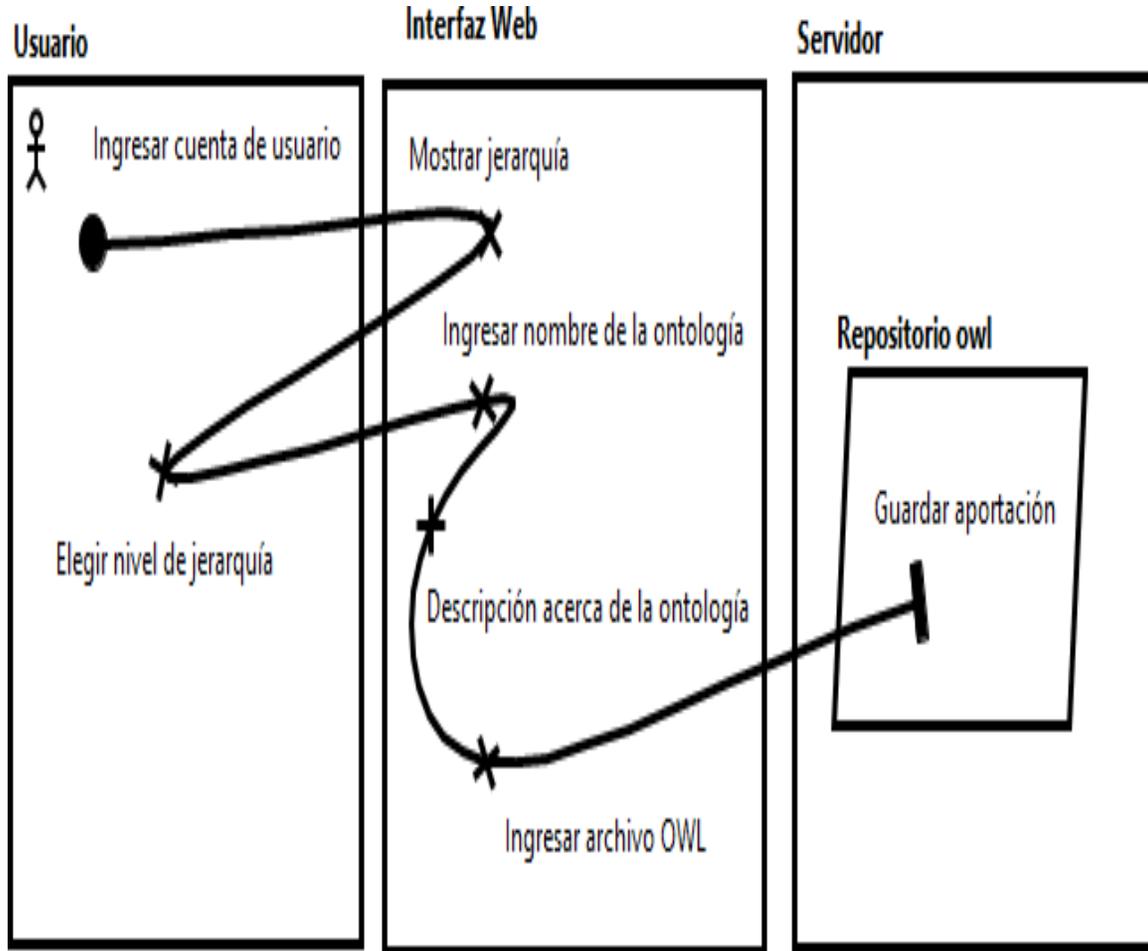


Figura 4. 7 Diagrama UCM meta 6. Aportaciones de ontologías por usuarios registrados del diagrama GRL.

Actualizar las ontologías de patrones de análisis, meta 7. Se requiere que se ingrese a la aplicación con cuenta de usuario, seleccione la sección de ontologías de patrones de análisis y seleccione la ontología a actualizar. El servidor comprobará si el usuario realizó la aportación de la ontología seleccionada. En caso de resultar verídico, se le permite al usuario realizar las modificaciones pertinentes, de lo contrario no se le permitirá realizar cambios a la ontología seleccionada. Considerar la Figura 4.8.

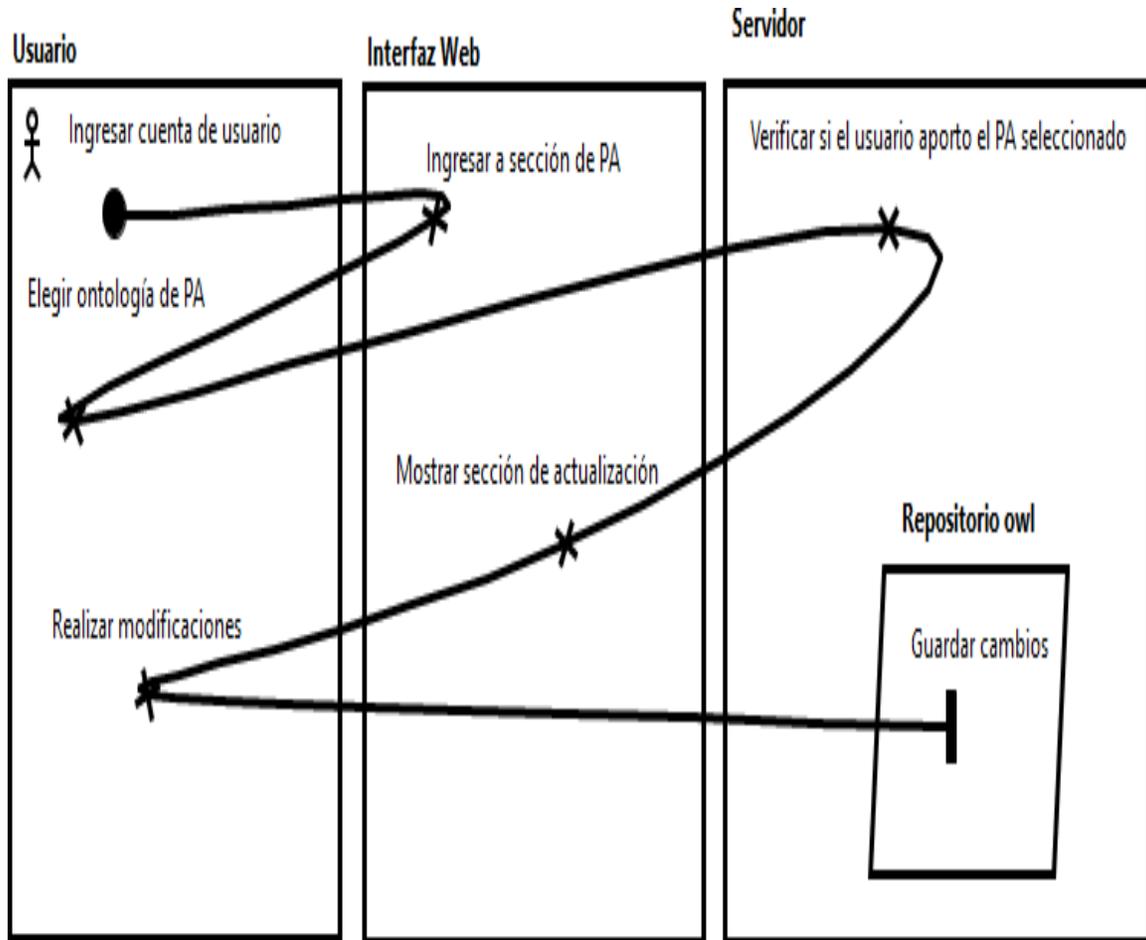


Figura 4. 8 Diagrama UCM meta 7. Actualizar información de cada ontología del diagrama GRL.

La eliminación de ontologías de patrones de análisis, meta 8. Se realiza a través del administrador de la aplicación, garantizando que usuarios maliciosos no eliminen las ontologías del repositorio. Al suprimir la ontología del patrón de análisis, se descartan los diagramas correspondientes a cada ontología. Revisar Figura 4.9.

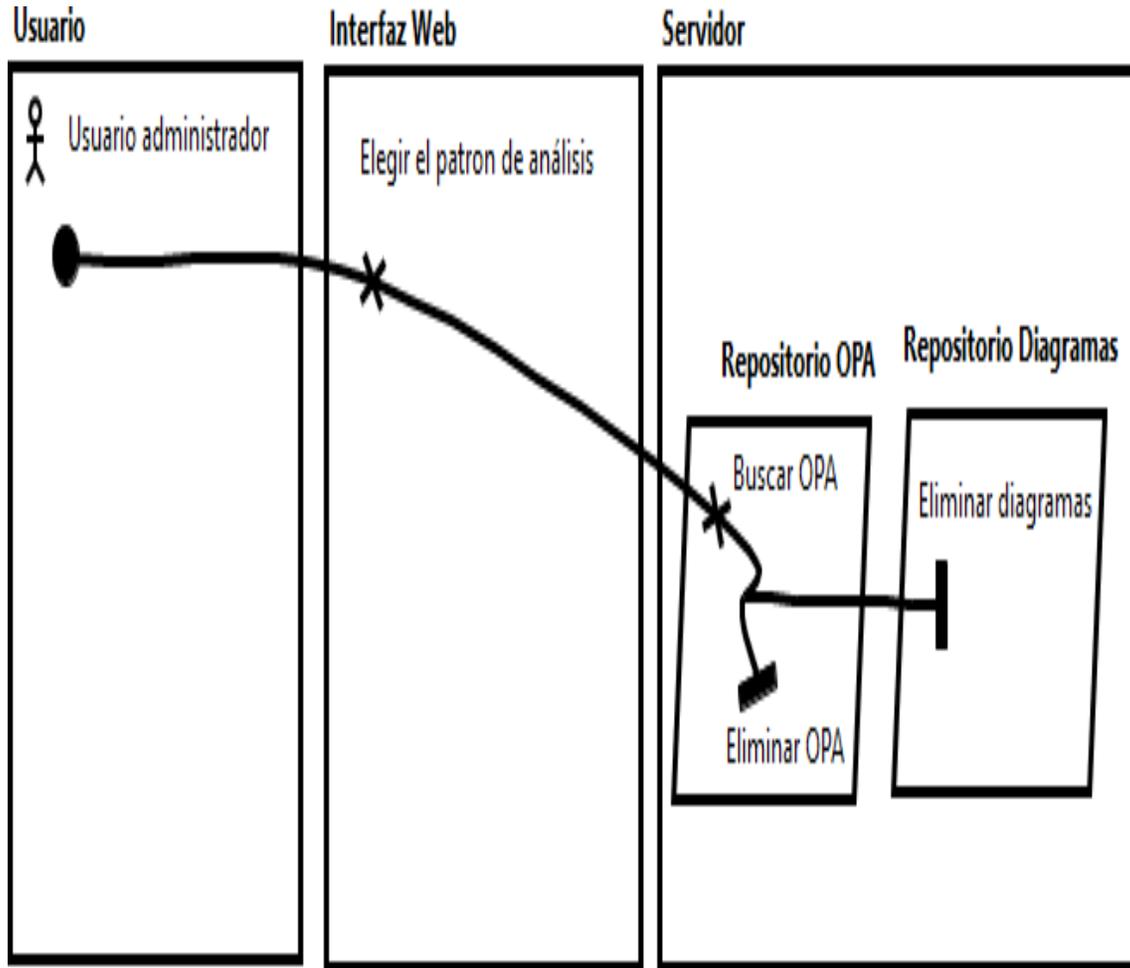


Figura 4. 9 Diagrama UCM meta 8.Eliminar ontologías del diagrama GRL.

Las metas blandas del 9 al 13 mostradas en la Figura 4.1, son los requerimientos no funcionales por lo cual no se genera su diagrama UCM. Y se definen en esta investigación de la siguiente manera:

9. Disponibilidad: Es la capacidad de un producto de software de permanecer en un estado para realizar una función requerida en cualquier momento, bajo condiciones de uso específicas (ISO/IEC 9126-1, 2001).

10. Seguridad: Es la capacidad de un producto de software para proteger la información y datos de forma que personas o sistemas no autorizados no puedan leerlos o modificarlos. Además, que personas y sistemas autorizados tengan acceso a los datos (ISO/IEC 9126-1,2001).

11. Mantenimiento: Son las características de un producto de software que lo hacen fácil de modificar. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software, para responder a cambios en el ambiente, las especificaciones funcionales y los requerimientos (ISO/IEC 9126-1, 2001).

12. Portabilidad: Son las características de un producto de software que facilita ser transferido de un ambiente a otro (ISO/IEC 9126-1, 2001).

13. Escalabilidad: Son las características que permiten extender un sistema de software con nuevas características y componentes, sin pérdida de funcionalidad o cualidades específicas como requerimientos (Henttonen, Matinlassi, Niemela, & Kanstrén, 2007).

4.1.2 Especificación textual de requerimientos funcionales.

Del análisis del diagrama GRL y los diagramas UCM se identificaron y se listan los requerimientos funcionales para la construcción del software que almacene patrones de análisis.

1. Crear cuentas de usuarios

- 1.1 Guardar el nombre del usuario, en caso de que el usuario cuente con varios nombres, sólo es necesario uno.
- 1.2 Guardar el apellido del usuario, sólo es necesario el primer apellido.
- 1.3 Guardar la dirección de correo electrónico.
- 1.4 Guardar una cadena de texto que fungirá como el nombre de la cuenta del usuario.
- 1.5 Guardar una contraseña para el nombre de la cuenta que el usuario proporcionará.

2. Representar la jerarquía de ontologías propuesta

- 2.1 Almacenar la ontología de la plantilla.
- 2.2 Almacenar individuos de la plantilla, estos son patrones descritos de acuerdo a la estructura de la ontología de la plantilla.

- 2.3 Almacenar ontologías de los patrones de análisis.
 - 2.4 Almacenar los individuos de cada ontología de patrón de análisis, estos son patrones de análisis descritos acorde a la estructura de cada ontología del patrón de análisis.
 - 2.5 Almacenar las ontologías de casos específicos de los individuos de cada patrón de análisis.
3. Realizar consultas al repositorio de ontologías.
- 3.1 Realizar consultas a través del nombre de una clase de la ontología (palabra o concepto), para localizar una ontología en particular.
 - 3.2 Realizar consultas a través de un texto breve, con esto se pretende que la aplicación señale las ontologías que contengan todas o algunas de las palabras del texto ingresado.
 - 3.3 Consultar información de una ontología a través del nombre de alguna propiedad.
4. Guardar diagramas XMI
- 4.1 Guardar los diagramas GRL correspondientes a cada descripción del patrón de análisis, en formato XMI.
 - 4.2 Guardar los diagramas UCM correspondientes a cada descripción del patrón de análisis, en formato XMI.
 - 4.3 Guardar los diagramas UML correspondientes a cada descripción del patrón de análisis, en formato XMI.
5. Descargar archivos
- 5.1 Descargar el archivo XMI de los diagramas GRL correspondientes a cada descripción del patrón de análisis, en formato XMI.
 - 5.2 Descargar el archivo XMI de los diagramas UCM correspondientes a cada descripción del patrón de análisis, en formato XMI.
 - 5.3 Descargar el archivo XMI de los diagramas UML correspondientes a cada descripción del patrón de análisis.

- 5.4 Descargar por separado cada ontología almacenada en un archivo con formato OWL.
6. Permitir que los usuarios registrados tengan la opción de aportar ontologías de patrones de análisis en la aplicación.
 - 6.1 Guardar nombre de la ontología.
 - 6.2 Guardar breve descripción acerca de la ontología
 - 6.3 Guardar el archivo owl de la ontología.
7. Permitir actualizar la información de cada ontología almacenada.
8. Permitir eliminar ontologías, sólo el administrador de la aplicación podrá realizarlo.

4.1.3 Requerimientos no funcionales

Los atributos del sistema de software son cualidades no funcionales que debe cumplir la aplicación a realizar. Del diagrama GRL que se muestra en la Figura 4.1 se identifican y se listan los requerimientos funcionales para la construcción del software que almacene patrones de análisis.

9. Disponibilidad. La aplicación web estará disponible en todo momento debido a que el servidor donde esté alojada la aplicación debe de estar siempre en ejecución.
10. Seguridad. El ambiente contará con autenticación de usuarios con el objetivo de conocer los usuarios que realizan aportaciones de ontologías de patrones de análisis al repositorio.
11. Mantenimiento. Se debe de tener cuidado de no desarrollar una estructura de control de flujo compleja e incomprensible, porque dificultaría el mantenimiento. Por lo regular la persona o personas que llevan a cabo esta tarea no es quien realizó el desarrollo.
12. Portabilidad. El ambiente debe ser desarrollado de forma que sea ejecutado en todas las plataformas. Se debe considerar configurar el servidor en una máquina virtual, de esta manera se asegura que el servidor se ejecute en diferentes plataformas utilizando un software virtualizador.
13. Escalabilidad. La aplicación se presentará como un trabajo de investigación a nivel maestría por lo que habrá diversos desarrolladores que irán enriqueciéndola, en investigaciones posteriores a ésta.

4.2. Adaptación de los requerimientos con Bioportal

En este apartado se describe el diseño de la infraestructura que almacenará ontologías de patrones de análisis, siguiendo la estructura propuesta en IEE 1016. La especificación del diseño se basa en el documento de especificación de requerimientos de software. Para satisfacer los requerimientos funcionales, se utilizan elementos y funcionalidades de la aplicación NCBO BioPortal.

4.2.1 Vista del contexto

Representa los servicios que provee el software en referencia a los actores que interactúan con él. Esta vista provee una perspectiva de caja negra.

4.2.1.1 Estructura de vista

Explicación de los actores que intervienen con el software.

3.2.1.2 Actores

Los actores que interactuarán con el ambiente son: usuario y administrador. Un usuario es la persona que accede a la aplicación para consultar o realizar aportaciones de ontologías de patrones de análisis. El administrador tiene los privilegios para realizar la funcionalidad disponible para un usuario y además tiene control sobre funcionalidad de gestión del ambiente (Ej. Puede eliminar ontologías).

4.2.2 Forma de satisfacer los requerimientos funcionales utilizando la aplicación BioPortal

Los requerimientos funcionales a satisfacer se determinaron en la sección 5.1 para la resolución del problema de refinamiento de descripciones de patrones de análisis siguiendo una jerarquía de ontologías que consta de tres niveles de abstracción y almacenar las ontologías de patrones de análisis en un repositorio que esté disponible al público.

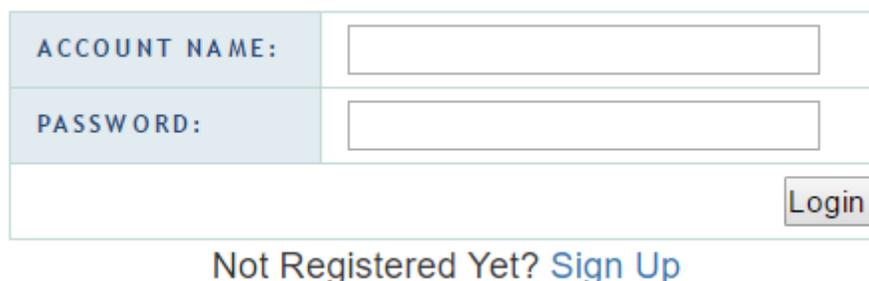
Para refinar la descripción de patrones de análisis se propuso tener por separado la estructura de la ontología de la plantilla estable de sus individuos, de igual manera se separó las ontologías de patrones de análisis de sus individuos, para más detalles observar la Figura 2.1. Para almacenar las ontologías de patrones de análisis se utiliza las funciones que ofrece la aplicación NCBO BioPortal de la Universidad de Stanford.

A continuación se describe el uso de elementos y funcionalidades del NCBO BioPortal para diseñar las soluciones que satisfacen cada uno de los requerimientos mostrados en la sección 5.1.2.

1. Crear cuentas de usuarios

- 1.1 Guardar el nombre del usuario, en caso de que el usuario cuente con varios nombres, sólo es necesario uno.
- 1.2 Guardar el apellido del usuario, sólo es necesario el primer apellido.
- 1.3 Guardar la dirección de correo electrónico.
- 1.4 Guardar una cadena de texto que fungirá como el nombre de la cuenta del usuario.
- 1.5 Guardar una contraseña para el nombre de la cuenta que el usuario proporcionará.

Para satisfacer este requerimiento, se empleará la funcionalidad que tiene BioPortal para registrar usuarios, en el portal de la aplicación seleccionar el apartado de *Sign In*, escrito con color azul, como se puede ver en la Figura 4.10.



ACCOUNT NAME:	<input type="text"/>
PASSWORD:	<input type="password"/>
<input type="button" value="Login"/>	
Not Registered Yet? Sign Up	

Figura 4. 10 Registrar usuarios.

2. Representar la jerarquía de ontologías propuesta

- 2.1 Almacenar la ontología de la plantilla.
- 2.2 Almacenar individuos de la plantilla, estos son patrones descritos de acuerdo a la estructura de la ontología de la plantilla.
- 2.3 Almacenar ontologías de los patrones de análisis.
- 2.4 Almacenar los individuos de cada patrón de análisis.
- 2.5 Almacenar las ontologías de los individuos de cada patrón de análisis.

Para cumplir este requerimiento se crearán categorías en la aplicación BioPortal, a través de la consola de Ruby (ver sección 5.6), para adecuar la jerarquía que se requiere. En la Figura 4.11 con un círculo de color rojo se indica la sección en BioPortal donde se ubicarán las categorías que representarán a la jerarquía de ontologías (después de crearlas).

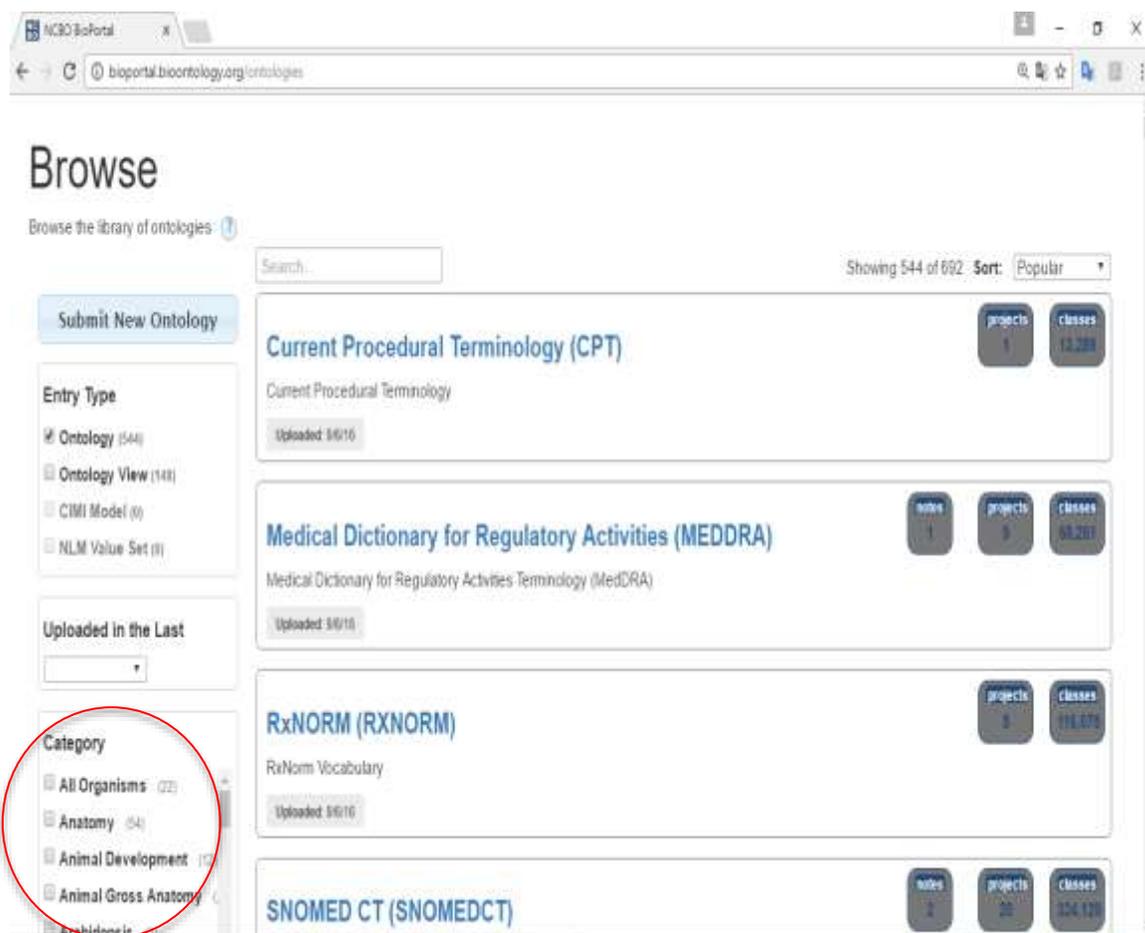


Figura 4. 11 Utilizar la sección de categorías para representar la jerarquía de ontologías.

3. Realizar consultas al repositorio de ontologías.

- 3.1 Realizar consultas a través del nombre de una clase de la ontología (palabra o concepto), para localizar una ontología en particular.

Por el momento solo se podrá llevar a cabo el requerimiento 3.1, por la complejidad de adaptar nuevo código fuente a la aplicación *NCBO*, ya que no se posee la documentación de diseño de su arquitectura. Por este motivo los requerimientos 3.2 y 3.3 se realizarán en futuras investigaciones. En la Figura 4.12 se muestra la funcionalidad de búsqueda de clases que proporciona la interfaz de BioPortal, la cual se utilizará para cumplir el requerimiento 3.1 mencionado en la sección 4.1.2. Para acceder a esta ventana, situarse en la interfaz de BioPortal y hacer clic en el menú desplegable *Tools*, y posteriormente dar clic en *Search*.



Figura 4. 12 Buscar una clase en múltiples ontologías, funcionalidad de BioPortal.

4. Descargar archivos

- 4.1 Descargar por separado cada ontología almacenada en un archivo con formato owl.

El requerimiento 5.4 (Descargar por separado cada ontología almacenada en un archivo con formato OWL) se ejecutará empleando la funcionalidad de la aplicación BioPortal. Los requerimientos 5.1, 5.2 y 5.3 que se listan en la sección 4.1.2 Especificación textual de requerimientos funcionales no se desarrollaran hasta que se realicen los requerimientos 4.1, 4.2 y 4.3 de esa misma sección, relacionados a guardar diagramas XMI. Estos requerimientos se desarrollarán en futuras investigaciones. En la Figura 4.13 se presenta la opción que tiene la interfaz de BioPortal para descargar archivos OWL señalada con un círculo de color rojo. Para acceder a esta ventana, situarse en la interfaz de BioPortal y hacer clic en el menú desplegable *Tools*, y posteriormente dar clic en *Ontology Browser*. En la interfaz de *Ontology Browser* elegir la ontología que desea descargar, haciendo clic sobre el nombre de la ontología, a continuación se presenta una interfaz con los detalles de la ontología seleccionada, en la parte inferior se localiza la sección que permite descargar el archivo de la ontología.

Submissions

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
2016-10-11 (Parsed, Indexed, Metrics, Annotator)	10/13/2016	10/13/2016	OWL CSV RDF/XML

Figura 4. 13 Descargar archivos OWL en BioPortal.

5. Permitir que los usuarios registrados tengan la opción de aportar ontologías de patrones de análisis en la aplicación.

- 5.1 Guardar nombre de la ontología.
- 5.2 Guardar breve descripción acerca de la ontología
- 5.3 Guardar el archivo *owl* de la ontología.

Estos requerimientos se cumplirán con la funcionalidad que brinda la pestaña de *Browser* de la aplicación NCBO BioPortal, que se menciona en el documento Descripción técnica de BioPortal (ver Anexo A). En la Figura 4.14 se exhibe la opción para ingresar una ontología en la interfaz de BioPortal, marcada con un círculo de color rojo.



Figura 4. 14 Agregar una ontología a BioPortal.

7. Permitir actualizar la información de cada ontología almacenada.

El requerimiento se ejecutará con la funcionalidad que proporciona la aplicación BioPortal, de actualizar ontologías. En la Figura 4.15 se presentan las tres opciones que brinda la interfaz de BioPortal para actualizar la información de la ontología, marcadas con una elipse de color rojo. Para acceder a esta ventana, situarse en la interfaz de BioPortal y pulsar el menú desplegable Tools, y posteriormente pulsar en el menú Ontology Browser. En la interfaz de Ontology Browser elegir la ontología que desea descargar, pulsando sobre el nombre de la ontología, a continuación se presenta una interfaz con los detalles de la ontología seleccionada, debajo del nombre de la ontología se localizan las opciones para editar la información acerca de la ontología.



Figura 4. 15 Actualizar la información de la ontología en BioPortal.

8. Permitir eliminar ontologías, solo el administrador de la aplicación podrá realizarlo.

Para la eliminación de una ontología, se ingresará al servidor de la aplicación NCBO BioPortal. Desde el servidor se utilizará la consola del lenguaje de programación Ruby. Los encargados de la aplicación NCBO BioPortal a dicha consola le llaman consola de ruby por lo que en este documento cada vez que se mencione la consola del lenguaje de programación de Ruby, nos referiremos a ella como consola de Ruby. Para conocer los pasos que se siguen para ingresar a la consola de Ruby, consultar la sección 5.7.1 del capítulo 5. En la Figura 4.16 se muestra la interfaz de la consola de Ruby.

```
[root@localhost ncbo_cron]# ./bin/ncbo_cron --console
(LD) >> Using rdf store localhost:8081
(LD) >> Using search server at http://localhost:8082/solr/core1
(LD) >> Using HTTP Redis instance at localhost:6380
(LD) >> Using Goo Redis instance at localhost:6382
(AN) >> Using ANN Redis instance at localhost:6379
(CNFG) >> OntologyRecommender not available, cannot load config
(CNFG) >> OntologiesAPI not available, cannot load config
(CR) >> Using Redis instance at localhost:6379
Starting ncbo_cron service...
Running ncbo_cron with redis: #<Redis client v3.2.1 for redis://localhost:6379/0
>
[1] pry(main)> _
```

Figura 4. 16 Consola de Ruby en servidor CentOS de la aplicación NCBO BioPortal.

En este capítulo se presentó el diagrama *GRL* y los diagramas *UCM* de cada una de las metas del diagrama *GRL* que especifican de manera visual los requerimientos funcionales, así mismo se listan los requerimientos funcionales y los no funcionales. Describir de manera textual y visual los requerimientos de un sistema, ayudan al desarrollador a tener una mejor comprensión sobre lo que se requiere que realice el sistema. En el capítulo siguiente se describe el proceso que se realizó para almacenar las descripciones de patrones de análisis y la semántica de estos utilizando la herramienta NCBO BioPortal.

Capítulo 5. Descripción del desarrollo

En este capítulo se describe el proceso realizado para almacenar ontologías de descripciones de patrones de análisis. En la sección 5.1 se expone la propuesta de solución para el almacenamiento de patrones de análisis. En la sección 5.2 se definen el nombre que se asignó a las cinco categorías para representar la jerarquía de ontologías de los patrones de análisis (Este requerimiento se lista en la sección 2.1.2, en el punto 2 de este documento), en la sección 5.3 se describe el proceso para crear una cuenta de usuario en la aplicación NCBO BioPortal, en la sección 5.4 se describen los pasos para ingresar una ontología al portal web de BioPortal por usuarios registrados. En la sección 5.5 se la manera ingresar un individuo en las categorías que se permiten agregar individuos.

En la sección 5.6 se explica el proceso para agregar una ontología siguiendo la estructura de información de nuestra solución propuesta. En la sección 5.7 se explica la manera de crear categorías en la aplicación *NCBO BioPortal*, dichas categorías se emplearon para representar la jerarquía de ontologías que se presenta en la Figura 5.1. En la sección 5.8 se menciona la forma de eliminar categorías, se prevé que en futuras investigaciones se requiera cambiar el nombre de alguna categoría o agregar más categorías en la aplicación. En la sección 5.9 se comenta la forma de crear y eliminar grupos (un grupo es un conjunto de ontologías similares o no y una categoría es la forma de clasificar ontologías de acuerdo a su utilidad o función que cumplen o desarrollan), para la resolución de nuestro problema no empleamos grupos, pero es muy posible que en futuras investigaciones hagan uso de los grupos. En la sección 5.10 se mencionan el nombre de las categorías y sus correspondientes acrónimos, conocer el acrónimo de la categoría es importante para eliminar la categoría. En la sección 5.11 se detalla el procedimiento para la eliminación de ontologías, si se requiere. Eliminar ontologías es una operación que requiere el privilegio de administrador de la aplicación.

Y por último, en la sección 5.12 se describe las modificaciones que son necesarias para enviar correos por parte del usuario desde el *Feedback* hacia el administrador de la aplicación. Esta es una opción de comunicación entre el usuario y el administrador de la aplicación, que mejora eficiencia, pues ahorra tiempo en ingresar a su cuenta de correo electrónico y perder un poco de tiempo. Cabe mencionar que en la copia de la aplicación NCBO BioPortal, que nos compartió la universidad de Stanford, no trabajan todas sus funcionales. Para que funcionara la función *Feedback*, por ejemplo, se tuvo que investigar para conocer líneas de código involucradas en la funcionalidad faltante.

5.1 Propuesta de solución

Para cumplir con el objetivo de este problema, se propuso utilizar las funcionalidades de la aplicación NCBO BioPortal la cual almacena ontologías biomédicas. Se adaptaron las funcionalidades de la aplicación NCBO BioPortal para el almacenamiento de descripciones de patrones de análisis, para que satisfaga los requerimientos de almacenar su semántica mediante una jerarquía de ontologías y sus individuos. Cabe señalar que el almacenamiento de las ontologías tiene como fin utilizar la semántica de los patrones de análisis para evitar problemas de ambigüedad y habilitar su formalización mediante lógicas descriptiva y temporal.

La idea general se muestra en la Figura 5.1, la cual consiste en almacenar, los tres niveles de abstracción siguientes:

I Nivel (Mayor abstracción):

- Ontología de la plantilla: contiene la ontología de plantilla estable de patrones de análisis (de utilidad para la captura de las descripciones de los patrones).
- Descripción de patrón de análisis: contiene la descripción textual del patrón de análisis como lo establece la plantilla estable (Ej. Patrón de Entidades Reusables).
- Archivos XMI: contienen los diagramas UML, GRL y UCM que abarca cada descripción de patrones de análisis.

II Nivel (Mediana abstracción):

- Ontología de patrón: contiene la ontología del patrón de análisis (enfocándose a los puntos 9 y 10 de la plantilla estable).
- Caso específico de un patrón: contiene la descripción textual del patrón de análisis en un caso específico como lo establece la plantilla estable (Ej. Instancia de un patrón de análisis para una empresa o institución, como la biblioteca del CENIDET).

III Nivel (Menor abstracción)

- Ontología del caso específico: contiene la ontología de un caso específico del patrón de análisis.

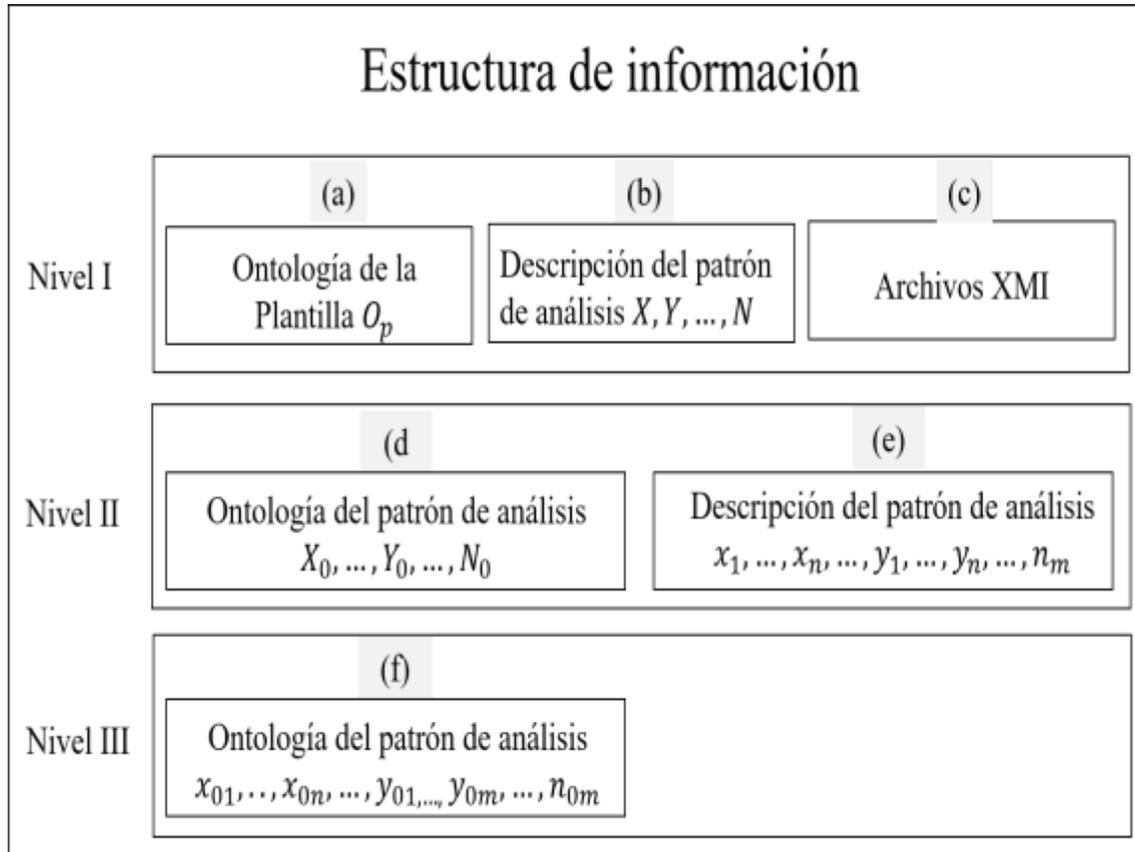


Figura 5. 1 Estructura de información

5.1.1 Cardinalidad de las relaciones que se observan en la Figura 5.1

En el almacén de información sólo existirá la ontología de la plantilla (Martínez, 2013) O_p (a) y esta tendrá uno o más individuos de la ontología de la plantilla, con las descripciones de patrones de análisis (b) X, Y, \dots, N . Para cada descripción (individuo) del patrón de análisis X, Y, \dots, N (b) descrito de acuerdo a la estructura de la ontología de la plantilla (Ej.1 $X =$ Descripción del patrón de análisis para reservación de entidades, Ej.2 $Y =$ Descripción del patrón de análisis de cuenta, etc.), existe una única ontología de patrón de análisis X_0, Y_0, \dots, N_0 (d) (Ej.1 $X_0 =$ Ontología del patrón de análisis para reservación de entidades, Ej.2 $Y_0 =$ Ontología del patrón de análisis de cuenta, etc.). Este tipo de ontologías se crean a partir de analizar la información que se encuentra en la ontología de descripción del patrón de análisis y utilizando la metodología MECOPA definida en (López, 2015).

Cada ontología de los patrones de análisis X_0, Y_0, \dots, N_0 (d) tendrá una o más descripciones de patrones de análisis ($x_1, \dots, x_n, y_1, \dots, y_n, \dots, n_m$) (e) (Ej. $x_1 =$ Biblioteca del CENIDET, $x_2 =$

Biblioteca del IIE, $y_1 = \text{Hotel los volcanes}$, $y_2 = \text{Hotel Holiday In}$, etc.). A cada descripción del patrón de análisis (Caso) $(x_1, \dots, x_n, y_1, \dots, y_n, \dots, n_m)$ (e), si existen diferencias con la ontología del patrón de análisis, le corresponderá una única ontología para ese caso particular (f), este tipo de ontologías representará las diferencias, en caso de haberlas, entre el caso y la ontología original, del patrón de análisis correspondiente (Ej. Diferencias del caso Biblioteca del CENIDET respecto al Patrón de análisis para reservación de entidades).

En la Figura 5.1 se observa un rectangulo con la leyenda archivos XMI (c), estos archivos contendrán diagramas UML, GRL y UCM. Actualmente en las ontologías de descripciones de patrones de análisis (X, Y, \dots, N) se incluyen la ubicación y nombre de los archivos XMI que contienen la representación de dichos tipos de diagramas. Pero en la versión resultante de esta investigación, no se incluye el almacenamiento de este tipo de archivos. Se propone que en investigaciones posteriores, el almacenamiento de este tipo de archivos, se incluirá en el almacén de patrones de análisis.

5.2 Definición de las categorías

Se especificaron cinco nombres de categorías en la aplicación NCBO BioPortal para adecuar la estructura de información requerida que se presenta en la Figura 5.1.

Los nombres de las cinco categorías son:

Para representar los tres niveles de abstracción de ontologías, que se muestran en la parte izquierda de la Figura 5.1 se definieron, las tres categorías siguientes:

- I-O Ontología de la Plantilla (a)
- II-O Ontologías de PAs (d)
- III-O Ontologías Casos Específicos de PAs (f)

Para representar los dos niveles de individuos b y e, que se muestran en la parte central de la Figura 3.1, se definieron, las dos categorías siguientes:

- I-I Descripciones de PAs (b)
- II-I Individuos de PAs (e)

En la sección 5.10 de este documento se describe el proceso para la creación de categorías en la aplicación NCBO.

La leyenda Archivos XMI que se encuentra en la parte c a la derecha de la Figura 5.1, se propone que se incluya en futuras investigaciones para completar la los elementos de la jerarquía de ontologías de patrones de análisis.

Los nombres de las siete categorías mencionadas anteriormente utilizan como prefijo:

I representa el primer nivel, **II** segundo nivel y **III** tercer nivel.

La letra **O** que se localiza después del guion que sigue al nivel, simboliza Ontología.

La letra **I** que se localiza después del guion que sigue al nivel, simboliza Individuo.

Nota: Las categorías individuos (-I) incluyen los valores de los atributos de nodos y relaciones de las ontologías. Así, las categorías de ontologías (-O) no contienen valores, comprenden solo la estructura y los metadatos de las ontologías correspondientes (separadas de los individuos).

En la Figura 5.2 se muestra las cinco categorías establecidas en la aplicación BioPortal.

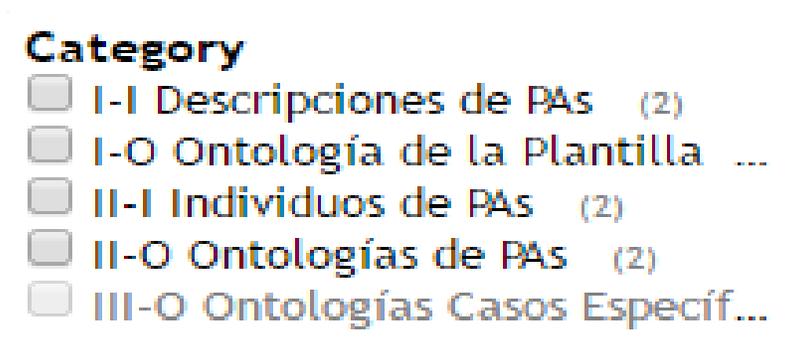


Figura 5. 2 Categorías en la aplicación BioPortal representando la estructura de información de la Figura 5.1.

Categoría I-O Ontología de la plantilla

La ontología ubicada en esta categoría es la Ontología para la Plantilla de Patrones de Análisis Estable (OPPAE) (Martínez, 2013), se utiliza de base para describir patrones de análisis de diferentes dominios.

Categoría I-I Individuos de la Plantilla

Las ontologías situadas en esta categoría son descripciones de patrones de análisis. Estas descripciones son individuos de la plantilla (que incluye valores para los atributos definidos en las clases y las relaciones de la ontología).

Categoría II-O Ontologías de PAs

Las ontologías localizadas en ambas categorías son el resultado de analizar la información que contienen las ontologías correspondientes en la categoría I-I Individuos de la Plantilla y el empleo de la metodología MECOPA (López, 2015). Cada ontología sirve de base para describir la semántica de la descripción del patrón de análisis correspondiente. Por el

momento, se cuenta con la ontología del Patrón de Análisis para la Reservación de Entidades (PARE) (López, 2015) y la ontología del Patrón de Análisis de Cuentas (PAC) (Júarez, 2016).

Categoría II-I Individuos de PAs

La categoría II-I Individuos de PAs tendrá descripciones del Patrón de Análisis para Reservación de Entidades, posiblemente en dominios diferentes (ejemplo 1, Individuos de PARE - Renta de video películas, Renta de cuartos en un hotel, Reserva de Libros, Reserva de Vuelos, etc., ejemplo 2, Individuos del PA PAC - Cuentas de una aplicación Web, Cuentas de usuarios de una biblioteca, etc.). Por el momento, solo existen ejemplos ficticios que muestran la apariencia que tendrán los individuos.

Categoría III-O Ontologías Casos Específicos de PAs

Cada ontología, ubicada en esta categoría, es el resultado de analizar la ontología del PA correspondiente, situada en la Categoría II-I Individuos de PAs, y son el resultado de aplicar la metodología MECOPA (López, 2015). Actualmente, no se cuenta con una ontología de caso específico, que contenga las diferencias respecto a los patrones de análisis PARE y PAC. En consecuencia no se incluyó una ontología de este tipo.

Hasta este momento, se tiene previsto que las ontologías ubicadas en la categoría III-O Ontologías Casos Específicos de PAs, describirán cambios en la semántica de un problema específico, respecto a la semántica de la ontología del nivel II-O. Un usuario menos experto en cierto dominio, puede utilizar una ontología de caso específico para entender el desempeño de un problema de un dominio particular.

Actualmente pensamos que el nivel III-Ontologías Casos Específicos de PAs es el último nivel de la jerarquía propuesta en la Figura 2.1 y que no se requerirá otro nivel de ontologías.

5.3 Crear cuenta de usuario

En esta sección se detalla el proceso para la creación de cuentas de usuario, en el portal *web* de BioPortal.

Primero, hacer clic en la leyenda Sign In que se localiza en la parte derecha de la franja azul del menú principal, contemple la Figura 5.3.

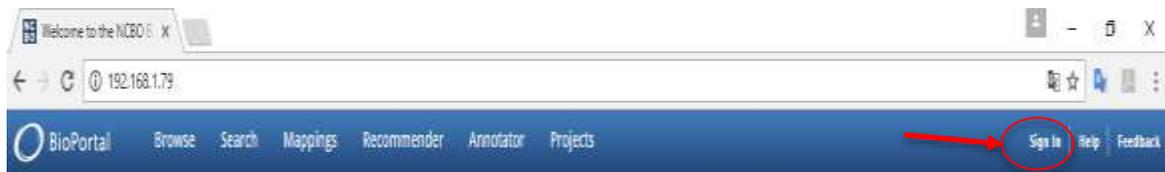
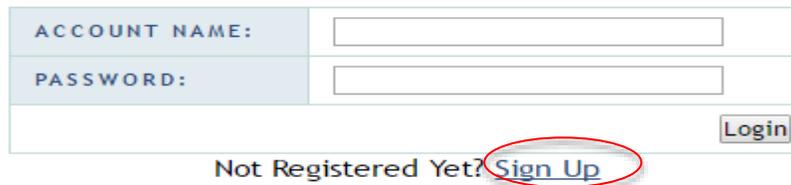


Figura 5. 3 Leyenda Sing In.

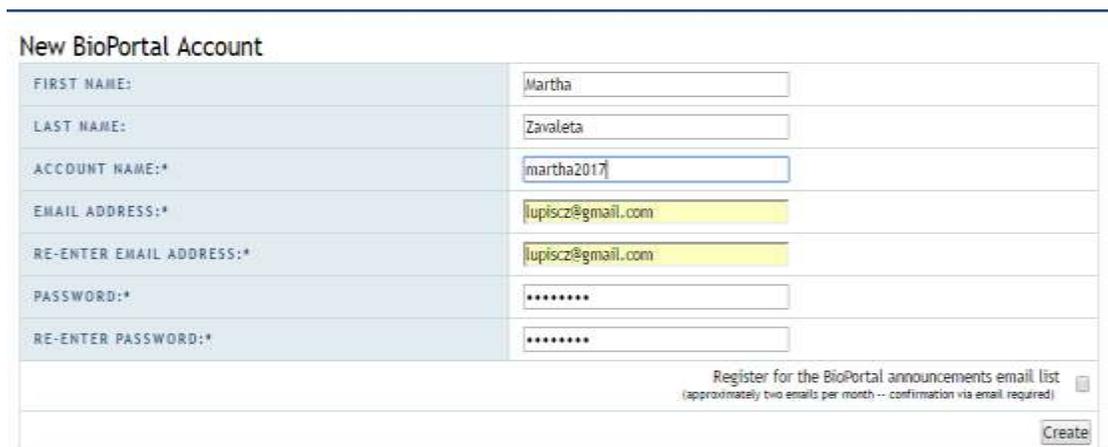
En la siguiente ventana que se muestra en la Figura 5.4, dar clic sobre la etiqueta Sign UP. .



ACCOUNT NAME:	<input type="text"/>
PASSWORD:	<input type="password"/>
<input type="button" value="Login"/>	
Not Registered Yet? Sign Up	

Figura 5. 4 Etiqueta Sign Up.

Finalmente se muestra una ventana solicitando el nombre del usuario, apellido, identificador de la cuenta, correo electrónico y contraseña, un ejemplo se muestra en la Figura 5.5. Al término del ingreso de los datos hacer clic en el botón Create.



New BioPortal Account	
FIRST NAME:	<input type="text" value="Martha"/>
LAST NAME:	<input type="text" value="Zavaleta"/>
ACCOUNT NAME:*	<input type="text" value="martha2017"/>
EMAIL ADDRESS:*	<input type="text" value="lupiscz@gmail.com"/>
RE-ENTER EMAIL ADDRESS:*	<input type="text" value="lupiscz@gmail.com"/>
PASSWORD:*	<input type="password" value="*****"/>
RE-ENTER PASSWORD:*	<input type="password" value="*****"/>
<input type="checkbox"/> Register for the BioPortal announcements email list <small>(approximately two emails per month -- confirmation via email required)</small>	
<input type="button" value="Create"/>	

Figura 5. 5 Ejemplo de los datos solicitados para crear una cuenta de usuario.

5.4 Agregar una ontología al portal web BioPortal

El proceso que se describe a continuación se utiliza para ingresar una ontología (sin valores en sus atributos), que en nuestro diseño se refiere a los archivos .owl que se incluirán en las categorías I-O Ontología de la Plantilla, II-O Ontologías de PAs o III-O Ontologías Casos Específicos de PAs. Este proceso también se utilizará para incluir individuos de las ontologías (con valores en sus atributos) que en nuestro diseño se refiere a los archivos .owl que se incluirán en las categorías I-I Descripciones de PAs o II-I Individuos de PAs.

Las categorías I-O, II-O y III-O, se separan de las categorías I-I y II-I, en nuestro diseño del almacenamiento, de manera diferente al almacenamiento que se realiza en los editores de ontologías (Ej. protégé). Pues en los editores de ontologías los individuos de las ontologías se almacenan sin diferenciarlos de su ontología, almacenando los valores de los atributos en el mismo archivo .owl.

En nuestro caso se considera útil separar las ontologías de sus individuos, pues las ontologías son estructuras generales (patrones) que pueden tener asociados varios individuos (versiones con valores en sus atributos), posiblemente de dominios diferentes.

En esta sección se explica el proceso para agregar archivos .owl, ontologías o individuos, al portal web BioPortal de la aplicación NCBO. Es importante que cuente con una cuenta de usuario de BioPortal, para crear una cuenta de usuario se requiere seguir las instrucciones de la sección 6.3 Crear cuenta de usuario. Si ya cuenta con una cuenta de usuario se sigue cada uno de los siguientes pasos.

1. Entrar al portal web BioPortal e ingresar su identificador de cuenta de usuarios. Como respuesta el portal BioPortal mostrará una ventana similar a la que se observa en la Figura 5.6.



Figura 5. 6 Ingresar a la pestaña Browser del portal web BioPortal.

2. Pulsar el botón *Submit New Ontology* el cual se localiza en la parte izquierda de la ventana de *Browser*, como se muestra en la Figura 5.7.

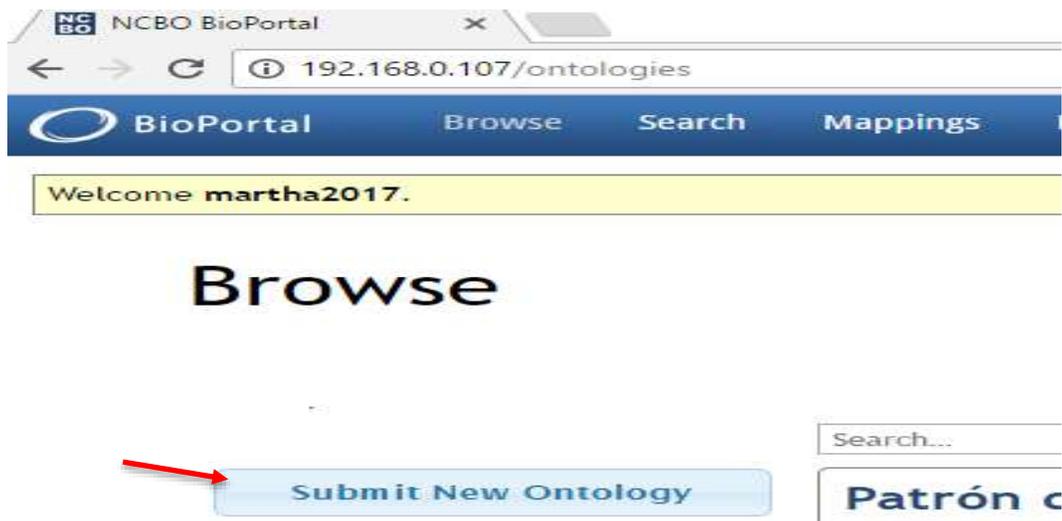


Figura 5. 7 Ubicación del botón Submit New Ontology.

3. Se da clic sobre el botón Submit New Ontology. A continuación se muestra una ventana, como la que se exhibe en la Figura 5.8.

NAME: *	<input type="text"/>
ACRONYM: *	<input type="text"/>
ADMINISTRATORS: *	<input type="text" value="martha2017"/>
VIEWING RESTRICTIONS	<input type="text" value="Public"/>
CATEGORIES:	<input type="text" value="Select category (domain)"/>
VIEW:	<input type="checkbox"/> This ontology is a view of: <input type="text" value="Select an ontology to create a view on"/>
<input type="checkbox"/> Subscribe to email notifications for new notes	
<input type="button" value="Cancel"/> <input type="button" value="Create ontology"/>	

Figura 5. 8 Ventana para el ingreso de nombre, acrónimo y elegir la categoría a la que pertenecerá la ontología agregar.

4. En la ventana que se exhibe en la Figura 5.8 se ingresan los datos solicitados. Como ejemplo ingresaremos los datos de una ontología a incluir en la categoría I-I Descripciones de PAs, esto es la descripción del patrón de análisis para la reservación de entidades (PARE) como se puede ver en la Figura 5.9. En este paso, se selecciona la categoría donde se desea incluir la ontología o el individuo de una ontología. Después de ingresar los datos solicitados en la ventana, dar clic sobre el botón Create Ontology.

The screenshot shows the 'Submit New Ontology' form in BioPortal. The form has a blue header with the BioPortal logo and navigation links: Browse, Search, Mappings, Recommender, Annotator, and Projects. The form title is 'Submit New Ontology' with a red asterisk and the text '* fields are required'. The form fields are:

- NAME:** Descripción de PARE
- ACRONYM:** DPARE
- ADMINISTRATORS:** martha2017
- VIEWING RESTRICTIONS:** Public
- CATEGORIES:** I-I Individuos de la Plantilla
- VIEW:** This ontology is a view of: Select an ontology to create a view on

At the bottom right, there is a checkbox for 'Subscribe to email notifications for new notes' and two buttons: 'Cancel' and 'Create ontology'. A red arrow points to the 'Create ontology' button.

Figura 5. 9 Ejemplo del ingreso de datos de una ontología.

5. A continuación se muestra una ventana solicitando una descripción breve acerca de la ontología, el tipo de formato de la ontología, la fecha de ingreso y la ubicación del archivo de la ontología, entre otras cosas. Al finalizar el ingreso de los datos solicitados en la ventana, dar clic en el botón Add Submission que se marca con una elipse de color rojo en la Figura 5.10

Capítulo 5. Descripción del Desarrollo

NG Add new ontology subm X

172.16.10.15/ontologies/DPARE/submissions/new

BioPortal Browse Search Mappings Recommender Annotator Projects

Add New Submission * fields are required

DESCRIPTION *	Se describe el patrón de análisis para <u>reservación</u> y uso de entidades re utilizables.												
FORMAT *	<p>OWL ▼</p> <table border="1"> <tr> <td>PREFERRED NAME PROPERTY default: http://www.w3.org/2004/02/skos/core#prefLabel</td> <td>use default</td> </tr> <tr> <td>SYNONYM PROPERTY default: http://www.w3.org/2004/02/skos/core#altLabel</td> <td>use default</td> </tr> <tr> <td>DEFINITION PROPERTY default: http://www.w3.org/2004/02/skos/core#definition</td> <td>use default</td> </tr> <tr> <td>AUTHOR PROPERTY default: http://purl.org/dc/elements/1.1/creator</td> <td>use default</td> </tr> </table> <p>Determining Obsolete Classes You can use the following two methods to identify obsolete classes in your ontology:</p> <ol style="list-style-type: none"> Designate a root ID for an obsolete branch. Identify a property id that indicates obsolete status. The value must be set to "true". <table border="1"> <tr> <td>1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)</td> <td></td> </tr> <tr> <td>2. PROPERTY TO CHECK FOR OBSOLETE STATUS</td> <td></td> </tr> </table>	PREFERRED NAME PROPERTY default: http://www.w3.org/2004/02/skos/core#prefLabel	use default	SYNONYM PROPERTY default: http://www.w3.org/2004/02/skos/core#altLabel	use default	DEFINITION PROPERTY default: http://www.w3.org/2004/02/skos/core#definition	use default	AUTHOR PROPERTY default: http://purl.org/dc/elements/1.1/creator	use default	1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)		2. PROPERTY TO CHECK FOR OBSOLETE STATUS	
PREFERRED NAME PROPERTY default: http://www.w3.org/2004/02/skos/core#prefLabel	use default												
SYNONYM PROPERTY default: http://www.w3.org/2004/02/skos/core#altLabel	use default												
DEFINITION PROPERTY default: http://www.w3.org/2004/02/skos/core#definition	use default												
AUTHOR PROPERTY default: http://purl.org/dc/elements/1.1/creator	use default												
1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)													
2. PROPERTY TO CHECK FOR OBSOLETE STATUS													
VERSION	1.0												
STATUS *	alpha ▼												
RELEASE DATE *	2017-02-16												
	<input type="radio"/> Metadata Only (choose this option if you want users to search and view only the metadata for your ontology, but not its classes and properties)												
LOCATION *	<input type="radio"/> Load from URL (new versions loaded daily)												
	<input checked="" type="radio"/> Upload Local File (choose a file on your local file system to upload)												
	<input type="button" value="Seleccionar archivo"/> PARE_Des...illa.owl												
CONTACT *	<table border="1"> <tr> <td>Name:</td> <td>Moisés González García</td> <td><input type="button" value="Add contact"/></td> </tr> <tr> <td>Email:</td> <td>moises@cenidet.edu.mx</td> <td></td> </tr> <tr> <td>Name:</td> <td>Ramiro Mar López Ramírez</td> <td></td> </tr> <tr> <td>Email:</td> <td>rmar@cenidet.edu.mx</td> <td></td> </tr> </table>	Name:	Moisés González García	<input type="button" value="Add contact"/>	Email:	moises@cenidet.edu.mx		Name:	Ramiro Mar López Ramírez		Email:	rmar@cenidet.edu.mx	
Name:	Moisés González García	<input type="button" value="Add contact"/>											
Email:	moises@cenidet.edu.mx												
Name:	Ramiro Mar López Ramírez												
Email:	rmar@cenidet.edu.mx												
HOME PAGE													
DOCUMENTATION PAGE													
PUBLICATIONS PAGE													
<input type="button" value="Add submission"/>													

Figura 5. 10 Continuación del ingreso de datos correspondientes a la ontología a ingresar.

6. Después de dar clic al botón *Add Submission* se muestra una ventana como la que se presenta en la Figura 5.11, que informa del ingreso satisfactorio de la ontología.

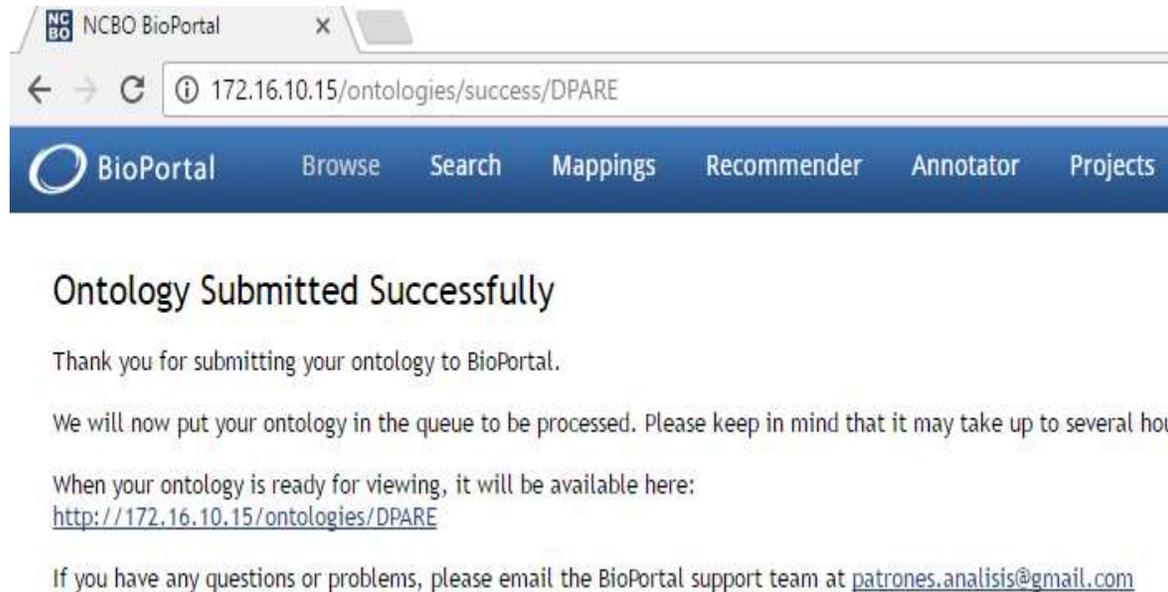


Figura 5. 11 Ontología ingresada satisfactoriamente.

5.5 Agregar una descripción de PA, mediante un individuo en la categoría I-I Descripciones de PA

En esta sección se explica el proceso para agregar un individuo en la categoría I-I Descripciones de PAs. Es importante que cuente con una cuenta de usuario de *BioPortal*, en la sección 5.6 Crear cuenta de usuario, se explica el proceso para crear una cuenta de usuario. Si ya cuenta con una cuenta de usuario se realizan cada uno de los pasos siguientes.

1. Entrar al portal *web BioPortal* e ingresar su identificador de cuenta de usuarios, como respuesta se mostrará una ventana similar a la que se observa en la Figura 5.12.



Figura 5. 12 Ingresar a la pestaña Browser del portal web BioPortal.

2. Elegir la categoría para este ejemplo utilizaremos la categoría I-O Ontología de la Plantilla. En la Figura 5.13 se muestra la categoría que se eligió.



Figura 5. 13 Seleccionar la categoría.

3. Dar clic sobre el nombre la categoría y sobre el nombre del archivo .owl deseado (en este caso: Ontología de la Plantilla de Patrones de Análisis Estable (OPPAE)). A continuación se presenta una interfaz similar a la que se exhibe en la Figura 5.14. En esta interfaz, se ubica la sección *DOWNLOADS* y se da clic sobre el enlace *OWL*, entonces comenzará a descargarse el archivo *owl* de la ontología de la plantilla.

Ontología de la Plantilla de Patrones de Análisis Estable
 Summary Classes Properties Notes Mappings Widgets

Details

ACRONYM	OPPAE
VISIBILITY	Public
DESCRIPTION	La ontología de la plantilla estable se utiliza para describir patrones de análisis.
STATUS	Beta
FORMAT	OWL
CONTACT	Moisés González García, moises@cenidet.edu.mx Rubí Celia Martínez Jiménez, ruby_martinez11@cenidet.edu.mx
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	I-O Ontología de la Plantilla
GROUPS	

Metrics

NUMBER OF CLASSES:	34
NUMBER OF INDIVIDUALS:	22
NUMBER OF PROPERTIES:	35
MAXIMUM DEPTH:	2
MAXIMUM NUMBER OF CHILDREN:	8
AVERAGE NUMBER OF CHILDREN:	3
CLASSES WITH A SINGLE CHILD:	0
CLASSES WITH MORE THAN 25 CHILDREN:	0
CLASSES WITH NO DEFINITION:	34

Visits
We are still collecting data for this ontology

Reviews Add your review
No reviews available.

Submissions

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
1.0 (Parsed, Indexed, Metrics, Annotator)	11/16/2016	11/16/2016	OWL CSV RDF/XML

Figura 5. 14 Ubicar la sección DOWNLOADS y dar clic sobre el enlace RDF/XML.

4. Para crear la descripción de un patrón de análisis se usa como entrada la ontología OPPAE. Para utilizar la ontología OPPAE e ingresar la descripción de un PA, se abre el archivo .owl descargado del portal BioPortal en el paso anterior, utilizando el editor Protégé. En el editor Protégé se puede editar el archivo de la ontología conforme a su estructura, modificando (ingresando) la información (los valores de los atributos) que describen a cada individuo (PA), y sus propiedades, como se muestra en la Figura 4.15.

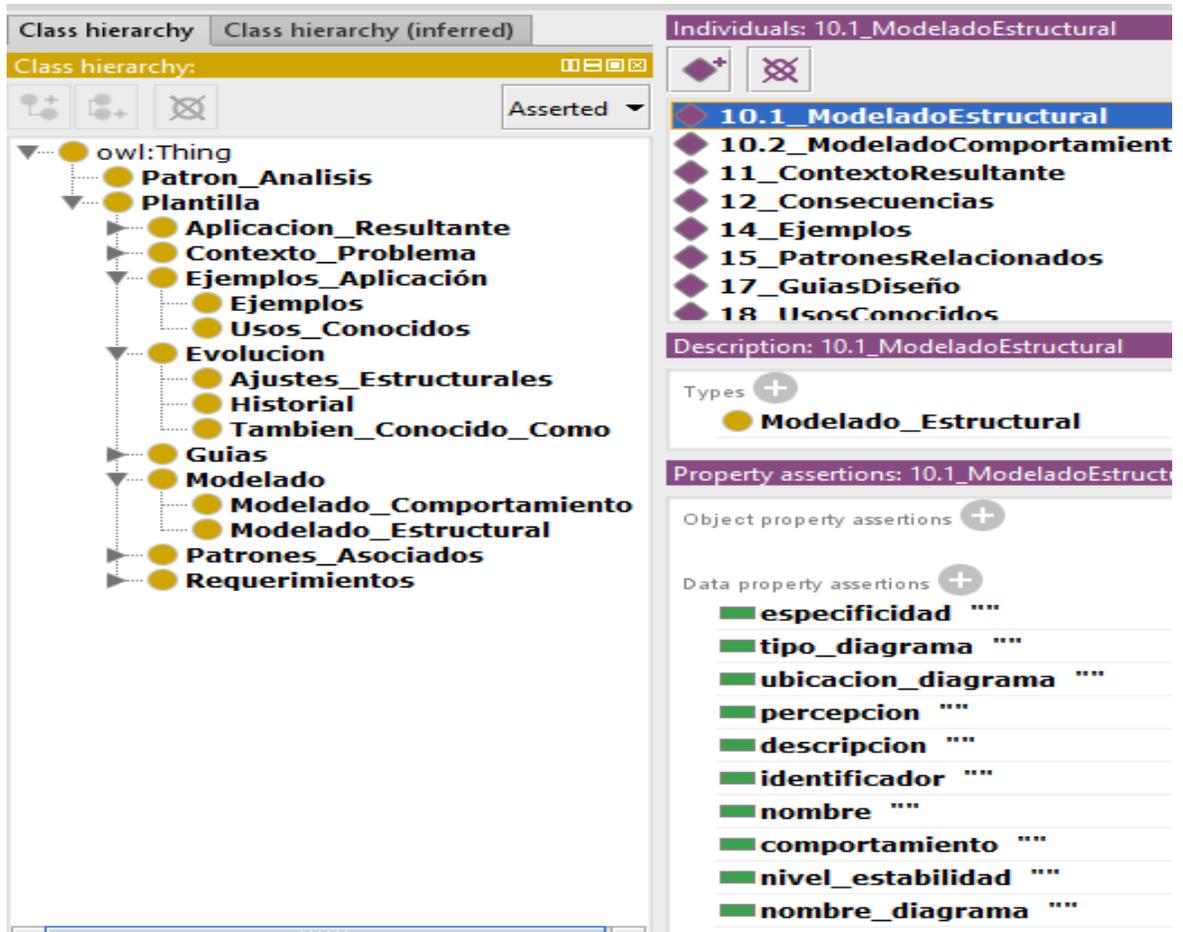


Figura 5. 15 Archivo de la ontología de la Plantilla abierto en Protégé.

5. Cuando haya terminado de editar el archivo, como se describe en el punto 4, guarde los cambios y asigne un nuevo nombre al archivo (Ej. I-I Nombre del PA descrito = I-I PA de Reservación de Entidades (PARE)). Para guardar la nueva ontología en el repositorio revise la sección 3.4 Agregar una ontología al portal web BioPortal de este documento.

5.6 Agregar una ontología o individuo en alguna de las categorías existentes.

En esta sección se explica el proceso para agregar una ontología o individuo en alguna categoría en la jerarquía de nuestra solución propuesta (Figura 5.1). En general se deben realizar los cinco pasos mencionados en la sección 5.5, aunque cambian: 1) el archivo .owl que se baja en el paso 3 y 2) el proceso del paso 4, que se sigue para producir el archivo .owl que se agregará.

Caso 1.- Para agregar un individuo en la categoría I-I Descripciones de PA, el proceso se describió en la sección 3.5.

Caso 2.- Para producir los archivos que se incluirán en la categoría II-O Ontologías de PAs, se bajará el archivo .owl de la descripción del PA seleccionado, desde la categoría I-I Descripciones de PAs, en el paso 3. Para crear la ontología del PA seleccionado, se deberá seguir la metodología para creación de ontologías como la MECOPA (López, 2015) o MEVASE (Júarez, 2016), en el paso 4.

Caso 3.- Para producir los archivos que se incluirán en la categoría II-I Individuos de PAs, se bajará el archivo .owl de la ontología del PA seleccionado, desde la categoría II-O Ontologías de PAs, en el paso 3. En el paso 4, se utiliza el archivo .owl bajado y se ingresan los valores de sus atributos. Para hacerlo se abre el archivo .owl utilizando el editor Protegé. En el editor Protegé se puede editar el archivo de la ontología conforme a su estructura, modificando (ingresando) la información (los valores de los atributos) que describen a cada individuo del PA, y sus propiedades.

Caso 4.- Para producir los archivos que se incluirán en la categoría III-O Ontologías de Casos Específicos de PAs, se bajará el archivo .owl del individuo del PA seleccionado, desde la categoría II-I Individuos de PAs, en el paso 3. Para crear la ontología del PA seleccionado, se deberá seguir una metodología para creación de ontologías como la MECOPA (López, 2015) o MEVASE (Júarez, 2016), para describir los elementos que difieren de lo especificado en la ontología del PA, en el paso 4.

5.7 Creación de categorías en BioPortal

A continuación se describe la manera de ingresar a la consola de Ruby en el servidor de la aplicación con la intención de guiar a la persona que requería hacer uso para replicar lo realizado hasta el momento, o para la realización de otras acciones que se necesiten.

5.7.1 Ingresar a la consola de Ruby en el servidor de BioPortal

Para crear (sección 5.7.2) y eliminar (5.8) una categoría en la aplicación NCBO BioPortal, se requiere entrar al servidor de la aplicación y seguir los siguientes pasos:

1. En la consola, entrar en la carpeta *ncbo_cron* con el comando: `cd /srv/ncbo/ncbo_cron`, como se puede ver en la Figura 5.16.

```
[root@localhost ~]# cd /srv/ncbo/ncbo_cron
[root@localhost ncbo_cron]# _
```

Figura 5. 16 Comando para ingresar en la carpeta *ncbo_cron*.

2. Ubicado dentro de la carpeta *ncbo_cron*, se escribe el comando `./bin/ncbo_cron --console` (antes de *console* se escriben dos `-`, `--`). Este comando permite hacer uso de la consola del lenguaje de programación *Ruby*. Si no se escribe correctamente el comando, se muestra una serie de comandos similares a los que se tiene que mostrar, pero no se ejecutaran los comandos que se requiera ingresar y se despliega un error. En la Figura 5.17 se muestran los comandos que se despliegan, cuando el comando de salida no se escribe correctamente.

```
[root@localhost ncbo_cron]# bin/ncbo_cron
(LD) >> Using rdf store localhost:8081
(LD) >> Using search server at http://localhost:8082/solr/core1
(LD) >> Using HTTP Redis instance at localhost:6380
(LD) >> Using Goo Redis instance at localhost:6382
(AN) >> Using ANN Redis instance at localhost:6379
(CNFG) >> OntologyRecommender not available, cannot load config
(CNFG) >> OntologiesAPI not available, cannot load config
(CR) >> Using Redis instance at localhost:6379
Log file: /srv/ncbo/ncbo_cron/logs/scheduler.log
Process is already started
[root@localhost ncbo_cron]# ontology = LinkedData::Models::Ontology.find("MUCHO"
).first
-bash: syntax error near unexpected token `('
[root@localhost ncbo_cron]# _
```

Figura 5. 17 Comando escrito incorrectamente.

En caso de escribirse correctamente el comando de entrada, se debe desplegar un mensaje como el que se muestra en la Figura 5.18.

```
[root@localhost ncho_cron]# ./bin/ncho_cron --console
(LD) >> Using rdf store localhost:8081
(LD) >> Using search server at http://localhost:8082/solr/core1
(LD) >> Using HTTP Redis instance at localhost:6380
(LD) >> Using Goo Redis instance at localhost:6382
(AN) >> Using ANN Redis instance at localhost:6379
(CNFG) >> OntologyRecommender not available, cannot load config
(CNFG) >> OntologiesAPI not available, cannot load config
(CR) >> Using Redis instance at localhost:6379
Starting ncho_cron service...
Running ncho_cron with redis: #<Redis client v3.2.1 for redis://localhost:6379/0
>
[1] pry(main)> _
```

Figura 5. 18 Comando para entrar a la consola de Ruby.

Nota: Para salir de la consola de Ruby oprimir las teclas Ctrl + D.

5.7.2 Crear categorías en BioPortal

Estando en la consola de Ruby, el procedimiento para crear categorías es el siguiente.

1. En la consola de Ruby escribir el comando `category = LinkedData::Models::Category.new`, considere la Figura 5.19.

```
[1] pry(main)> category = LinkedData::Models::Category.new
=> #<LinkedData::Models::Category:0x007f7c264c4e70
  @aggregates=nil,
  @id=nil,
  @loaded_attributes=#<Set: {}>,
  @modified_attributes=#<Set: {}>,
  @persistent=false,
  @previous_values=nil,
  @unmapped=nil>
```

Figura 5. 19 Comando para crear categorías.

2. En seguida, se escribe el comando `category.name = "Nombre_Categoría"`, lo escrito en comillas varia de acorde al nombre de cada categoría que se desee crear, en la Figura 5.20 se presenta como se creó la categoría I-O Ontología de la Plantilla.

```
[2] pry(main)> category.name = "I-O Ontología de la Plantilla"
=> "I-O Ontología de la Plantilla"
[3] pry(main)> |
```

Figura 5. 20 Asignar nombre a la categoría.

3. En seguida, se teclea el comando `category.acronym = "Abreviatura de la Categoría"`, lo escrito en comillas varia conforme al acrónimo que se le concede a la categoría, en la Figura 5.21 se muestra el acrónimo que se le asignó a la categoría I-O Ontología de la Plantilla.

```
[3] pry(main)> category.acronym = "Ont_Plant"
=> "Ont_Plant"
[4] pry(main)> |
```

Figura 5. 21 Asignar acrónimo a la categoría.

4. Y Por último, para guardar el nombre y el acrónimo de la categoría, se escribe el comando `category.save`, revise la Figura 5.22.

```
[4] pry(main)> category.save
=> #<LinkData::Models::Category:0x007fe6117e7468
  @acronym="Ont_Plant",
  @aggregates=nil,
  @created=
    #<DateTime: 2017-02-20T08:57:13-08:00 ((2457805j, 61033s, 418251878n), -28800s, 2299161j)>,
  @errors={},
  @id=
    #<RDF::URI:0x3ff308b59484(http://data.bioontology.org/categories/Ont_Plant)>,
  @loaded_attributes=
    #<Set: {:acronym, :name, :description, :created, :parentCategory}>,
  @modified_attributes=#<Set: {}>,
  @name="I-O Ontología de la Plantilla",
  @persistent=true,
  @previous_values=nil,
  @unmapped=nil>
[5] pry(main)> |
```

Figura 5. 22 Guardar nombre y acrónimo de la categoría.

5.8 Eliminar categorías

La eliminación de categorías en la aplicación NCBO BioPortal, se realiza a través de la consola del lenguaje de programación *Ruby*.

Para ingresar a la consola de *Ruby* revisar la sección 5.7.1. Para eliminar una categoría se necesita el acrónimo de la categoría. Por este requerimiento, en la sección 5.10 se presentan los nombres, con su respectivo acrónimo, de las categorías que se han creado en la aplicación NCBO BioPortal, para representar la estructura de información de la Figura 5.1 que se muestra en la sección 5.1.

Para conseguir la eliminación de la categoría, efectuar los siguientes pasos:

1. En la consola de *Ruby* escribir el siguiente comando:

`category = LinkedData::Models::Category.find("Acronimo de la Categoría").first`, en la Figura 2.23 se presenta la eliminación de la categoría I-O Ontología de la Plantilla.

```
[1] pry(main)> category = LinkedData::Models::Category.find("Ont_Plant").first
=> #<LinkedData::Models::Category:0x007f5081077050
  @aggregates=nil,
  @id=
    #<RDF::URI:0x3fa84083b9f4(http://data.bioontology.org/categories/Ont_Plant)>,
  @loaded_attributes=#<Set: {}>,
  @modified_attributes=#<Set: {}>,
  @persistent=true,
  @previous_values=nil,
  @unmapped=nil>
[2] pry(main)> _
```

Figura 5. 23 Comando para buscar una categoría.

2. Escribir el comando `category.delete`, en la Figura 2.24, se muestra una lista de variables, dentro de las cuales, la variable de nombre `@persistent = false`, indica que la categoría ha sido eliminada. Para que los cambios se reflejen BioPortal, se debe reiniciar el servidor.

```
[2] pry(main)> category.delete
=> #<LinkData::Models::Category:0x007f5081077050
  @acronym="Ont_Plant",
  @aggregates=nil,
  @created=
    #<DateTime: 2016-11-09T04:44:37-08:00 ((2457702j,45877s,0n),-28800s,2299161j)>
,
  @description=nil,
  @id=
    #<RDF::URI:0x3fa84083b9f4(http://data.bioontology.org/categories/Ont_Plant)>,
  @loaded_attributes=
    #<Set: {:acronym, :name, :description, :created, :parentCategory}>,
  @modified=true,
  @modified_attributes=#<Set: {}>,
  @name="I-O Ontología de la Plantilla",
  @parentCategory=nil,
  @persistent=false,
  @previous_values=nil,
  @unmapped=nil>
[3] pry(main)> _
```

Figura 5. 24 Comando para eliminar una categoría.

5.9 Crear grupos

En BioPortal, así como se pueden utilizar Categorías, también se pueden usar grupos, aunque no se requirió usarlos, para nuestra jerarquía de ontologías e individuos. Pero se prevé que en futuras investigación se requiera hacer uso ellos. Un grupo es un conjunto de ontologías, similares o no, y una categoría es la forma de clasificar ontologías de acuerdo a su utilidad o función que cumplen o desarrollan.

Para establecer un grupo en la aplicación *NCBO BioPortal*, entrar al servidor de la aplicación y seguir los siguientes pasos:

1. En la consola de Ruby escribir el comando `group = LinkData::Models::Group.new`

2. Posterior a esto, se escribe el comando `group.name = "Nombre_Grupo"`, lo escrito en comillas varia de acorde al nombre de cada grupo que se desee crear.

3. A continuación, teclear el comando `group.acronym = "Abreviatura del Grupo"`, lo escrito en comillas varia de acorde al nombre del acrónimo que se le asignó al grupo.

4. Y Por último para guardar el nombre y el acrónimo del grupo, se escribe el comando `group.save`.

Nota: para eliminar un grupo se sigue un proceso semejante al de eliminar una categoría.

5.10 Nombre y acrónimos de las categorías

Los nombres y acrónimos que se les asignó a las categorías creadas en la aplicación NCBO BioPortal son:

1. **Nombre:** I-O Ontología de la Plantilla
Acrónimo: Ont_Plant
2. **Nombre:** I-I Descripciones de PAs
Acrónimo: Descrip_PA
3. **Nombre:** II-O Ontologías de PAs
Acrónimo: Ont_PA
4. **Nombre:** II-I Individuos de PAs
Acrónimo: Ind_PA
5. **Nombre:** III-O Ontologías Casos Específicos de PAs
Acrónimo: Ont_CaEs

5.11 Eliminar ontologías

La eliminación de ontologías en la aplicación NCBO BioPortal, se realiza a través de la consola del servidor de la aplicación y esta opción sólo la puede realizar el administrador de la aplicación. Para conseguir la eliminación de la ontología, el administrador efectúa los siguientes pasos:

1. En la consola de Ruby escribir el comando:

`ontology = LinkedData::Models::Ontology.find("Acronimo de la Ontología").first`, en la Figura 5.25 se muestra la eliminación de la ontología de lantilla a través del acrónimo que se le asigno.

```
[1] pry(main)> ontology = LinkedData::Models::Ontology.find("OPPAE").first
=> #<LinkedData::Models::Ontology:0x007f53b2956e08
  @aggregates=nil,
  @id=#<RDF::URI:0x3fa9d94b6050 (http://data.bioontology.org/ontologies/OPPAE)>,
  @loaded_attributes=#<Set: {}>,
  @modified_attributes=#<Set: {}>,
  @persistent=true,
  @previous_values=nil,
  @unmapped=nil>
```

Figura 5. 25 Comando para encontrar una ontología.

2. Escribir el comando *ontology.delete*, para eliminar la ontología, en la Figura 5.26 se observa una lista de variables, dentro de las cuales, la variable de nombre *@unmapped = nil*, indica que la ontología con acrónimo OPPAE ya no pertenece a una categoría y no tiene un usuario asociado por lo tanto la ontología se ha eliminado del repositorio. Para que los cambios se reflejen el portal web BioPortal, se debe reiniciar el servidor.

```
[2] pry(main)> ontology.delete
=> #<LinkData::Models::Ontology:0x007f53b2956e08
  @acl=[],
  @acronym="OPPAE",
  @administeredBy=
  [#<LinkData::Models::User:0x007f53b1fa0210
    @aggregates=nil,
    @id=#<RDF::URI:0x3fa9d8fd0f04 (http://data.bioontology.org/users/admin)>,
    @loaded_attributes=#<Set: {}>,
    @modified_attributes=#<Set: {}>,
    @persistent=true,
    @previous_values=nil,
    @unmapped=nil>],
  @aggregates=nil,
  @doNotUpdate=nil,
  @flat=nil,
  @group=[],
  @hasDomain=
  [#<LinkData::Models::Category:0x007f53b1f9f5b8
    @aggregates=nil,
    @id=#<RDF::URI:0x3fa9d8fd0b08 (http://data.bioontology.org/categories/Ont_Plant)>,
    @loaded_attributes=#<Set: {}>,
    @modified_attributes=#<Set: {}>,
    @persistent=true,
    @previous_values=nil,
    @unmapped=nil>],
  @id=#<RDF::URI:0x3fa9d94b6050 (http://data.bioontology.org/ontologies/OPPAE)>,
  @loaded_attributes=
  #<Set: {:notes,
    :reviews,
    :subscriptions,
    :projects,
    :provisionalClasses,
    :submissions,
    :acronym,
    :views,
    :name,
    :administeredBy,
    :group,
    :viewingRestriction,
    :doNotUpdate,
    :flat,
    :hasDomain,
```

Figura 5. 26 Comando para eliminar una ontología.

5.12 Enviar correos electrónicos desde Feedback del portal de BioPortal

Enviar correos por parte del usuario desde el Feedback hacia el administrador de la aplicación, es una opción de comunicación entre el usuario y el administrador de la aplicación, que mejora la eficiencia, pues ahorra tiempo en ingresar a su cuenta de correo electrónico y perder un poco de tiempo. Cabe mencionar que la copia de la aplicación NCBO BioPortal, que nos compartió la universidad de Stanford, no trabajan todas sus funcionales. Por ejemplo para que funcionara la función Feedback, se tuvo que investigar para conocer líneas de código involucradas en la funcionalidad faltante.

Para el envío de correos electrónicos a través de Feedback, se realizan los siguientes ajustes en el archivo *bioportal_config.rb*, ubicado en la siguiente ruta:

```
srv/rails/BioPortal/releases/bioportal_web_ui/config/bioportal_config.rb
```

Dentro del archivo *bioportal_config.rb* localizar la línea #Email setting, debajo de esta línea agregar lo siguiente:

```
ActionMailer::Base.smtp_setting = {  
  :address      => "smtp.gmail.com",  
  :port         => 587,  
  :domain       => "gmail.com",  
  :user_name    => "patrones.analisis@gmail.com",  
  :password     => 'contraseña_del_correo_electronico'  
  :authentication => "plain",  
  :enable_starttls_auto => true  
}
```

Las líneas de código, se obtuvieron buscando en internet, la configuración que nos permita enviar correos electrónicos a través de una aplicación, utilizando el lenguaje de programación *Ruby on Rails*. En Gmail, se intentó obtener nuestro propio dominio, pero no se cuenta con recursos económicos por lo, que se optó por crear una cuenta de correo electrónico de manera gratuita.

En el mismo archivo ubicar la línea *#Email addresses used for sending notifications (errors, feedback, support)* y se escriben después de ella, las siguientes líneas:

```
$SUPPORT_EMAIL = "patrones.analisis@gmail.com"  
$ADMIN_EMAIL   = "patrones.analisis@gmail.com"  
$ERROR_EMAIL    = "patrones.analisis@gmail.com"
```

Con esto se consigue que los correos que se envíen a los usuarios, va enviada una copia al administrador de la aplicación, el nombre del correo se puede cambiar, en este caso se utilizó el mismo correo para dar soporte a los usuarios, como correo del administrador y para enviar errores.

En este capítulo se explicó el proceso que se realizó para almacenar descripciones de patrones de análisis y la semántica de estos, utilizando la aplicación NCBO BioPortal. Con la meta de lograr el almacenamiento de la jerarquía propuesta, se analizó la manera de almacenar ontologías del editor Protegé y se determinó que solo utiliza un archivo para la ontología para la Plantilla de Patrones de Análisis Estable y los dos individuos con los que actualmente se cuenta.

El almacenamiento en un solo archivo de la ontología de la plantilla y sus individuos, dificultaría la consulta de las descripciones de los PAs, por lo que se optó por separarlo en tres: uno que contenga la ontología de la plantilla y cada uno de los dos archivos restantes para los individuos de la plantilla.

La estructura de información propuesta requiere que en cada nivel se tengan los archivos por separado y no todos en un solo archivos. Esto evita confusión al momento de consultar y utilizar cada archivo. Es decir que se puedan consultar por separado: 1) Las descripciones de los PAs, 2) las ontologías de los PAs, 3) los individuos de los PAs, y 4) las diferencias de los individuos respecto a sus ontologías.

La jerarquía de ontologías orienta al usuario a conocer el nivel de abstracción en el que se localiza cierta ontología:

- Los individuos de la ontología para la Plantilla de Patrones de Análisis Estable son las descripciones de los patrones de análisis, acorde a la estructura de la plantilla. Por ejemplo: La descripción del Patrón de Análisis para la Reutilización de Entidades (PARE), o la descripción del Patrón de Análisis de Cuenta (PAC).
- Las ontologías de los PAs, obtenidas a partir de sus descripciones, representa la semántica de los conceptos (clases) y las relaciones entre ellos, así como sus propiedades y restricciones. Por ejemplo: la ontología de PARE representa a sus clases (Ej. Reservación, Persona, Cliente, Empresa, Entidad, etc.), sus relaciones (Ej. Cliente

reserva Entidad, Empleado crea Reservación, etc.), sus propiedades (Ej. Entidad tiene los atributos nombre, identidad, disponibilidad, etc.), etc.

- Los individuos de la ontología de un patrón de análisis, son la descripción del patrón de análisis conforme a la estructura de sus clases, relaciones, propiedades y restricciones, para un caso específico. Por ejemplo, el PARE con los nombres de clase, relaciones y propiedades, así como los valores de las propiedades, para un caso específico (el préstamo de libros de la biblioteca de la UAEM).

En el capítulo siguiente se presentan los resultados obtenidos de las pruebas realizadas a los requerimientos funcionales que se especificaron en la sección 4.2.2.

Capítulo 6. Resultados y pruebas

En este capítulo se presentan los resultados obtenidos de las pruebas realizadas acorde al diseño de pruebas realizado bajo la norma de la IEEE para documentación de pruebas de software y sistemas (IEEE Computer Society, 2008). También se describen las pruebas realizadas a los requerimientos que se determinaron.

6.1 Resultados generales

En la Tabla 6.1 se exhiben los resultados de las pruebas obtenidos en esta investigación, debajo de la columna con el nombre de Requerimientos se listan los requerimientos, de los mostrados en la sección 4.2.2, satisfechos en esta investigación. En la columna con el nombre de Caso de Prueba, se listan los nombres que se asignaron a cada uno de los siete casos de prueba resultantes y en la columna con el nombre de Resultado de la prueba, se presenta el resultado de cada caso de prueba. En la tercera columna se observa que todas las pruebas fueron exitosas.

Tabla 6. 1 Resultados obtenidos.

Requerimiento	Caso de Prueba	Resultado de la prueba
1. Crear cuentas de usuario	TC-01	Exitoso
2. Representar la jerarquía de ontologías propuesta	TC-02	Exitoso
3. Realizar consultas al repositorio de ontologías	TC-03	Exitoso
4. Descargar archivos	TC-04	Exitoso
5. Aportar ontologías por usuarios registrados	TC-05	Exitoso
6. Actualizar información de cada ontología	TC-06	Exitoso
7. Eliminar ontología	TC-07	Exitoso

6.1.1 Resultados detallados

Se ha referido anteriormente la concesión de la máquina virtual de la aplicación *NCBO* (Centro Nacional de Ontologías Biomédicas) *BioPortal*, por parte de la universidad de Stanford a través del director del proyecto John Graybeal. La aplicación para el desarrollo de esta investigación, es una copia similar del sistema *NCBO BioPortal*, este sistema está en función desde el 29 de mayo del 2009. Por lo que el software ha sido probado anteriormente y se constata al no presentar anomalías al realizar las pruebas siguiendo el IEEE 829-2008 Standard for Software and System Test Documentation (IEEE Computer Society, 2008). Se presentaron dificultades al adaptar la aplicación para lo que se requería pero se consiguió establecer la representación de la estructura de la información para almacenar descripciones de patrones de análisis. No se descarta la posibilidad que en futuras investigaciones sufra variaciones la estructura de información en la aplicación.

El propósito de representar conocimiento a través de una estructura de información que almacene descripciones de patrones de análisis es brindar al usuario información acerca de un dominio de aplicación que desconozca o sobre el cual quiera incrementar su conocimiento. Actualmente no se cuenta con la información que se requiere con tres niveles definidos de ontologías, se confía la culminación de la idea a través de futuras investigaciones. El resultado

principal de esta investigación es la materialización, de la representación gráfica de la estructura de información propuesta, mediante una aplicación web.

Se espera que cuando la aplicación entre en función en la red de internet, el usuario experto en un dominio, aporte su conocimiento agregando descripciones de patrones de análisis, asimismo el experto en ontologías genere las ontologías de los PAs y los agregue al repositorio.

6.2 Detalles del diseño de la prueba

Esta sección describe las características a probar, los casos de pruebas, los criterios de aceptación y los entregables a producir en las pruebas.

6.2.1 Características a probar

El diseño de pruebas engloba las características referentes a los siete requerimientos funcionales: crear cuentas de usuario, representar la jerarquía de ontologías propuesta, realizar consultas al repositorio de ontologías, descargar archivos, aportar ontologías por usuarios registrados, actualizar información de cada ontología y eliminar ontología.

A continuación se muestra en la Tabla 6.2 la matriz de trazabilidad de pruebas, en la cual se describen las características a probar, se menciona el nombre del caso de prueba y del procedimiento de prueba, correspondientes a cada requerimiento funcional.

Tabla 6. 2 Matriz de trazabilidad.

Requerimiento	Funcionalidad	Caso de prueba	Procedimiento de prueba
1. Crear cuentas de usuario (cualquier usuario puede ejecutar esta acción)	<ol style="list-style-type: none"> 1. Solicitar el nombre de pila del usuario. 2. Solicitar el primer apellido. 3. Solicitar la dirección del correo electrónico. 4. Solicitar el identificador de usuario. 5. Solicitar contraseña. 	TC-01	TPr-01
2. Representar la jerarquía de ontologías propuesta (es la estructura que se propuso como propuesta de solución)	<ol style="list-style-type: none"> 1. Probar la existencia de la categoría para almacenar la ontología de la plantilla. 2. Probar la existencia de la categoría para almacenar los individuos de la plantilla. 3. Probar la existencia de la categoría para almacenar 	TC-02	TPr-02

Requerimiento	Funcionalidad	Caso de prueba	Procedimiento de prueba
Continuación del requerimiento 2.	<p>ontologías del patrón de análisis.</p> <p>4. Probar la existencia de la categoría para almacenar individuos de cada patrón de análisis.</p> <p>5. Probar la existencia de la categoría para almacenar las ontologías de casos específicos de los individuos de cada patrón de análisis.</p>		
3. Realizar consultas al repositorio de ontologías (cualquier usuario puede ejecutar esta acción)	Realizar consultas a través del nombre de una clase (palabra o concepto) de la ontología, para localizar una ontología en particular.	TC-03	TPr-03
4. Descargar archivos (cualquier usuario puede ejecutar esta acción)	Descargar el archivo de la ontología en formato OWL.	TC-04	TPr-04
5. Aportar ontologías por usuarios registrados (esta opción solo los usuarios registrados pueden realizarla)	<p>1. Requerir el nombre de la ontología.</p> <p>2. Requerir una breve descripción acerca de la ontología.</p> <p>3. Requerir el archivo de la ontología con formato OWL.</p> <p>4. Guardar los tres puntos anteriores en la aplicación.</p>	TC-05	TPr-05
6. Actualizar información de cada ontología (esta acción solo es realizada por el usuario que ingreso la ontología)	Actualizar la información de cada ontología si el autor respectivo lo desea.	TC-06	TPr-06
7. Eliminar ontología (solo el administrador de la aplicación puede realizar esta acción)	Eliminar ontologías.	TC-07	TPr-07

6.2.2 Caso de Prueba TC-01

A continuación se describen el identificador, el objetivo, la entrada y salida del caso de prueba TC-01.

6.2.2.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.3, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 3 Identificador del caso de prueba 01.

Requerimiento funcional	Identificador del caso de prueba
1. Crear cuentas de usuario	TC-01

6.2.2.2 Objetivo

Verificar el cumplimiento del requerimiento funcional: crear cuentas de usuario.

6.2.2.3 Entrada

Los datos de entrada son: el nombre de pila del usuario, su primer apellido, una dirección de correo electrónico válida, identificador del usuario y una contraseña, ver Figura 4.1.

New BioPortal Account

FIRST NAME:	<input type="text" value="Martha"/>
LAST NAME:	<input type="text" value="Zavaleta"/>
ACCOUNT NAME:*	<input type="text" value="martha2017"/>
EMAIL ADDRESS:*	<input type="text" value="lupiscz@gmail.com"/>
RE-ENTER EMAIL ADDRESS:*	<input type="text" value="lupiscz@gmail.com"/>
PASSWORD:*	<input type="password" value="....."/>
RE-ENTER PASSWORD:*	<input type="password" value="....."/>

Figura 6. 1 Datos de entrada para el primer requerimiento funcional crear cuentas de usuario.

6.2.2.4 Salida

La salida es la creación de la cuenta del usuario de manera satisfactoria. En la Figura 6.2 se observa el mensaje *Account was successfully created*, que demuestra que la cuenta de usuario se creó de manera satisfactoria.



Figura 6. 2 Cuenta de usuario con identificador martha2017.

6.2.3 Caso de Prueba TC-02

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-02.

6.2.3.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.4, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 4 Identificador del caso de prueba 02

Requerimiento funcional	Identificador del caso de prueba
2. Representar la jerarquía de ontologías propuesta	TC-02

6.2.3.2 Objetivo

Comprobar el cumplimiento del requerimiento: representar la jerarquía propuesta en la aplicación.

6.2.3.3 Entrada

La entrada es la estructura de información propuesta que se observa en la Figura 5.1 del capítulo 5. Propuesta de solución, En el nivel I, incluye el almacén con el archivo de la ontología de la plantilla (a), el almacén con los dos diferentes archivos de los individuos de la plantilla que contienen las descripciones de los patrones de análisis PARE y PAC (b) y el almacén de archivos *XMI* (c). El almacenado de archivos *XMI* se resolverá en investigaciones futuras por lo cual no se definió un caso de prueba para este almacén.

En el nivel II, se incluye el almacén de los archivos de las ontologías de los patrones de análisis PARE y PAC (d) y el almacén de los archivos de individuos de los patrones de análisis PARE (ejemplo Para la reserva de vuelos) y PAC (ejemplo Cuenta de usuario para la adquisición de nieves) (e).

En el nivel III, se incluye el almacén de archivos de ontologías de patrones de análisis de casos específicos (f). Por el momento no se cuenta con un archivo y por lo tanto no se definió un caso de prueba, para este almacén.

En el nivel II, el almacén de los archivos de individuos de los patrones de análisis PARE (ejemplo Para la Reserva de Vuelos) y PAC, contiene datos ficticio, porque actualmente no se cuenta con ontologías de individuos de estos patrones y se optó por crear individuos con datos ficticios, para mostrar la apariencia que tendrá el repositorio (d) en el nivel II.

6.2.3.4 Salida

Las cinco categorías creadas en la aplicación, siguiendo la estructura mostrada en la Figura 2.1 son: I-O Ontología de la plantilla (a), I-I Descripciones de PAs (b), II-O Ontologías de PAs(d), II-I Individuos de PAs(e), III-O Ontologías de Casos Específicos (f). La primera categoría mencionada contiene la ontología de la plantilla, la segunda categoría posee las ontologías de individuos de la plantilla, la tercera categoría tiene ontologías de patrones de análisis, la cuarta categoría tienen ontologías de los individuos de cada ontología del patrón de análisis (los datos que contienen estas ontologías son datos ficticios se crearon para realizar pruebas), la quinta categoría no contiene ontologías porque actualmente no se cuenta con una ontología para este nivel, se resolverá en futuras investigaciones. En la Figura 6.3 se muestra el nombre de las cinco categorías creadas.

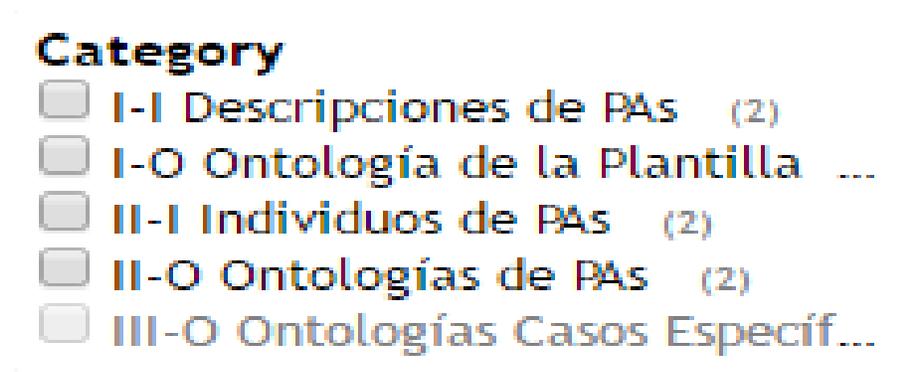


Figura 6. 3 Categorías definidas en la aplicación NCBO BioPortal.

A continuación se presenta un ejemplo de cómo se encuentra la categoría I-O Ontología de la Plantilla en la aplicación NCBO BioPortal. Esta categoría contiene la ontología de la plantilla para patrones de análisis estable y se encuentra en el nivel I dentro de la jerarquía de ontologías. En las otras categorías el proceso es similar por lo que cual sólo se presenta un caso para no parecer repetitivo. En la Figura 6.4 se muestra el contenido de la categoría I-O Ontología de la Plantilla.

Submit New Ontology

Entry Type

- Ontology (1)
- Ontology View (0)
- CIMI Model (0)
- NLM Value Set (0)

Uploaded in the Last

Category

- I-I Descripciones de PAs (2)
- I-O Ontología de la Plantilla ...
- II-I Individuos de PAs (3)
- II-O Ontologías de PAs (2)
- III-O Ontologías Casos Especif...

Ontología de la Plantilla Estable para Patrones de Análisis Estables (OPPAE)

La ontología de la plantilla estable se utiliza para describir patrones de análisis

Uploaded: 4/1/17

Figura 6. 4 Nivel I de la jerarquía de ontologías representado en la aplicación NCBO BioPortal.

En la Figura 6.5 se exhiben detalles de la plantilla, así como el estado en el que se encuentra la ontología, el tipo de formato, el nombre y correo electrónico de los creadores, el nombre de la categoría a la cual pertenece la ontología.

Ontología de la Plantilla Estable para Patrones de Análisis Estables
 Summary Classes Properties Notes Mappings Widgets

Details

ACRONYM	OPPAE
VISIBILITY	Public
DESCRIPTION	La ontología de la plantilla estable se utiliza para describir patrones de análisis.
STATUS	Beta
FORMAT	OWL
CONTACT	Rubí Celia Martínez Jiménez, ruby_martinez11@cenidet.edu.mx Moisés González García, moises@cenidet.edu.mx
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	I-O Ontología de la Plantilla
GROUPS	

Metrics ?

- NUMBER OF CLASSES:
- NUMBER OF INDIVIDUAL
- NUMBER OF PROPERTIES:
- MAXIMUM DEPTH:
- MAXIMUM NUMBER OF C
- AVERAGE NUMBER OF C
- CLASSES WITH A SINGL
- CLASSES WITH MORE TI
- CLASSES WITH NO DEFI

Visits

Figura 6. 5 Detalles acerca de la ontología de la plantilla para describir patrones de análisis.

En la Figura 6.6 se presenta parte del código de la ontología de la plantilla ubicada en el repositorio localizado en el servidor de la aplicación NCBO BioPortal.

```

GNU nano 2.0.9 File: OntologíaPlantilla1.owl
    <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainL$
</Annotation>
<Declaration>
  <Class IRI="#Plantilla"/>
</Declaration>
<Declaration>
  <Class IRI="#Contexto_Resultante"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#es_parte_de_7"/>
</Declaration>
<Declaration>
  <Class IRI="#Historial"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#11_ContextoResultante"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#es_parte_de_3"/>
</Declaration>
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
  
```

Figura 6. 6 Código de la ontología de la plantilla alojado en servidor de la aplicación NCBO BioPortal.

En la Figura 6.7 se presenta el archivo de la ontología de la plantilla en forma visual en el editor Protegé. El archivo de la ontología de la plantilla se descargó desde el portal BioPortal y se visualizó en el editor Protegé para probar que efectivamente en la aplicación NCBO BioPortal se encuentra almacenado la ontología de la plantilla.

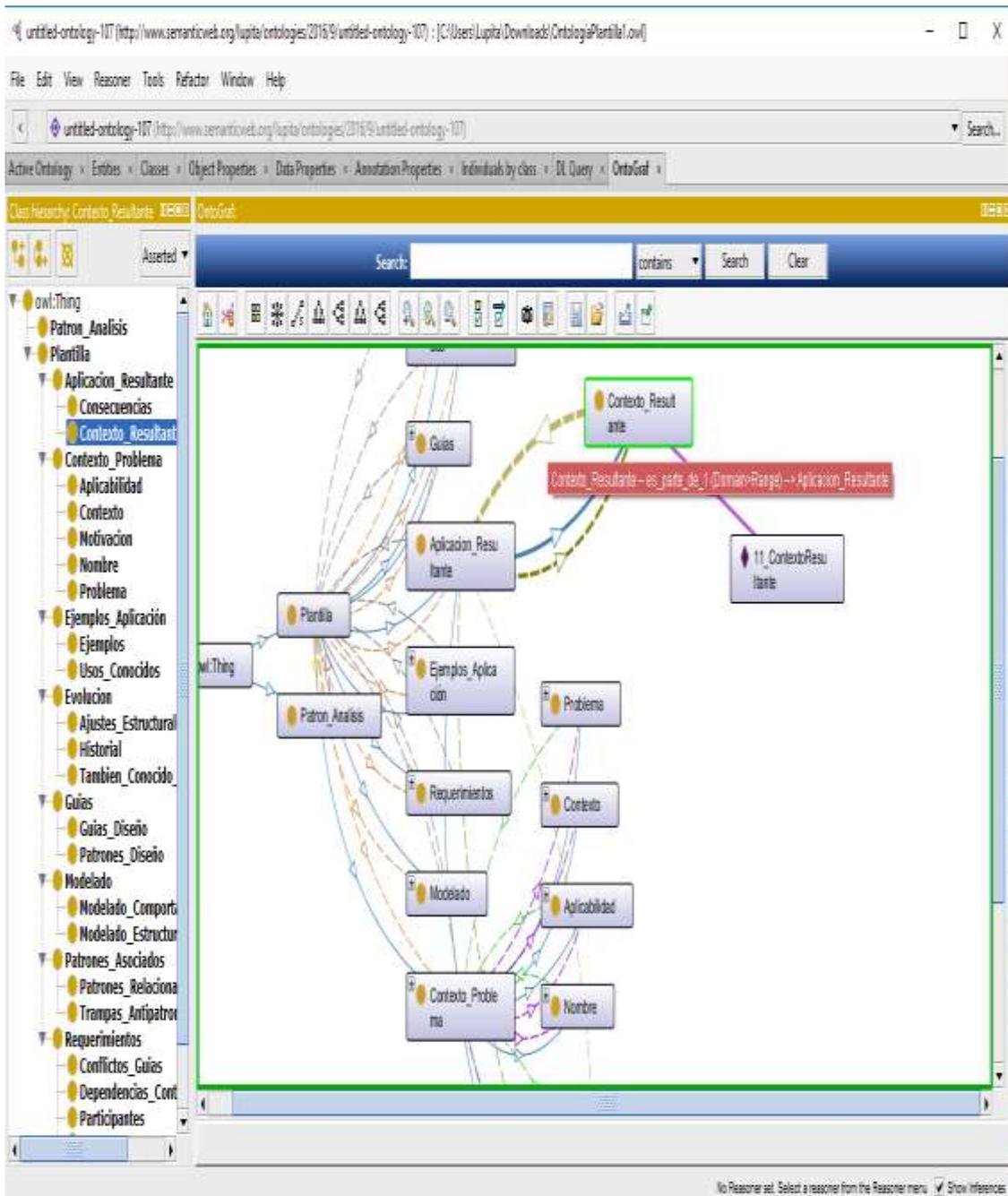


Figura 6. 7 Código de la ontología de la plantilla representado en editor Protegé.

6.2.4 Caso de Prueba TC-03

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-03.

6.2.4.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.5, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6.5 Identificador del caso de prueba 03.

Requerimiento funcional	Identificador del caso de prueba
3. Realizar consultas al repositorio de ontologías	TC-03

6.2.4.2 Objetivo

Comprobar el cumplimiento del requerimiento: buscar el nombre de una clase de una ontología en el repositorio.

6.2.4.3 Entrada

En el portal web BioPortal, dar clic sobre la pestaña Search, ubicada en el menú que se localiza en la parte superior del portal, para que salga la ventana de búsqueda. El dato de entrada es el nombre de una clase de una ontología, se optó por el nombre de la clase Historial, para el caso en el que el texto buscado se encuentra en el nombre de una clase, ver Figura 6.8. En el caso de falta de existencia del nombre de la clase se ingresó la palabra Prueba y el resultado se presenta en la Figura 6.9.

6.2.4.4 Salida

En la Figura 6.8, se observa el listado de las ontologías que contienen la clase con el nombre Historial.



Figura 6. 8 Consulta al repositorio de ontologías.

En la Figura 6.9, se muestra el cuadro de texto para la búsqueda de clases en las ontologías, en el cual, al ingresar el nombre de una clase que no pertenezca a una ontología, el resultado mostrado es la no existencia de coincidencias.

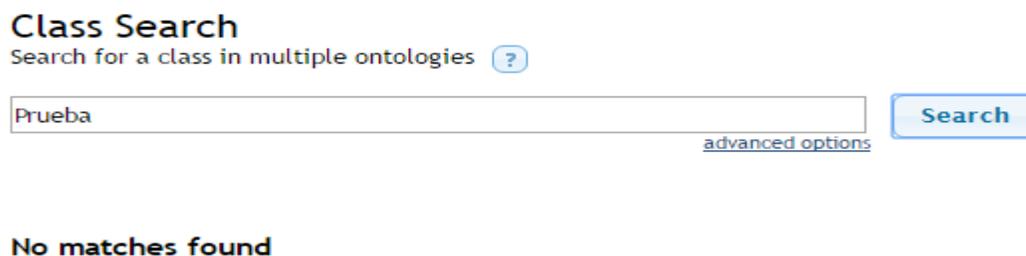


Figura 6. 9 No existen coincidencias con el nombre de la clase ingresado.

En el cuadro de búsqueda de clases, se cuenta con el comodín “*” para la búsqueda de una clase específica que no se recuerda exactamente cómo se escribe. En la Figura 6.10 se muestra un ejemplo del uso de este comodín. En este ejemplo se escribió la silaba ‘de’ seguida del comodín ‘*’, mostrando como resultado el nombre de la clase Dependencias_Contribuciones de la ontología de la plantilla estable para patrones de análisis.

Class Search

Search for a class in multiple ontologies ?

Search

[advanced options](#)

[Dependencias Contribuciones - Ontología de la Plantilla Estable por \(OPPAE\)](#)

http://www.semanticweb.org/lupita/ontologies/2016/9/untitled-ontology-107#Dependencias_Contribuciones - [visualize](#)

Figura 6. 10 Ingreso de caracteres especiales.

6.2.5 Caso de Prueba TC-04

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-04.

6.2.5.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.6, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 6 Identificador del caso de prueba 04.

Requerimiento funcional	Identificador del caso de prueba
4. Descargar archivos	TC-04

5.2.5.2 Objetivo

Comprobar el cumplimiento del requerimiento de descargar archivos.

5.2.5.3 Entrada

Dar clic sobre el nombre de la ontología que se desea descargar, lo que ocasiona que la aplicación despliegue una interfaz similar a la que se exhibe en la Figura 4.15 del capítulo 4, que incluya la ventana que se muestra en la Figura 6.11. A continuación se da clic en la liga OWL, situada en la opción de *DOWNLOADS*, de la Figura 6.11. Para este ejemplo se descarga el archivo de la ontología de la plantilla.

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
1.0 (Parsed, Indexed, Metrics, Annotator)	11/16/2016	11/16/2016	OWL CSV RDF/XML

Figura 6. 11 Enlace para la descarga del archivo OWL de la ontología de la plantilla.

5.2.5.4 Salida

En la Figura 6.12 se muestra el archivo descargado, en la Figura 6.13 se contempla parte del código del archivo de la ontología de la plantilla.

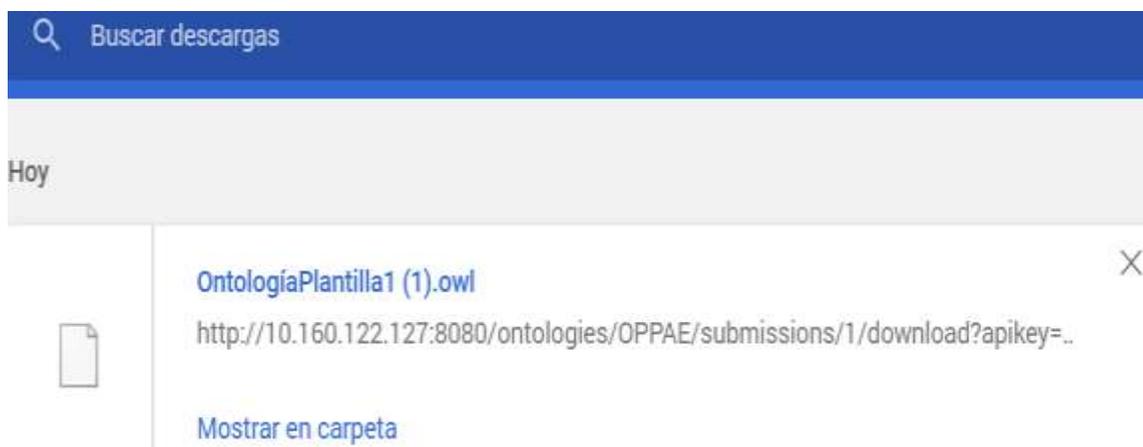


Figura 6. 12 Descarga del archivo de la ontología de la plantilla con formato owl.

```

18 </Annotation>
19 <Annotation>
20   <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
21   <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">Tes
22 </Annotation>
23 <Declaration>
24   <Class IRI="#Plantilla"/>
25 </Declaration>
26 <Declaration>
27   <Class IRI="#Contexto_Resultante"/>
28 </Declaration>
29 <Declaration>
30   <ObjectProperty IRI="#es parte de 7"/>
31 </Declaration>
32 <Declaration>
33   <Class IRI="#Historial"/>
34 </Declaration>
35 <Declaration>
36   <NamedIndividual IRI="#11_ContextoResultante"/>
37 </Declaration>
38 <Declaration>
39   <ObjectProperty IRI="#es parte de 3"/>

```

Figura 6. 13 Parte del código del archivo de la ontología de la plantilla.

6.2.6 Caso de Prueba TC-05

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-05.

6.2.6.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.7, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 7 Identificador del caso de prueba 05.

Requerimiento funcional	Identificador del caso de prueba
5. Aportar ontologías por usuarios registrados	TC-05

6.2.6.2 Objetivo

Comprobar el cumplimiento del requerimiento: aportar ontologías por usuarios registrados.

6.2.6.3 Entrada

Se elige para verificar la aportación de una ontología, crear un individuo del patrón de análisis para la Reservación de Entidades, que es un archivo con datos ficticios. El nombre que se le asigna a dicha ontología es Reserva de Vuelos. La aportación se da a través de la cuenta con identificador martha17, la cuenta fue creada en el caso de prueba TC-01. En la Figura 6.14 se muestra el estado de la categoría II-I Individuos de PAs antes realizar la aportación.



Figura 6. 14 Estado de la categoría II-I Individuos de PARE antes de realizar la aportación.

6.2.6.4 Salida

Una ontología más (Reserva de Vuelos (RV)) en la categoría II-I Individuos de PAs, como se puede observar en la Figura 6.15, aportación realizada por el usuario martha2017.



Figura 6. 15 Aportación de la ontología Reserva de Vuelos.

En la Figura 6.16 se presenta un ejemplo de error si se ingresa un archivo que no tenga la extensión *.owl*.

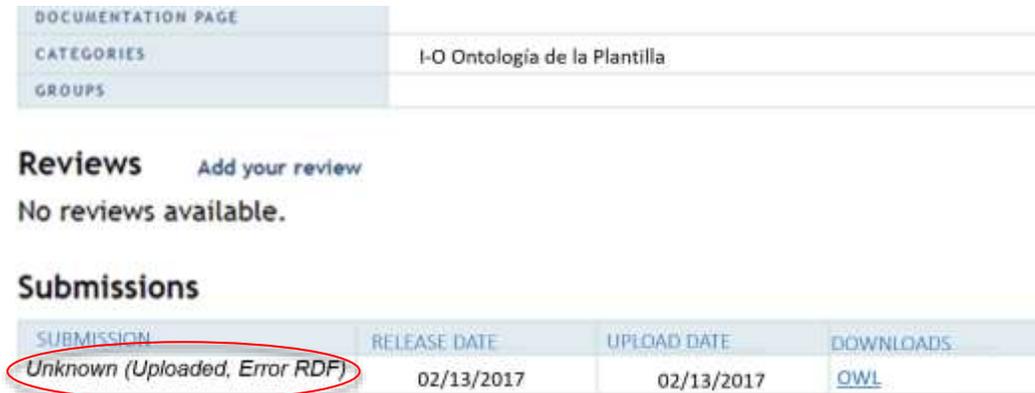


Figura 6. 16 Ejemplo de error al ingresar un archivo sin extensión *.owl*.

6.2.7 Caso de Prueba TC-06

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-06.

6.2.7.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.8, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 8 Identificador del caso de prueba 06.

Requerimiento funcional	Identificador del caso de prueba
6. Actualizar información de cada ontología	TC-06

6.2.7.2 Objetivo

Comprobar el cumplimiento del requerimiento: actualizar información de cada ontología.

6.2.7.3 Entrada

La actualización de la información de una ontología se permite solo para al autor que ingresó la ontología a la aplicación.

Existen tres formas de actualizar la información de la ontología. La primera forma es a través de la opción *Edit ontology information*, la segunda manera es con la opción *Edit Submission information* y la tercera es mediante la opción *Add submission*, de la pantalla mostrada en la Figura 6.17.

La primera forma es a través de la opción *Edit ontology information* y la pantalla utilizada se muestra en la Figura 4.18. La segunda forma es editar la información presentada para realizar esto dar clic en la leyenda *Edit Submission information*, y la pantalla utilizada se muestra en la Figura 6.19. La tercera forma es agregar una nueva presentación de la información de la ontología esto se consigue dando clic en la leyenda *Add submission*, y la pantalla utilizada se muestra en la Figura 6.20.

Para esta prueba se utiliza como dato de entrada la información del individuo Reserva de Vuelos, el cual contiene datos ficticios.

Reserva de Vuelos	
Summary Classes Properties Notes Mappings Widgets	
Details	
ACRONYM	RV
VISIBILITY	Public
DESCRIPTION	Describe el proceso para la reservación de boletos de avión (Los datos de la ontología son ficticios, el archivo se usa para probar la realización de una aportación, y para mostrar la presentación de la categoría II-I individuos de PAs)
STATUS	Alpha
FORMAT	OWL
CONTACT	Guadalupe Candia Zavaleta, gcandia@cenidet.edu.mx Moisés González García, moises@cenidet.edu.mx
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	II-I Individuos de PAs
GROUPS	

Figura 6. 17 Detalle de la ontología con datos ficticios Reserva de Vuelos.

BioPortal Browse Search Mappings Recommender Annotator Projects

Edit Ontology Information * fields are required

NAME: *	<input type="text" value="Reserva de Vuelos"/>
ACRONYM: *	<input type="text" value="RV"/>
ADMINISTRATORS: *	<input type="text" value="admin x"/>
VIEWING RESTRICTIONS	<input type="text" value="Public"/>
CATEGORIES:	<input type="text" value="II-I Individuos de PAs x"/>
VIEW:	<input checked="" type="checkbox"/> This ontology is a view of: <input type="text" value="Select an ontology to create a view on"/>
<input type="checkbox"/> Subscribe to email notifications for new notes	
<input type="button" value="Cancel"/> <input type="button" value="Save ontology"/>	

Figura 6. 18 Primera forma: Editar Información de la Ontología.

Edit Submission Information

* fields are required

DESCRIPTION *	Describe el proceso para la reservación de boletos de avión (Los datos de la ontología son ficticios, el archivo se usa para probar la realización de una aportación,													
FORMAT *	OWL ▼	<table border="1"> <tr> <td style="background-color: #e0e0e0;">PREFERRED NAME PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#prefLabel</small></td> <td>use default</td> </tr> <tr> <td style="background-color: #e0e0e0;">SYNONYM PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#altLabel</small></td> <td>use default</td> </tr> <tr> <td style="background-color: #e0e0e0;">DEFINITION PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#definition</small></td> <td>use default</td> </tr> <tr> <td style="background-color: #e0e0e0;">AUTHOR PROPERTY <small>default: http://purl.org/dc/elements/1.1/creator</small></td> <td>use default</td> </tr> </table> <p>Determining Obsolete Classes You can use the following two methods to identify obsolete classes in your ontology:</p> <ol style="list-style-type: none"> 1. Designate a root ID for an obsolete branch. 2. Identify a property id that indicates obsolete status. The value must be set to "true". <table border="1"> <tr> <td style="background-color: #e0e0e0;">1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)</td> <td><input type="text"/></td> </tr> <tr> <td style="background-color: #e0e0e0;">2. PROPERTY TO CHECK FOR OBSOLETE STATUS</td> <td><input type="text"/></td> </tr> </table>	PREFERRED NAME PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#prefLabel</small>	use default	SYNONYM PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#altLabel</small>	use default	DEFINITION PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#definition</small>	use default	AUTHOR PROPERTY <small>default: http://purl.org/dc/elements/1.1/creator</small>	use default	1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)	<input type="text"/>	2. PROPERTY TO CHECK FOR OBSOLETE STATUS	<input type="text"/>
PREFERRED NAME PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#prefLabel</small>	use default													
SYNONYM PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#altLabel</small>	use default													
DEFINITION PROPERTY <small>default: http://www.w3.org/2004/02/skos/core#definition</small>	use default													
AUTHOR PROPERTY <small>default: http://purl.org/dc/elements/1.1/creator</small>	use default													
1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)	<input type="text"/>													
2. PROPERTY TO CHECK FOR OBSOLETE STATUS	<input type="text"/>													
VERSION	<input type="text" value="1.0"/>													
STATUS *	alpha ▼													
RELEASE DATE *	<input type="text" value="2017-01-20T00:00:00+00:00"/> <input type="button" value="📅"/>													
	<input type="radio"/> Metadata Only (choose this option if you want users to search and view only the metadata for your ontology, but not its classes and properties)													

Figura 6. 19 Segunda forma: Editar la información presentada.

BioPortal Browse Search Mappings Recommender Annotator Projects

Add New Submission * fields are required

DESCRIPTION * Describe el proceso para la reservación de boletos de avión (Los datos de la ontología son ficticios, el archivo se usa para probar la realización de una aportación,

FORMAT * OWL

PREFERRED NAME PROPERTY default: <http://www.w3.org/2004/02/skos/core#prefLabel> use default

SYNONYM PROPERTY default: <http://www.w3.org/2004/02/skos/core#altLabel> use default

DEFINITION PROPERTY default: <http://www.w3.org/2004/02/skos/core#definition> use default

AUTHOR PROPERTY default: <http://purl.org/dc/elements/1.1/creator> use default

Determining Obsolete Classes You can use the following two methods to identify obsolete classes in your ontology:

1. Designate a root ID for an obsolete branch.
2. Identify a property id that indicates obsolete status. The value must be set to "true".

1. ROOT OF OBSOLETE BRANCH (ROOT IS NOT OBSOLETE)

2. PROPERTY TO CHECK FOR OBSOLETE STATUS

VERSION 1.0

STATUS * alpha

RELEASE DATE * 2017-01-20T00:00:00+00:00

Metadata Only (choose this option if you want users to search and view only the metadata for your ontology, but not its classes and properties)

Figura 6. 20 Tercera forma: Agregar una nueva presentación de la información de la ontología.

En la Figura 6.21 se presenta el estado de la interfaz antes de actualizar la ontología a través de la forma dos y tres, en la manera uno no se actualiza el archivo de la ontología.

Submissions

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
1.0 (Parsed, Indexed, Metrics, Annotator)	11/15/2016	11/15/2016	OWL CSV RDF/XML

Figura 6. 21 Estado del archivo de la ontología antes de ser actualizado.

6.2.7.4 Salida

En la Figura 6.22 se aprecia el resultado después de efectuar la prueba con la opción *Editar ontology Information*, en este caso se actualizo el nombre de la ontología de Reserva de Vuelos a Reserva de Vuelos para American.

Reserva de Vuelos para American	
Summary Classes Properties Notes Mappings Widgets Edit ontology information Add submission Edit submission information	
Details	
ACRONYM	RV
VISIBILITY	Public
DESCRIPTION	Describe el proceso para la reservación de boletos de avión (Los datos de la ontología son ficticios, el archivo se usa para probar la realización de una aportación, y para mostrar la presentación de la categoría II-I individuos de PARE)
STATUS	Alpha
FORMAT	OWL
CONTACT	Guadalupe Candia Zavaleta, gcandia@cenidet.edu.mx Moisés González García, moises@cenidet.edu.mx
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	II-I Individuos de PARE
GROUPS	

Figura 6. 22 Salida de la forma uno para actualizar ontologías.

En la Figura 6.23 se presenta la salida de actualizar ontologías para la forma dos y tres, se puede observar que se cambió el texto que tenía originalmente en la sección de *DESCRIPTION* por el texto Probando actualización, recuerde que solo es un ejemplo para probar que la aplicación permite actualizar la información de la ontología. Se presenta un ejemplo para la forma de actualizar ontologías dos y tres porque la salida para las dos formas es de la misma manera.

Reserva de Vuelos

- [Summary](#)
- [Classes](#)
- [Properties](#)
- [Notes](#)
- [Mappings](#)
- [Widgets](#)
- [Edit ontology information](#)
- [Add submission](#)
- [Edit submission info](#)

Details

ACRONYM	RV
VISIBILITY	Public
DESCRIPTION	Probando actualización
STATUS	Alpha
FORMAT	OWL
CONTACT	Guadalupe Candia Zavaleta, gcandia@cenidet.edu.mx Moisés González García, moises@cenidet.edu.mx
HOME PAGE	
PUBLICATIONS PAGE	
DOCUMENTATION PAGE	
CATEGORIES	II-I Individuos de PARE
GROUPS	

Submissions

SUBMISSION	RELEASE DATE	UPLOAD DATE	DOWNLOADS
1.0 (Uploaded)	02/14/2017	02/14/2017	OWL
1.0 (Parsed, Indexed, Metrics, Annotator)	02/14/2017	02/14/2017	OWL CSV RDF/XML Diff
1.0 (Parsed, Indexed, Metrics, Annotator)	01/20/2017	02/14/2017	OWL Diff
1.0 (Archived)	01/20/2017	02/14/2017	OWL

Figura 6. 23 Salida de la forma dos y tres para actualizar ontologías.

6.2.8 Caso de Prueba TC-07

En las siguientes subsecciones se menciona el identificador, el objetivo, la entrada y salida del caso de prueba TC-07.

6.2.18.1 Identificador del Caso de Prueba

Se enlista en la Tabla 6.9, el requerimiento funcional a probar y su correspondiente identificador de caso de prueba.

Tabla 6. 9 Identificador del caso de prueba 07

Requerimiento funcional	Identificador del caso de prueba
7. Eliminar ontología	TC-07

6.2.8.2 Objetivo

Comprobar el cumplimiento del requerimiento: eliminar ontología.

6.2.8.3 Entrada

La ontología a eliminar tiene por nombre Reserva de Vuelos, por lo tanto la entrada es el acrónimo de la ontología a suprimir, el cual es RV. En la Figura 6.24 se presenta el estado de la categoría II-I Individuos de PAs antes de ser eliminada la ontología ficticia que tiene el nombre de Reserva de Vuelos. Se recuerda que esta opción está habilitada solo para el administrador de la aplicación.



Figura 6. 24 Estado de la categoría II-I Individuos de PARE antes de eliminar la ontología Reserva de Vuelos.

6.2.8.4 Salida

En la Figura 6.25 se puede examinar que la ontología con el nombre de Reserva de Vuelos ha sido eliminada del repositorio.

The screenshot shows a web interface for an ontology repository. At the top, there is a search bar and a status indicator 'Showing 2 of 8. Sort: Popular'. Below this, there are two main entries:

- Cuenta para la Adquisición de Nieves (CAN)**: This entry has a description: 'Describe el proceso para adquirir nieves (esta descripción es ficticia, se usa para probar el llenado de la ontología PAC y mostrar la presentación de la categoría II-I Individuos de PAC)'. It has 7 classes and was uploaded on 4/1/17.
- Reserva de Libros (RL)**: This entry has a description: 'Se hace uso de la ontología del Patrón de Análisis para Reservación de Entidades, para describir el proceso de préstamo de libros dentro de la biblioteca de CENIDET'. It has 8 classes and was uploaded on 4/1/17.

On the left side, there are several filters and buttons:

- A 'Submit New Ontology' button.
- An 'Entry Type' filter with options: 'Ontology (2)', 'Ontology View (0)', 'CIMI Model (0)', and 'NLIA Value Set (0)'. 'Ontology' is selected.
- An 'Uploaded in the Last' filter with a dropdown menu.
- A 'Category' filter with options: 'I-I Descripciones de PAs (2)', 'I-O Ontología de la Plantilla ...', 'II-I Individuos de PAs (2)', 'II-O Ontologías de PAs (2)', and 'III-O Ontologías Casos Especif...'. 'II-I Individuos de PAs' is selected.

Figura 6. 25 Situación de la aplicación después de la eliminación de la ontología Reserva de Vuelos.

En esta investigación, para almacenar la semántica de las descripciones de patrones de análisis se propuso seguir una estructura de información que contempla tres niveles de abstracción, expuesta en la Figura 5.1 del capítulo 5.

La estructura de información que se propuso se implementó en una copia del software NCBO BioPortal proporcionado por la universidad de *Stanford*. Consiguiendo almacenar las descripciones de patrones de análisis y la semántica de los patrones de análisis conforme a la estructura de información propuesta. Los resultados obtenidos de las pruebas realizadas a la aplicación antes mencionada, siguiendo los requerimientos funcionales que se describen en la sección 4.2.2 indican que es factible almacenar las ontologías de las descripciones de patrones de análisis, además la aplicación NCBO BioPortal brinda flexibilidad y está orientado a la interacción con el usuario. En el capítulo siguiente se plasman las conclusiones a las que se llegó después de realizar esta investigación y se mencionan investigaciones futuras.

Capítulo 7. Conclusiones

En este capítulo se presentan las conclusiones a las que se llega y los trabajos futuros que se estima podrán partir de los resultados obtenidos en esta investigación.

5.1 Conclusiones

Con base en los resultados presentados en el capítulo 5 se concluye el cumplimiento del objetivo general, que se menciona en la sección 1.2 del capítulo 1:

“Almacenar descripciones de patrones de análisis que incluya su semántica, mediante una jerarquía de ontologías y sus individuos.”

Y se muestra que es factible almacenar descripciones de patrones de análisis porque fue posible crear e implementar la estructura de información presentada en este documento.

En base al análisis de las ontologías de la plantilla y los PAs se determinó la conveniencia de separarlo en varios archivos y se diseñó la estructura de información para facilitar su consulta. Los archivos son:

- 1) La ontología de la plantilla estable para describir patrones de análisis
- 2) La descripción del patrón de análisis para reserva de entidades
- 3) La descripción del patrón de análisis de cuenta.

La estructura de información orienta al usuario a conocer el nivel de abstracción en el que se localiza cierta ontología:

- Los individuos de la ontología para la Plantilla de Patrones de Análisis Estable son las descripciones de los patrones de análisis, acorde a la estructura de la plantilla. Por ejemplo: La descripción del Patrón de Análisis para la Reutilización de Entidades (PARE), o la descripción del Patrón de Análisis de Cuenta (PAC).
- Las ontologías de los PAs, obtenidas a partir de sus descripciones, representa la semántica de los conceptos (clases) y las relaciones entre ellos, así como sus propiedades y restricciones. Por ejemplo: la ontología de PARE representa a sus clases (Ej. Reservación, Persona, Cliente, Empresa, Entidad, etc.), sus relaciones (Ej. Cliente reserva Entidad, Empleado crea Reservación, etc.), sus propiedades (Ej. Entidad tiene los atributos nombre, identidad, disponibilidad, etc.), etc.
- Los individuos de la ontología de un patrón de análisis, son la descripción del patrón de análisis conforme a la estructura de sus clases, relaciones, propiedades y restricciones, para un caso específico. Por ejemplo, el PARE con los nombres de clase, relaciones y propiedades, así como los valores de las propiedades, para un caso específico (el préstamo de libros de la biblioteca de la UAEM).

La estructura de información que se propuso se implementó en una copia del software NCBO BioPortal proporcionado por la Universidad de Stanford. Consiguiendo almacenar las descripciones de patrones de análisis y la semántica de los patrones de análisis en la estructura

propuesta. Esto demuestra su factibilidad. El software NCBO BioPortal brinda flexibilidad y está orientado a la interacción con el usuario.

Contar con un repositorio de ontologías de patrones de análisis disponible en la web ayudará a compartir información entre las personas expertas en los procesos de un dominio y las personas menos expertas al mismo. Además permitirá la proliferación de patrones de análisis aumentando la diversidad de dominios. Ayudará a personas inexpertas en los procesos de un dominio a representar la problemática que se desea resolver. Siempre y cuando el repositorio cuente con un patrón de análisis correspondiente a ese problema y que se pueda aplicar en el dominio de interés.

5.2 Trabajos Futuros

Esta investigación es la base para el funcionamiento de un software que cuente con un repositorio de patrones de análisis que asiste al desarrollador o analista inexperto en el proceso de describir la problemática del negocio en un dominio determinado. Para continuar con la investigación se recomienda:

1. Terminar el nivel dos (e) de la estructura de información construyendo un individuo con datos reales de la ontología del patrón de análisis para entidades reutilizables o de la ontología del patrón de análisis de cuentas.
2. Después de la obtención del individuo de un patrón de análisis en el nivel dos (e), construir la ontología de un patrón de análisis de un caso específico para el nivel tres (f), con las diferencias respecto a la ontología del PA (d).
3. Aplicar ingeniería inversa a la aplicación NCBO BioPortal con el objetivo de entender el modo en que funciona y poder agregar funcionalidades que estén orientadas más a patrones de análisis, como:
 - Completar el nivel uno (c) de la jerarquía de ontología añadiendo los diagramas *GRL*, *UCM* y *UML* de los individuos de la platilla en la aplicación. Permitiendo visualizar los diagramas en la aplicación y brindando la opción de descargar los diagramas con extensión *.xmi*.
 - Agregar la funcionalidad de realizar consultas de un texto, en la información que contiene cada una de las propiedades de datos de cada individuo, dentro de una ontología.

Referencias

- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language*. California: Center for Environmental Structure de Berkeley.
- Amyot, D. (2003). Introduction to the User Requirements Notation: Learning by Example. *Computer Networks*, 285-301.
- Bobkawska, A., & Grabowski, J. (2009). *The Role of Analysis Patterns in Systems Analysis*. Germany: In EuroPLop 2009: 14th Annual European Conference on Pattern Languages of Programming.
- Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. Enschede, Países Bajos: Tesis doctoral de la facultad de Electrical Engineering, Mathematics and Computer Science (EEMCS) en la Universidad de Twente.
- Chakroun, C., Bellatreche, L., Ait-Ameur, Y., Berkani, N., & Jean, S. (2013). Be Careful When Designing Semantic Databases: Data and Concepts Redundancy. *In Seventh International Conference on Research Challenges in Information Science (RCIS)*, 1-12.
- da Matta Vegi, L. F., Alves Peixoto, D., Santos Soares, L., Lisboa-Filho, J., & de Paiva Oliveira, A. (2012). An Infrastructure Oriented for Cataloging Services and Reuse of Analysis Patterns. *Springer-Verlag Berlin Heidelberg, Part II, LNBIP 100*, 338-343.
- de Freitas Sondré, V., Lisboa F., J., Vilela, V. M., & Alvim Andrade, M. V. (2005). Improving Productivity and Quality of GIS Databases Desing using an Analysis Pattern Catalog. *Proceeding of the 2nd Asia-Pacific conference on Conceptual modelling -Volumen 43*, 107-114.
- Elamin, E., & Feki, J. (2014). Toward An Ontology Based Approach For Data Warehousing. *The international arab conference on information on information technology (acit2014)*, 172-179.
- Fayad, M., & Altman, A. (2001). An Introduction to Software Stability. *Communication of the ACM 44(9)*, 95-98.
- Fernandez, E. (1998). Building systems using analysis patterns. *Proceedings of the third international workshop on Software architecture*, 37-40.

- Fowler, M. (1997). *Analysis Patterns: Reusable Object Models* . Addison-Wesley Professional.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.
- González García, M., Martínez Jiménez, R. C., Villamil Brito, J. M., & López Ramírez, R. M. (2015). Stable Template for Stable Analysis Patterns. *IEEE LATIN AMERICA TRANSACTIONS*, 2430-2440.
- Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications . *Journal Knowledge Acquisition - Special issue: Current issues in Knowledge modeling*, 199-220.
- Hamza, H., & Fayad, M. (2002). Model-based Software Reuse Using Stable Analysis Patterns. *In ECOOP02 - Workshop on Model-based Software Reuse*.
- Henttonen, K., Matinlassi, M., Niemela, E., & Kanstrén, T. (2007). Integrability and Extensibility Evaluation from Software Architectural Models- A Case Study. *The Open Software Engineering Journal*, 1-20.
- IEEE. (1988). IEEE Recommended Practice for Software Requirements Specifications. *IEEE Xplore*.
- IEEE Computer Society. (2008). IEEE Standard for Software and System Test Documentation. *IEEE Xplore*, 829-2008.
- ITU-T, Z. (14 de Octubre de 2012). User requirements notation (URN). *LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS*. Ottawa, Canada.
- Jean, S., Bellatreche, L., Fokou, G., Baron, M., & Khouri, S. (2012). OntoDBench: Sistema de evaluación comparativa para bases de datos basadas en ontologías. *Springer-Velag Berlin Heidelberg, R. Meersman et al. (Eds): OTM, Par II, LNCS 7566, 897-914*.
- Júarez, J. D. (Julio de 2016). *Metodología de Validación de Consistencia de Patrones de Análisis*. Cuernavaca, Morelos, México, Morelos, México: Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- L. Whetzel, P., Noy, N., Shah, N., Alexander, P., Nyulas, C., Tudorache, T., & Musen, M. (2011). BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in. *Nucleic Acids Research*.

- López, R. M. (Mayo de 2015). *Almacenamiento y uso de patrones de análisis apegados a su semántica*. Cuernavaca, Morelos, México, Morelos, México: Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- Martínez, R. C. (2013). *Documentación de la Ontología para la Plantilla de Patrones de Análisis Estable (OPPAE)*. Cuernava, Morelos, México: Reporte de resultados de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico.
- McGuinness, D., & Harmelen, F. (10 de Febrero de 2004). Obtenido de W3C: <https://www.w3.org/TR/owl-features/>
- Musen, NF, N., NH, S., PL, W., CG, C., MA, S., . . . team, N. (2011). The National Center for Biomedical Ontology. *J Am Med Inform Assoc*.
- Pantoquilha, M., Raminhos, R., & Araujo, J. (2003). Analysis Patterns Specifications: Filling the Gaps. *II VIKING PLOP, Bergen, Norway*.
- Raminhos, R., Pantoquilha, M., Araújo, J., & Moreira, A. (2006). A Systematic Analysis Patterns Specification. *Conference: ICEIS 2006-Proceedings of the Eighth International Conference on Enterprise* .
- Rosanigo, Z. B. (Febrero de 2000). Maximizando reuso en software para Ingeniería Estructural, Modelos y Patrones. Argentina: Universidad nacional de la plata, tesis para obtener el grado de maestría.
- Salvadores, M., Alexander, P. R., Musen, M. A., & Noy, N. F. (2013). BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF. *Semantic web*, 277-284.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. Aravaca (Madrid): MC Graw Hil.
- Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering* 25, 161-197.
- Vaccare Braga, R., Germano, F., & Masiero, P. C. (1998). A Confederation of Patterns for Resource Management. 1-13.
- Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, & Musen MA. (14 de Junio de 2011). *BioPortal*. Obtenido de BioPortal: <http://bioportal.bioontology.org/>
- Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, & Musen MA. (14 de Junio de 2011). *BioPortal: enhanced functionality via new Web services from the*

Referencias

National Center for Biomedical Ontology to access and use ontologies in software applications. Obtenido de <https://bioportal.bioontology.org/>

Zhou, Z., & Yongkang, X. (2013). A study on ontoloy storage based on relational database. *IEEE*.

Anexo A. Descripción técnica de BioPortal

1. Introducción.

En el presente documento se describe el funcionamiento de la aplicación NCBO BioPortal, la cual permite entrar a una biblioteca de ontologías biomédicas. Accediendo a la aplicación través de un portal *web*. BioPortal es un repositorio abierto de ontologías biomédicas que almacena las ontologías desarrolladas en diversos formatos y es el más completo en el mundo de ontologías biomédicas.

Los usuarios de BioPortal pueden explorar ,buscar ontologías, presentar nuevas versiones de ontologías en el repositorio, comentar cualquier ontología o parte de alguna ontología en el repositorio, dar su opinión acerca de la ontología, describir su experiencia en el uso de la ontología o hacer sugerencias a los desarrolladores de ontologías. El enfoque es permitir que los miembros de la comunidad contribuyan al contenido de BioPortal activamente y aumentar el valor de ese contenido a los usuarios es lo que distingue a BioPortal de otros repositorios.

2. Aplicación NCBO BioPortal

La aplicación virtual *NCBO* (Centro Nacional de Ontologías Biomédicas o *National Center for Biomedical Ontology*) es una copia del software *NCBO* (Musen, y otros, 2011) (NCBO team, 2011) que se ejecuta en su propio sistema Linux. Tiene la facultad para almacenar ontologías biomédicas contribuidas por usuarios del software.

La forma que el usuario accede a la aplicación *NCBO* es a través de un portal web con el nombre de BioPortal (Whetzel PL, y otros, BioPortal, 2011)(Whetzel PL y otros, 2011) que proporciona acceso a una biblioteca de ontologías y terminologías biomédicas desarrolladas en Lenguaje de Ontologías para la *Web* (*OWL*), el Marco de Descripción de Recursos (*RDF*), formato abierto de ontologías biológicas y biomédicas (*OBO*), marcos de *Protégé* y Formato de Liberación Enriquecida (<http://bioportal.bioontology.org>).

BioPortal tiene una arquitectura orientada al servicio; los servicios *Web NCBO* proporcionan la funcionalidad encontrada en BioPortal y estos servicios *Web* pueden ser incorporados en otras aplicaciones de software para acceder y usar el contenido de la ontología. Los usuarios registrados pueden agregar asignaciones entre términos, agregar comentarios sobre términos individuales dentro de la ontología y proporcionar una revisión de ontologías. Este contenido generado por el usuario proporciona un mecanismo de evaluación crítica y de retroalimentación para los desarrolladores de ontologías. El enfoque específico en permitir la retroalimentación de la comunidad al contenido de BioPortal es una característica distintiva del sistema (L. Whetzel, y otros, 2011).

3. Interfaz de usuario BioPortal

3.1 Página principal

BioPortal proporciona acceso a ontologías biomédicas de uso común y a herramientas para trabajar con ellas. BioPortal permite navegar por la biblioteca de ontologías, buscar un término a través de múltiples ontologías, examinar *mappings* entre términos de diferentes ontologías, recibir recomendaciones sobre qué ontologías son más relevantes para usarlas en un corpus, registrar texto con términos de ontologías, buscar recursos biomédicos para un término, navegar por una selección de proyectos que utilizan los recursos de BioPortal, contemple la Figura A.1.

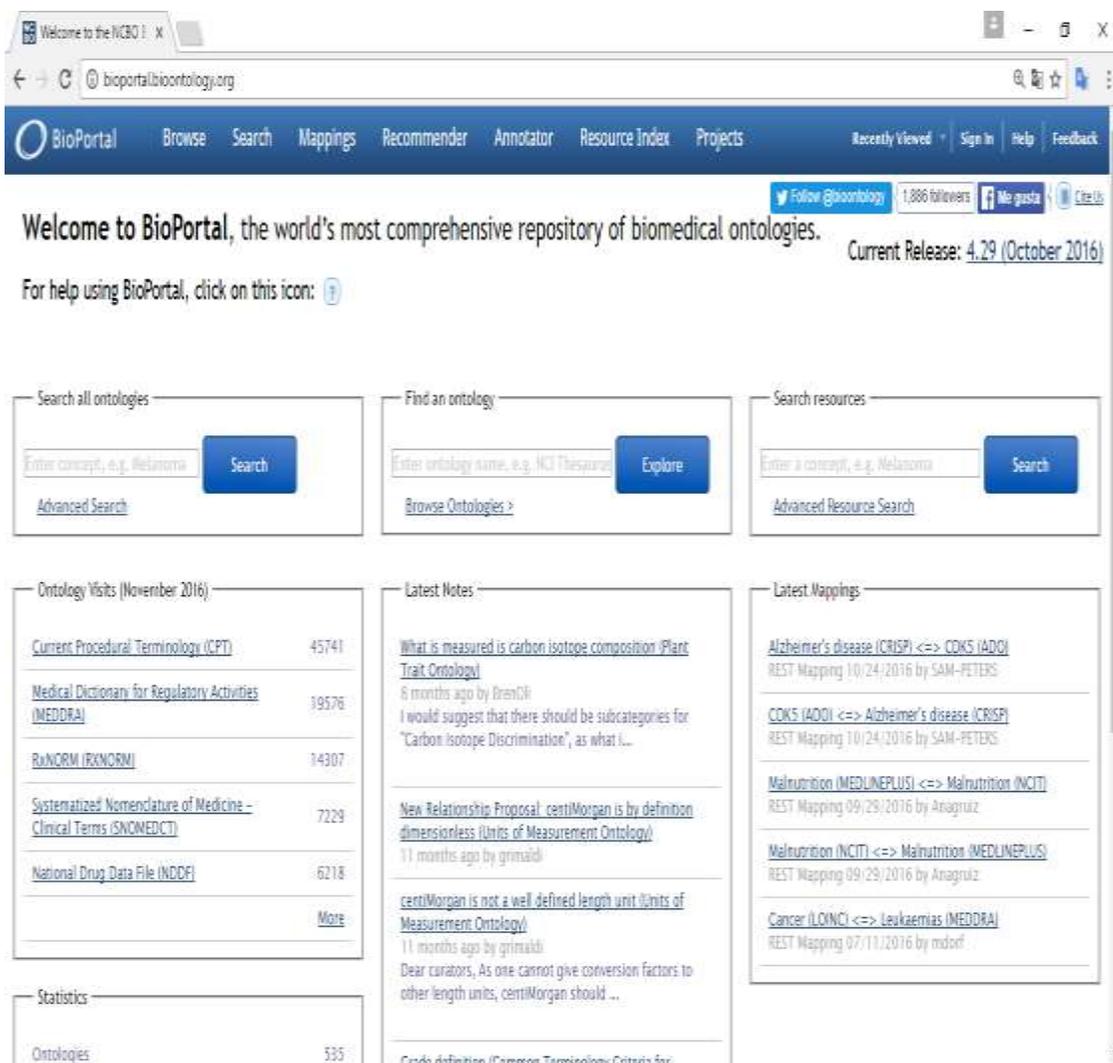


Figura A. 1 Pestaña de inicio de BioPortal.

3.2 Pestaña Browse

Permite examinar la biblioteca de ontologías biomédicas y a los usuarios registrados enviar su propia ontología. Esta página muestra información resumida sobre cada ontología, ver Figura A.2. Más detalles sobre la información resumida están disponible en la página de cada ontología. Se puede acceder a la página haciendo clic en el enlace del nombre de la ontología.

Para encontrar una ontología particular de forma rápida, escriba el nombre o abreviatura de la ontología en el campo “filtro de texto”. También puede reducir la lista de las ontologías que se muestran mediante la selección de una “categoría” específica (un dominio) o un “grupo” al que pertenece la ontología. Para realizar aportaciones de ontologías pulsar el botón *Submit New Ontology*.

Dependiendo de los remitentes las ontologías están marcadas con visibilidad privada o licencia. Para ontologías privadas el presentador de la ontología controla directamente el acceso a la ontología. Por lo tanto para tener acceso a una ontología privada hay que ponerse en contacto con el remitente. En cuanto a las ontologías con licencia, el remitente controla indirectamente el acceso a la ontología. Cuando se intenta acceder a una ontología con licencia, el sistema le pedirá que introduzca la información de la licencia requerida por el remitente. Cuando haya introducido la información requerida, el sistema le permitirá acceder a la ontología. Los términos de las ontologías no están disponibles en BioPortal en absoluto, solo se presenta un resumen de cada ontología, para obtener más información acerca de las ontologías póngase en contacto con el remitente.

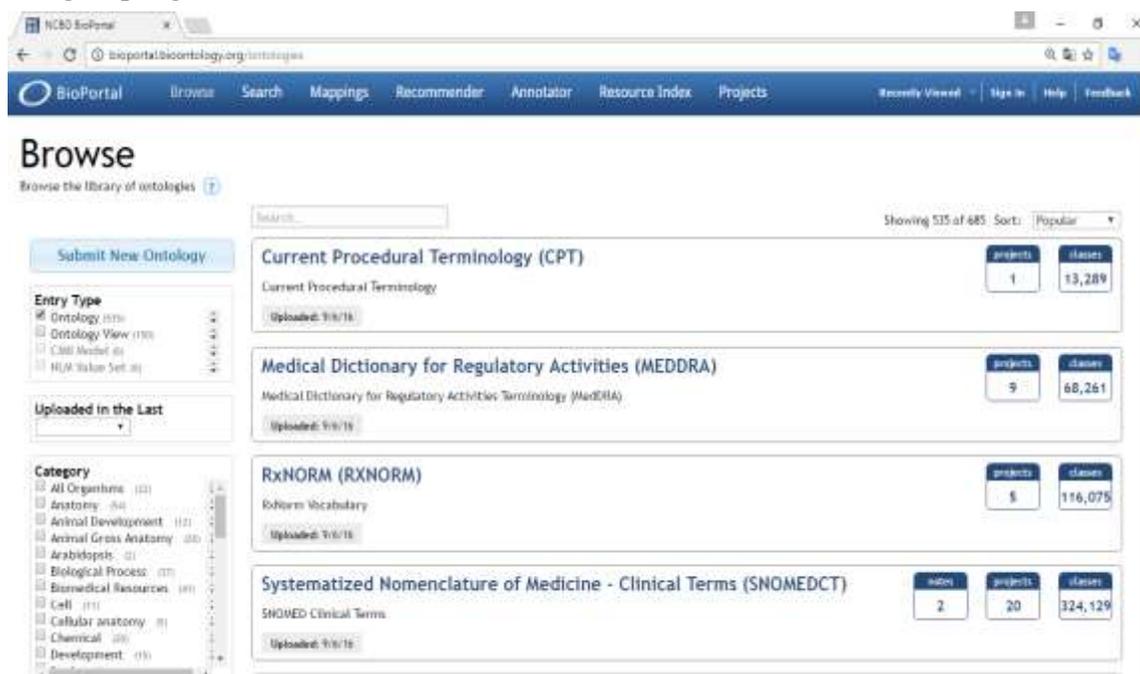


Figura A. 2 Pestaña Browser de BioPortal.

3.3 Pestaña Search

Concede buscar términos en las ontologías que contengan estos términos. Escribiendo en el cuadro de búsqueda el nombre del término a buscar, como se muestra en la Figura A.3. El sistema busca coincidencias en los nombres de los términos, identificadores y el valor de las propiedades.

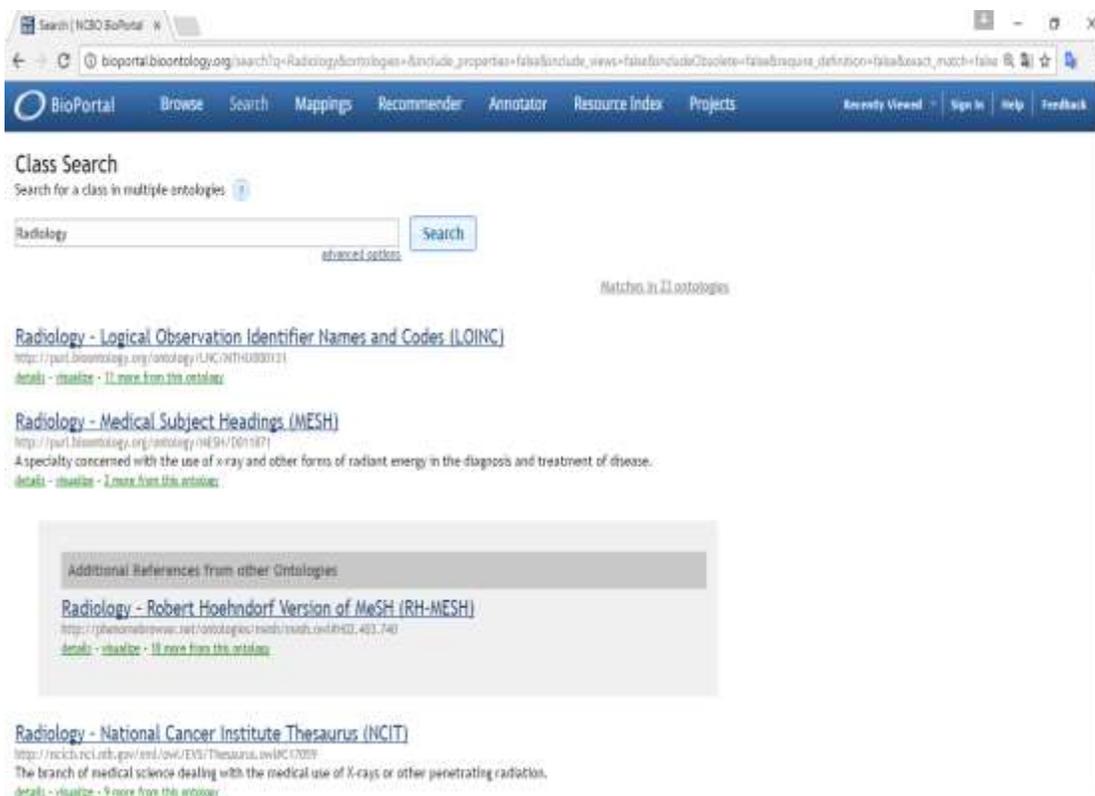


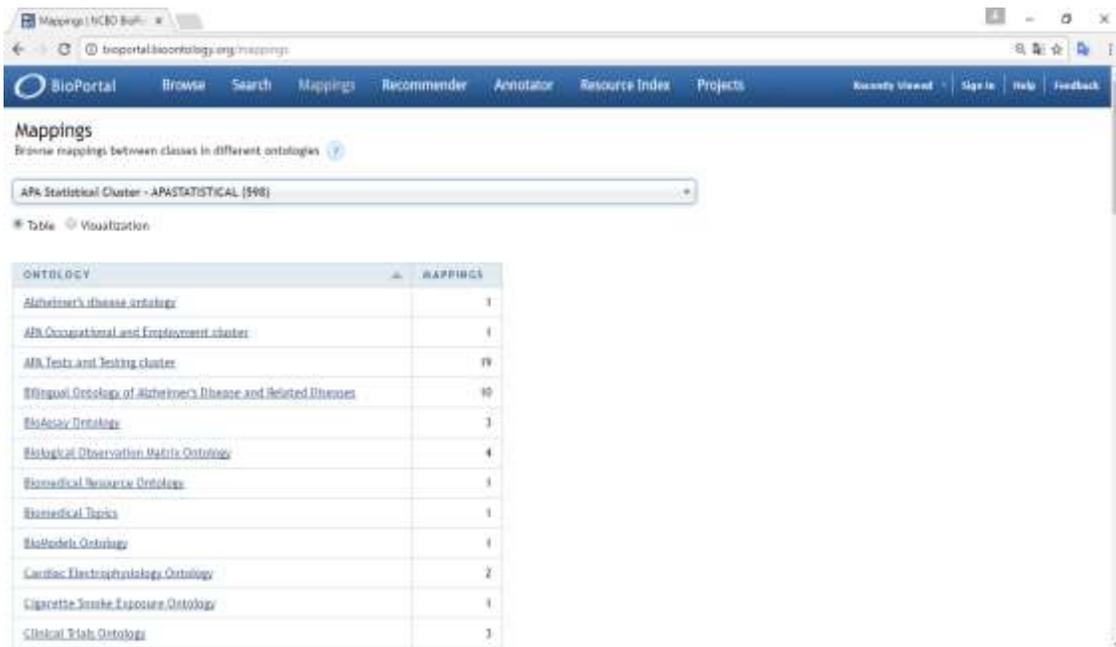
Figura A. 3 Pestaña Search de BioPortal.

3.4 Pestaña Mappings

El mapeo son asociaciones entre dos o más nombres de clases en diferentes ontologías. Esta asociación es típica, pero no siempre representa un grado de similitud entre los términos. El autor del mapeo define la semántica de un mapeo particular.

Para ver el mapeo de una ontología, seleccione la ontología de la lista desplegable. A continuación, aparecerá una tabla con todas las ontologías para los que existe al menos un mapeo entre estas ontologías y la ontología seleccionada, observé la Figura A.4. Seleccione una ontología de la tabla para navegar por el mapeo entre las dos ontologías, como se presenta en la Figura A.5.

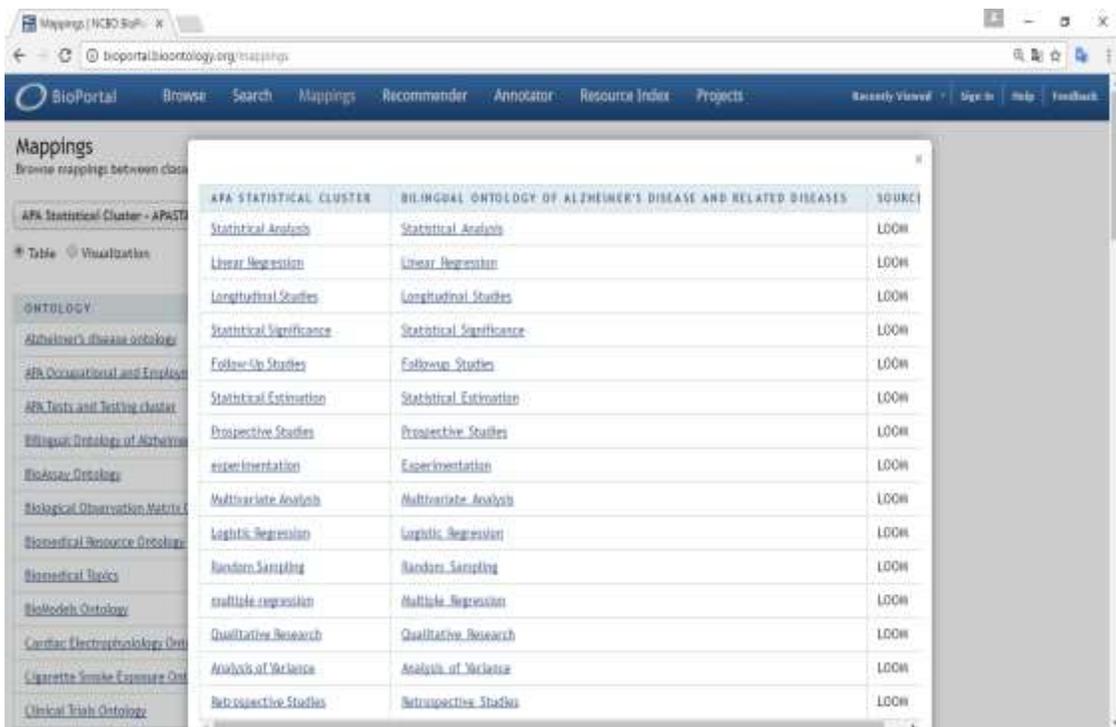
Anexo A. Descripción técnica de BioPortal



The screenshot shows the BioPortal Mappings page. The browser address bar is bioportal.bioontology.org/mappings. The page title is "Mappings" with the subtitle "Browse mappings between classes in different ontologies". A dropdown menu is set to "APA Statistical Cluster - APASTATISTICAL (598)". Below the dropdown, there are tabs for "Table" and "Visualization". A table lists various ontologies and their corresponding mapping counts.

ONTOLOGY	MAPPINGS
Alzheimer's disease ontology	1
APA Occupational and Employment cluster	1
APA Tests and Testing cluster	19
Bilingual Ontology of Alzheimer's Disease and Related Diseases	10
Biology Ontology	3
Biological Observation Matrix Ontology	4
Biomedical Resource Ontology	1
Biomedical Topics	1
BioModels Ontology	1
Cardiac Electrophysiology Ontology	2
Cigarette Smoke Exposure Ontology	1
Clinical Trials Ontology	3

Figura A. 4 Pestaña Mappings de BioPortal.



The screenshot shows a detailed view of the mappings between two ontologies. The browser address bar is bioportal.bioontology.org/mappings. The page title is "Mappings" with the subtitle "Browse mappings between classes in different ontologies". A dropdown menu is set to "APA Statistical Cluster - APASTATISTICAL (598)". Below the dropdown, there are tabs for "Table" and "Visualization". A table lists the mappings between the "APA STATISTICAL CLUSTER" and the "BILINGUAL ONTOLOGY OF ALZHEIMER'S DISEASE AND RELATED DISEASES".

APA STATISTICAL CLUSTER	BILINGUAL ONTOLOGY OF ALZHEIMER'S DISEASE AND RELATED DISEASES	SOURCE
Statistical Analysis	Statistical Analysis	LOOH
Linear Regression	Linear Regression	LOOH
Longitudinal Studies	Longitudinal Studies	LOOH
Statistical Significance	Statistical Significance	LOOH
Follow-Up Studies	Follow-up Studies	LOOH
Statistical Estimation	Statistical Estimation	LOOH
Prospective Studies	Prospective Studies	LOOH
experimentation	Experimentation	LOOH
Multivariate Analysis	Multivariate Analysis	LOOH
Logistic Regression	Logistic Regression	LOOH
Random Sampling	Random Sampling	LOOH
multiple regression	Multiple Regression	LOOH
Qualitative Research	Qualitative Research	LOOH
Analysis of Variance	Analysis of Variance	LOOH
Retrospective Studies	Retrospective Studies	LOOH

Figura A. 5 Tabla de mapeo de dos ontologías.

3.5 Pestaña Recommender

El *Recommender* toma como entrada un texto o una lista de palabras claves y sugiere ontologías apropiadas acorde a la consulta, note la Figura A.6. El algoritmo utilizado por el *Recommender* para la clasificación de las ontologías evalúa cada ontología adecuada a la entrada usando una combinación de cuatro criterios de evaluación:

Cobertura: ¿En qué medida la ontología representa la entrada? El Recommender invoca el servicio *Annotator NCBO* para obtener todas las anotaciones de entrada y luego usa esas anotaciones para calcular una puntuación de cobertura de cada ontología.

Aceptación: ¿Qué tan conocida y de confianza es la ontología para la comunidad biomédica? El número de visitas a la ontología en la página de *BioPortal* y la presencia o ausencia de la ontología en formato *UMLS* se utilizan para calcular una puntuación de aceptación para cada ontología.

Detalles de conocimiento: ¿Cuál es el nivel de detalle de la ontología proporcionado por los datos de entrada? Se calcula a partir del número de definiciones, sinónimos y propiedades de las clases de la ontología que cubren los datos de entrada.

Especialización: ¿Cómo es la ontología especializada en el dominio de los datos de entrada? Se calcula utilizando el número y tipo de anotaciones hechas a cada ontología y la clase position anotada en cada jerarquía de ontología. El resultado es normalizado por el tamaño de la ontología, con el fin de identificar pequeñas ontologías que se especializan con los datos de entrada.

The screenshot shows the BioPortal Recommender interface. The input field contains the text 'metology'. Below the input field, there is a table titled 'Recommended ontologies' with the following data:

POS.	ONTOLOGY	FINAL SCORE	COVERAGE SCORE	ACCEPTANCE SCORE	DETAIL SCORE	SPECIALIZATION SCORE	ANNOTATIONS	HIGHLIGHT ANNOTATIONS
1	NCIT	92.8	90.0	87.8	95.2	82.8	1	0

Figura A. 6 Pestaña Recommender de BioPortal.

3.6 Pestaña Annotator

Para generar anotaciones partiendo de un texto, sólo se tiene que introducir texto en la cuadro y pulse el botón enviar, observe la Figura A.7. El sistema buscara coincidencias comparando las palabras del texto con los nombre, sinónimo e identificadores de los términos en las ontologías.

The screenshot shows the BioPortal Annotator interface. At the top, there is a navigation bar with links for Browse, Search, Mappings, Recommender, Annotator, Resource Index, and Projects. The main heading is 'Annotator' with a subtext 'Get annotations for biomedical text with concepts from the ontologies'. Below this is a text input field containing the word 'radiology'. Underneath the input field are several configuration options: 'Select Ontologies' (with a search box), 'Select UMLS Semantic Types' (with a search box), and a set of checkboxes for 'Match Longest Only', 'Include Mappings', 'Exclude Numbers', 'Match Partial Words', and 'Exclude Synonyms'. There is also a dropdown for 'Include Ancestors Up To Level' set to 'None'. A 'Get Annotations' button is located below these options. The bottom section, titled 'Annotations', displays a table with 7 rows of results. The table has columns for CLASS, title, ONTOLOGY, title, TYPE, title, CONTEXT, MATCHED CLASS, title, MATCHED ONTOLOGY, and title. The results show matches for 'radiology' from various ontologies including Robert Hershner's Version of MeSH, Health Level Seven Reference Implementation Model, Logical Observation Identifier Names and Codes, International Classification of Primary Care - 2 PLUS, and National Cancer Institute Thesaurus.

CLASS	title	ONTOLOGY	title	TYPE	title	CONTEXT	MATCHED CLASS	title	MATCHED ONTOLOGY	title
Radiology		Robert Hershner's Version of MeSH		direct		radiology	Radiology		Robert Hershner's Version of MeSH	
Radiology		Health Level Seven Reference Implementation Model, Version 3		direct		radiology	Radiology		Health Level Seven Reference Implementation Model, Version 3	
Radiology		Logical Observation Identifier Names and Codes		direct		radiology	Radiology		Logical Observation Identifier Names and Codes	
Radiology		Health Level Seven Reference Implementation Model, Version 3		direct		radiology	Radiology		Health Level Seven Reference Implementation Model, Version 3	
radiology		International Classification of Primary Care - 2 PLUS		direct		radiology	radiology		International Classification of Primary Care - 2 PLUS	
Radiology		NCIC Thesaurus		direct		radiology	Radiology		NCIC Thesaurus	
Radiology		National Cancer Institute Thesaurus		direct		radiology	Radiology		National Cancer Institute Thesaurus	

Figura A. 7 Pestaña Annotator de BioPortal.

3.7 Pestaña Resource Index

El índice de recursos es un conjunto de notas pre-calculadas de recursos biomédicos seleccionados. Se ha utilizado el servicio web anotador para anotar estos recursos con todos los términos en todas las ontologías en *BioPortal*. Para recuperar los registros anotados con un término específico de la ontología, seleccione el término en el cuadro de búsqueda y presione el botón *Search Resource Index* como se aprecia en la Figura A.8.

Search Resource Index

Search biomedical resources

Radiology (HL7)

[advanced options](#)

Hierarchy Expansion
Find documents containing: Class(es) Class(es) and descendants

Search Results matches in 1 of 40 resources ([show resources with no matches](#))

RESOURCE	MATCHED RECORDS	% WATCHED	TOTAL RECORDS
Published	49,400	0.20%	24,358,440

Figura A. 8 Pestaña Resource Index de BioPortal.

3.6 Pestaña Projects

Esta página muestra una sección de los proyectos que hacen uso de la tecnología BioPortal. Para agregar su proyecto a esta pulse el botón *Create New Project*, como se exhibe en la Figura A.9. Debe ser un usuario registrado en BioPortal para crear su proyecto.

PROJECT	DESCRIPTION	CONTACTS	INSTITUTIONS	ONTOLOGIES
A Visualization Tool for Refining Mappings between Biomedical Ontologies Home Page	Ontology alignment, the task of identifying mappings between ontologies, is a key task for knowledge sharing between applications that use...	Watson W.K. Chua, Jung-jae Kim, Bui Duc Hieu	Nanyang Technological University	2
Access of OntoKBCF via a CCR-based EHR prototype Home Page	This is an exploration of introducing molecular genetic information into a simulated clinical setting- a Continuity of Care Record (CCR)...	Xia Jing, Stephen Kay, Nicholas Hardiker, Tom Marley, James Cimino	The University of Salford, NH	1
ACCORDYS Home Page	In the context of prenatal diagnosis of fetal malformation, knowledge of "similar" and resolved cases (i.e., previous cases with a...	Jean Charlet, Xavier Aim, Andreea Bodnari, Jean-Michel Daube, Louise Delger, Ferdinand Dhombrès, Egle Eensoo, Marie Gonzals, Brigitte Grau, Cyril Grouin, Marie-Christine Jaulent, Thomas Lavergne, Anne-Laure Ligozat, Irine Mairinka, Aurlie Mvol, Yves	Inserm, Paris, France	1
ACFSite Home Page	ACF			0
ACS-registry Home Page	ACS registry definitions	Gert-Jan Vos MD, PhD	AZN	1
AdaptEHR Home Page	Adaptive EHR (AdaptEHR) is built upon a context-based framework, integrating data from different sources in the patient record and		University of California, Los	3

Figura A. 9 Pestaña Projects de BioPortal.

3.7 Entornos de desarrollo con los que se construyó la aplicación NCBO

El servidor de la aplicación se ejecuta sobre el sistema operativo *CentOS* 6.6 de 64-bit (una versión del sistema operativo Linux) y cuenta con los siguientes entornos de desarrollo:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

3.8 Arquitectura de la aplicación NCBO BioPortal

La arquitectura NCBO se muestra en la Figura 6.1, los componentes que se representan visualmente en la Figura se anteceden con la letra, entre paréntesis, asociada en dicha figura.

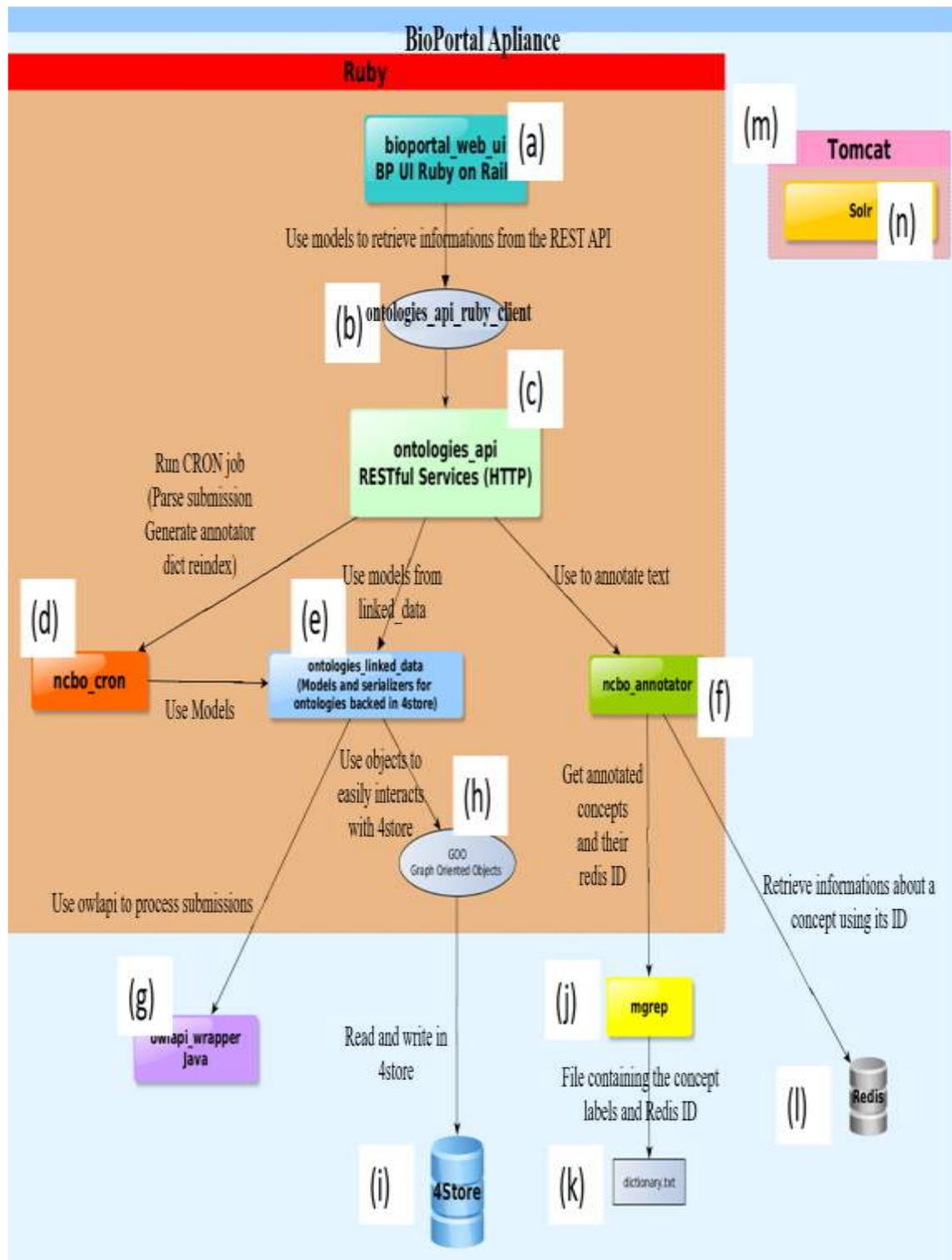


Figura A. 10 Arquitectura de la aplicación NCBO BioPortal.

(m) Tomcat 6.0.0.26

Es un contenedor *web* con soporte de *servlets* y *JSPs*. Incluye el compilador *Jasper* que compila *JSPs* convirtiéndolas en *servlets*. La jerarquía de directorios de instalación de *Tomcat* incluye:

bin – contiene el arranque, cierre y otros *scripts* y ejecutables.

common – contiene las clases comunes que puede utilizar Catalina (nombre del contenedor de *servlets* de *Tomcat*) y las aplicaciones *web*.

conf – tiene los ficheros *XML* y los correspondientes DTD (definición del tipo de documento) para la configuración de *Tomcat*.

log – logs de Catalina y de las aplicaciones *web*.

server – tiene las clases utilizadas solamente por Catalina.

shared – contiene las clases compartidas por todas las aplicaciones *web*.

webapps – es el directorio que contiene las aplicaciones *web*.

work – contiene el almacenamiento temporal de ficheros y directorios.

(n) Solr 4.10.4

Es una plataforma de búsqueda empresarial de código abierto, escrito en Java, desde el proyecto *Apache Lucene*. Sus principales características incluyen la búsqueda de texto completo, búsqueda facetada, indexación en tiempo real, el agrupamiento dinámico, la integración de bases de datos, características *NoSQL* y manejo de documentos enriquecidos (*Word*, *PDF*). Proporciona búsqueda distribuida y la replicación del índice. *Solr* está diseñado para la escalabilidad y tolerancia a fallos.

Solr se ejecuta como un servidor de búsqueda de texto completo independiente. Utiliza la biblioteca de búsqueda *Lucene Java* en su núcleo para la indexación de texto completo y la búsqueda tiene *REST* del mismo modelo *HTTP/XML* y *APIs JSON* que hacen que sean utilizables a partir de los lenguajes de programación externa de *Solr*, lo que permite que se adapte a muchos tipos de aplicaciones sin necesidad de programación Java y tiene una arquitectura *plugin* para permitir la personalización más avanzada.

Características conocidas

Usa la biblioteca de *Lucene* para la búsqueda de texto completo

Modo sin esquema y esquema de *REST API*, *JSON*, *XML*, *PHP*, *Ruby*, *Python*, *XSLT*, *Velocity* y formatos de salida binaria de *Java* personalizados a través de *HTTP*.

Java 7

Es una plataforma informática comercializada por primera vez en 1995 por *Sun Microsystems*. Hay muchas aplicaciones y sitios *web* que no funcionarán a menos que tenga *Java* instalado y cada día se crean más. *Java* es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, *Java* está en todas partes.

(a) Rails 3.2.x

Es un *framework* de desarrollo de aplicaciones *web*. *Rails* no solo es una gran opción cuando se quiere construir una aplicación de pila completa que se usa del lado del servidor mediante representaciones de plantillas *HTML*, sino también un gran ayudante para los *Java Script*, del lado del cliente o para aplicaciones nativas que sólo necesitan el *backend*, para usar *JSON*.

Ruby 2.1.5

Ruby es un lenguaje que incorpora tanto la programación funcional como la imperativa, resultado de mezclar los lenguajes de programación *Perl*, *Smalltalk*, *Eiffel*, *Ada* y *Lips*. En Ruby, todo es un objeto. Se pueden asignar propiedades y acciones a toda información y código. La programación orientada a objetos llama variables de instancia a las propiedades y las acciones son conocidas como métodos. *Ruby* no necesita declaración de variables. Utiliza convenciones simples para nombrar y determinar el alcance de las mismas.

Memcached

Es un sistema distribuido de propósito general para caché basado en memoria, es muy usado en la actualidad por múltiples sitios *web*. *Memcached* es empleado para el almacenamiento en caché de datos u objetos en la memoria *RAM*, reduciendo así las necesidades de acceso a un origen de datos externos (como una base de datos o una *API*). Su funcionamiento se basa en una tabla hash distribuida a lo largo de varios equipos. Conforme ésto se va llenando, los datos que más tiempo llevan sin ser utilizados se borrarán para dar espacio a los nuevos.

Normalmente, las aplicaciones comprueban primero si se pueden acceder a los datos a través de *Memcached* antes de recurrir a un almacén de datos más lento, como puede ser una base de datos. El sistema usa una arquitectura cliente-servidor. Los servidores mantienen un *array* asociativo clave-valor; los clientes agregan datos al *array* y acceden a él. Los clientes usan librerías para acceder a los servidores que por defecto utilizan el puerto 11211, cada cliente mantiene una lista de todos los servidores; los servidores no se comunican entre ellos. Si un cliente desea establecer o leer el valor correspondiente a cierta clave, la librería cliente primero hace un cálculo mediante un algoritmo *hash* para determinar el servidor que va a utilizar.

Entonces se pone en contacto con el servidor y éste usará otro *hash* para determinar dónde almacenar o leer el valor correspondiente.

(I) Redis 2.8.18

Es un motor de bases de datos en memoria, basado en el almacenamiento en tablas *hash* (clave/valor) pero que opcionalmente puede usarse como una base de datos durable o persistente. El modelo de datos de *Redis* se basa en la estructura de datos del tipo diccionario o tabla de hashes que relaciona una llave a un contenido almacenado en un índice. La principal diferencia entre *Redis* y otros sistemas similares son los valores que no están limitados a ser de tipo *string*, otros tipos de datos que soporta *Redis* son: Listas (*Lists*) de *string*, *sets* de *strings* (colecciones de elementos desordenados no repetidos), *hashes* donde la llave y el valor son de tipo *string*.

El tipo de valor determina las operaciones (los comandos) que están disponibles. *Redis* soporta operaciones atómicas de alto nivel del lado del servidor, como inserciones, uniones, diferencias entre *sets* y listas ordenadas.

Nginx 1.6.3

Es un software para proporcionar un servidor web HTTP (Hypertext Transfer Protocol o Protocolo de Transferencia de Hipertexto) y un servidor proxy inverso. Puede actuar como un servidor proxy genérico para protocolos *TCP/UDP* y como un servidor proxy para correos con los protocolos *SMTP* (Simple Mail Transfer Protocol o Protocolo para Transferencia Simple de Correo), *POP3* (Post Office Protocol o Protocolo de Oficina de Correo) e *IMAP* (Internet Message Access Protocol o Protocolo de acceso a mensajes de internet).

Características

Puede ser desplegado para servir contenido dinámico *HTTP* en la red usando *FastCGI*, manejadores *SCGI* para los scripts, servidores de aplicaciones *WSGI* (son interfaces simple y universal entre los servidores web y las aplicaciones web o frameworks) o módulos *Phusion Passenger* y puede servir como un equilibrador de carga de software. *Nginx* utiliza un enfoque orientado a eventos asíncronos para el manejo de solicitudes. *Nginx* es arquitectura modular orientada a eventos puede proporcionar un rendimiento más predecible bajo altas cargas.

(i) 4store

Es una base de datos *RDF* que fue diseñada por *Steve Harris* y desarrollada en *Garlik* para respaldar sus aplicaciones de *Web Semántica*. *Harris* ha estado proporcionando la plataforma base por cerca de 3 años. La plataforma base a veces mantiene y ejecuta consultas sobre bases

de datos compuestas por 15 mil millones de triples de datos RDF (15GT). *4store* hace uso de las bibliotecas *Raptor* y *Rasqal* que se han desarrollado para *Redland*.

Passenger/Apache.

Informalmente conocido también como *mod-rails* y *mod-rack* entre la comunidad *Ruby*. *Passenger* es un servidor *web* libre y servidor de aplicaciones con soporte para *Ruby*, *Python* y *Node.js*. Está diseñado para integrarse en el servidor *HTTP Apache* o el servidor *web Nginx*, pero también tiene un modo para el funcionamiento independiente sin un servidor *web* externo. Originalmente diseñado para aplicaciones *web* integradas en *Ruby on Rails*, más tarde se amplió para apoyar los marcos *web* arbitrarios de *Ruby* a través de la interfaz de *rack*. *Passenger* es un módulo de *Apache 2.x* que le permite ejecutar aplicaciones *Rails* o *rack* dentro de un servidor *web* de propósito general, como *HTTP Apache* o *nginx*.

Descripción de la aplicación BioPortal

La interfaz web BioPortal está construida principalmente en *Ruby on Rails*, que es un marco (*framework*) basado en la arquitectura modelo vista controlador (MVC), creado para la *Web*. En el desarrollo de la vista dentro de la arquitectura MVC, la creación de la interfaz de BioPortal utiliza plantillas *ERB* y *HAML*.

En los párrafos siguientes se mencionan los elementos (a, b, c, d, e, f, g, h, i, j, k, l, m, n), de la arquitectura de la aplicación NCBO BioPortal mostrada en la Figura 6.1, entre paréntesis (Ej. (a), (b),..., (n)), junto a una descripción textual breve de cada elemento. Algunos de los elementos son gemas, cuyo significado, para el ambiente *Ruby on Rails*, es equivalente a una librería en el lenguaje de programación java.

En la carpeta con el nombre de *bioportal_web_ui* (a), que se localiza en el servidor de la aplicación *NCBO*, se encuentra el código de la interfaz del usuario de *BioPortal*. La interfaz *BioPortal* usa modelos para recuperar información desde la API REST, con el nombre de *ontologies_api_ruby_client* (b). En la carpeta *ontologies_api* (c), se ubican los servicios *REST FULL (HTTP)*. Estos servicios utilizan la gema *ncbo_cron* (d) para ejecutar el demonio *CRON* el cuál analiza lo que se recibe de *ontologies_api* (b), para presentar el análisis sintáctico y genera una anotación a través del ejecutable *dict* y re indexa *solr* (n).

La gema *ncbo_cron* (d) y los servicios *RESTful* (c) usan modelos y serializadores para el respaldo de las ontologías en *4store* (i) que genera la gema *ontologies_linked_data* (e). La gema *ontologies_linked_data* (e) utiliza la gema *owlapi_wrapper Java* (g) para procesar peticiones. También emplea la biblioteca Grafica Orientada a Objetos (GOO) (h) para facilitar la iteración con *4store* (i). GOO lee y escribe en *4store* (i).

Los servicios *RESTful* (c) también hacen uso de la gema *ncbo_annotador* (f) para realizar anotaciones de texto y la gema *ncbo_annotador* (f) obtiene los conceptos y sus identificadores anotados en *Redis* (l) a través del ejecutable *dict* que se encuentra en la carpeta *mgrep* (j). En la carpeta *mgrep* (j) también se localiza el archivo *dictionary.txt* (k) que contiene las etiquetas de los conceptos con su identificador en *Redis* (l) (un ejemplo es 277843155 PARTICIPANTES). La gema *ncbo_annotador* (f) recupera la información de un concepto utilizando su identificador en *Redis* (l). *Redis* (l) se utiliza principalmente como una memoria caché y todos los datos que contiene están disponible en el *backend 4store* (i).

Tomcat (m) es el contenedor *web* con soporte de *servlets* y *Solr* (n), que se utiliza de servidor *web* de búsqueda de texto en la plataforma. *Solr* (n) utiliza *carrot2* que es un motor de búsqueda de Internet en un clúster de máquinas basado en el proyecto de código abierto de *Stanislaw Osinski* y *Dawid Weiss*. El objetivo del proyecto es organizar los resultados de las búsquedas en varios grupos de documentos relacionados. Existen tres algoritmos de agrupamiento disponibles en el sistema: *lingo*, *kmeans* y *stc*.

Anexo B. Instalación de BioPortal

6.3 Instalación de la aplicación NCBO BioPortal

Requisitos del sistema:

- Para que la aplicación NCBO BioPortal se instale es necesario tener como mínimo:
 - 2 CPU (2 GHz)
 - 4GB RAM
 - Espacio en disco duro de 20 GB
- Para uso pesado se recomienda tener:
 - 4 CPU (3 GHz)
 - 8GB RAM (o más dependiendo del tamaño o número de ontologías)
 - Espacio en disco duro de 20GB (o más dependiendo del tamaño o número de ontologías)
- Instalar el entorno de virtualización VMware Workstation 12 Pro (u otra versión superior) en la computadora, para instalar la máquina virtual NCBO.

Pasos para la instalación

1. En la interfaz de VMware Workstation dar clic en el icono que tiene la leyenda *Open a Virtual Machine* como se muestra en la Figura B.1.

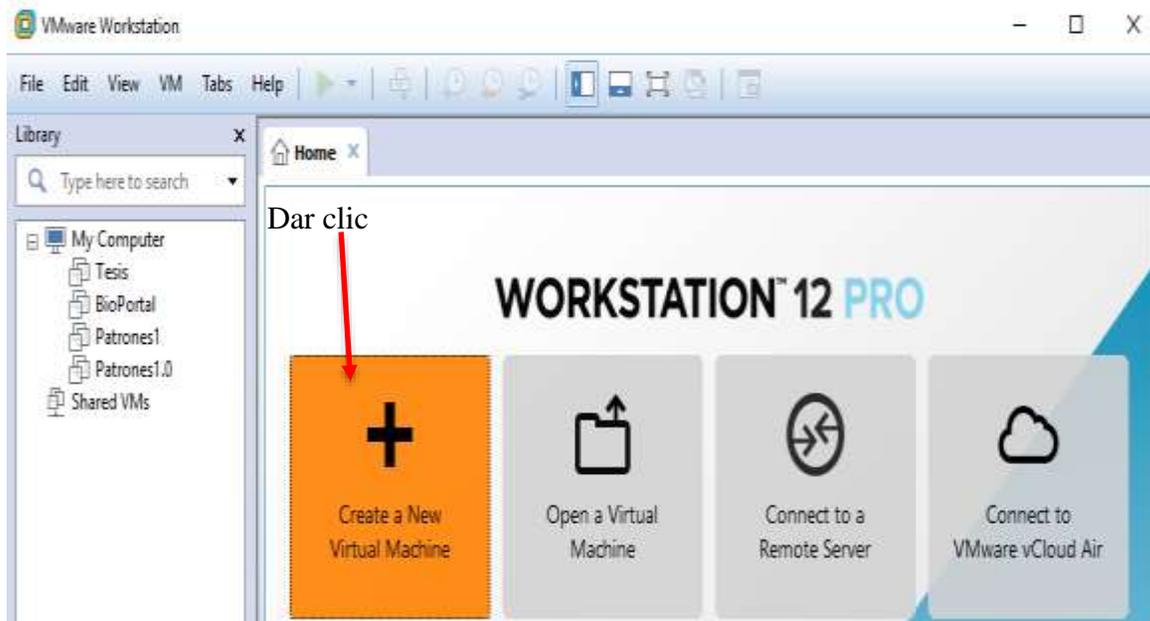


Figura B. 1 Crear una máquina virtual.

2. Se muestra una ventana, que permite seleccionar la ubicación en donde se tiene el archivo de la máquina virtual de la aplicación NCBO, como se muestra en la Figura B.2.

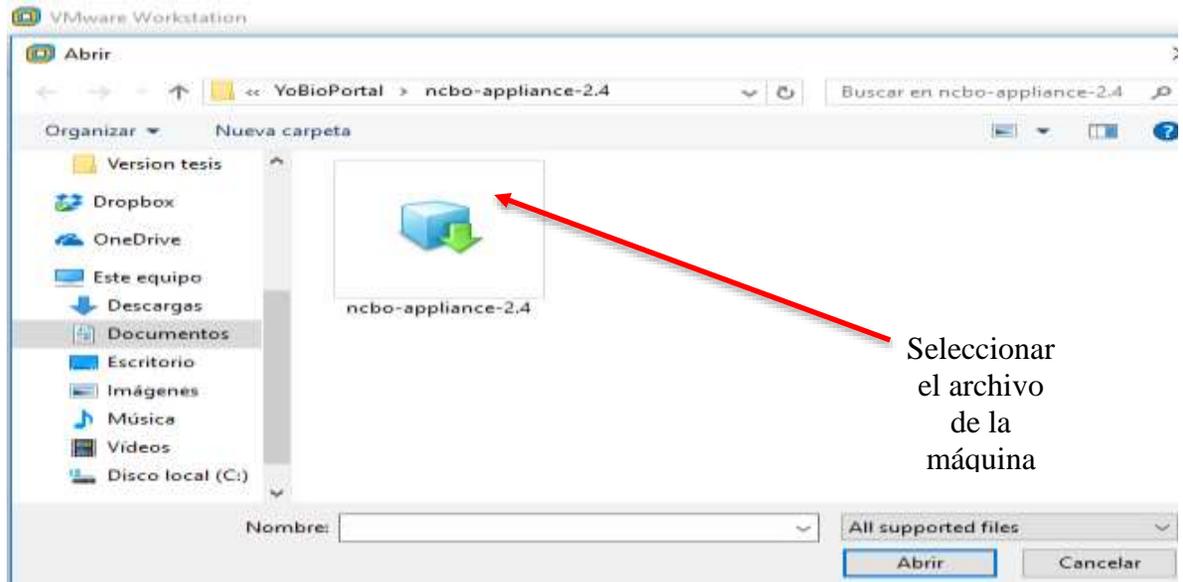


Figura B. 2 Ruta donde se localiza el archivo de la máquina virtual

3. Elegir la forma en que se instala la aplicación o sistema. En este caso se eligió la opción que tiene la leyenda *I will install the operating system later* y dar clic en el botón *Next*. Como se muestra en la Figura B.3.

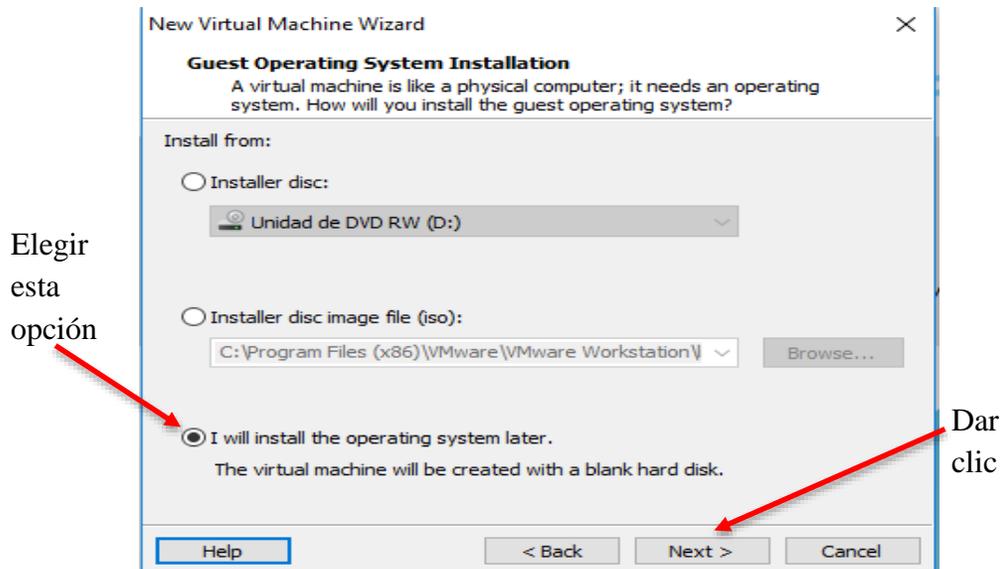


Figura B. 3 Forma de instalación del sistema operativo invitado.

4. Seleccionar el tipo de sistema operativo que se instalará en la máquina virtual, en este caso seleccionar el sistema operativo *Linux* y elegir la versión *CentOS*, dar clic en el botón *Next* como se muestra en la Figura B.4.

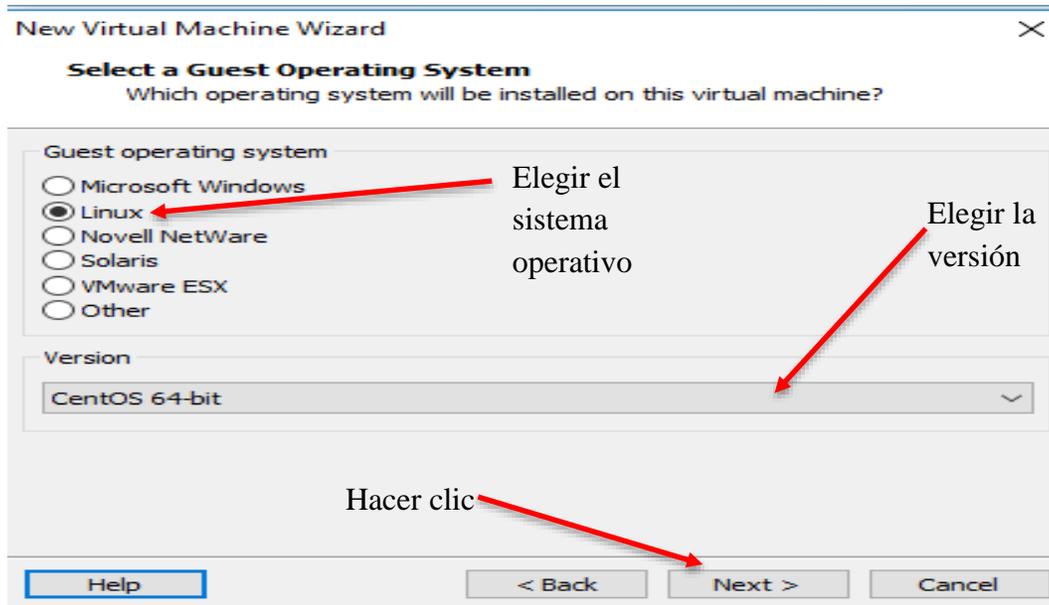


Figura B. 4 Seleccionar el sistema operativo a instalar.

5. Asignar un nombre a la máquina virtual y elegir la ruta de localización, para este ejemplo se asignó el nombre de *BioPortal* y se dejó la ruta de localización predeterminada, después se hizo clic en el botón *Import*, como se muestra en la Figura B.5.

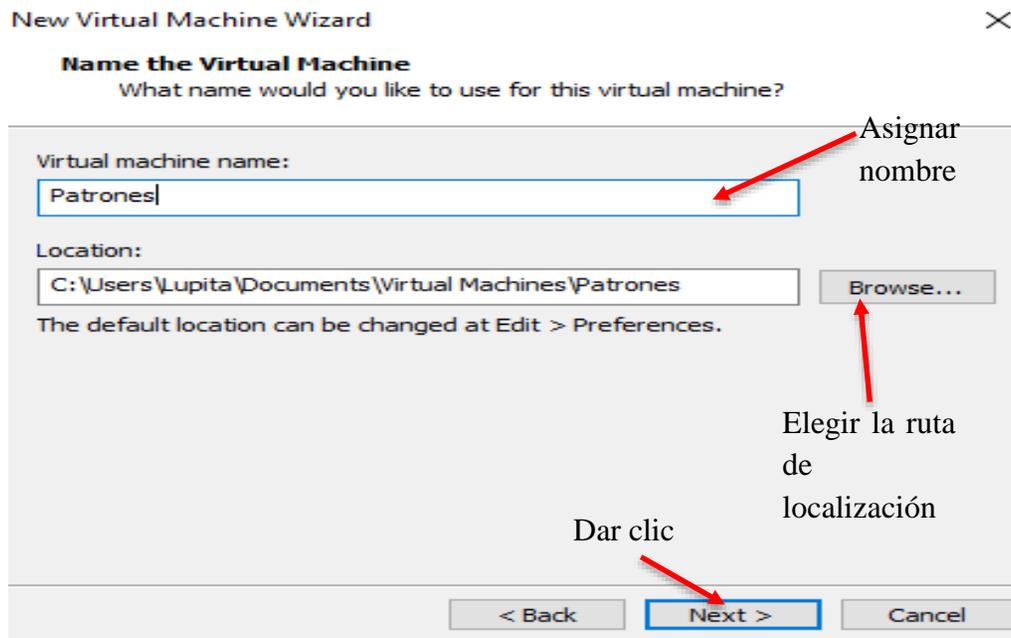


Figura B. 5 Asignar nombre a la máquina virtual.

6. En la interfaz de *VMware Workstation* del lado derecho, en la parte superior, se muestra una pestaña con el nombre que se le asignó a la máquina virtual. En el área de despliegue de esta pestaña, se puede ver las características de los dispositivos que cuenta la máquina virtual que se acaba de crear. Pero como estas no corresponden con las características que se requieren, para instalar la aplicación *NCBO*, se debe modificar haciendo clic en *Edit virtual machines settings*, como se muestra en la Figura B.6.

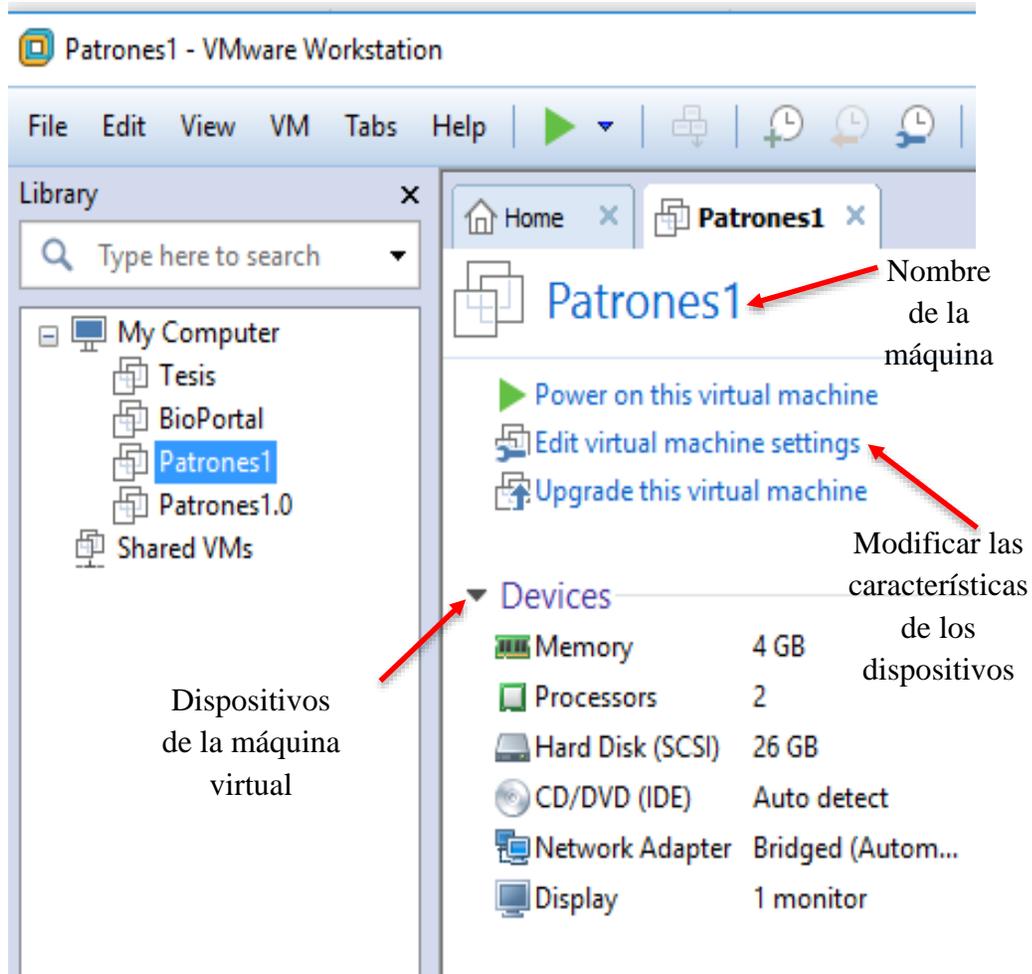


Figura B. 6 Características de los dispositivos de la máquina virtual.

7. En la ventana de configuración de la máquina virtual se asigna la cantidad 4 GB de memoria que es lo que se requiere como mínimo para la instalación de la aplicación NCBO y se seleccionan dos procesadores. Los dispositivos no mencionados no tienen mayor importancia por lo cual su configuración se deja tal y como está. Para guardar los cambios dar clic en el botón *OK*. Ver Figura B.7.

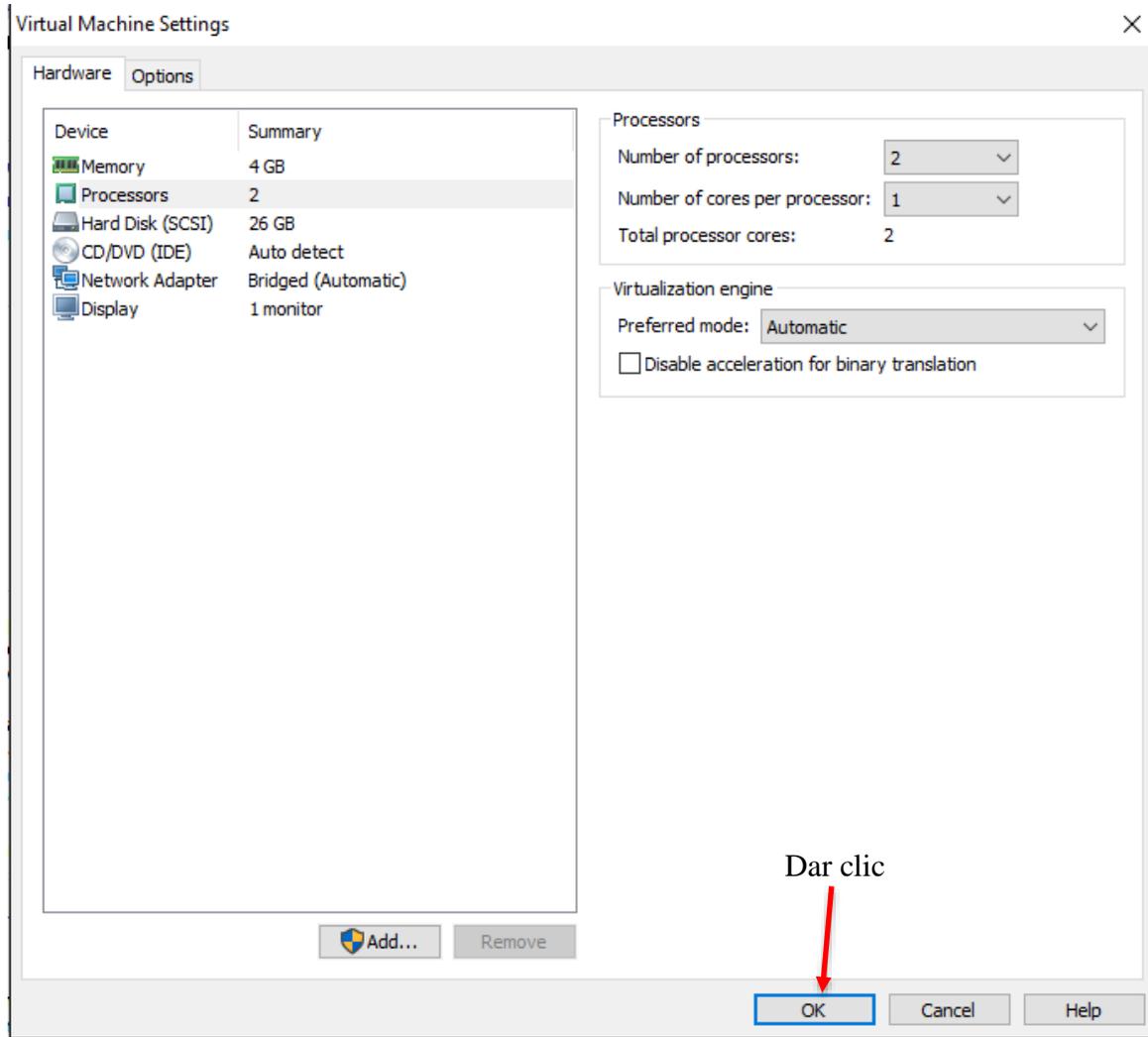


Figura B. 7 Configuración de los dispositivos de la máquina virtual.

8. Dar clic en la leyenda *Power on this virtual machine*, para iniciar la instalación de la aplicación NCBO, como se muestra en la Figura B.8. Una vez terminada la instalación con esa misma opción se enciende la máquina virtual.

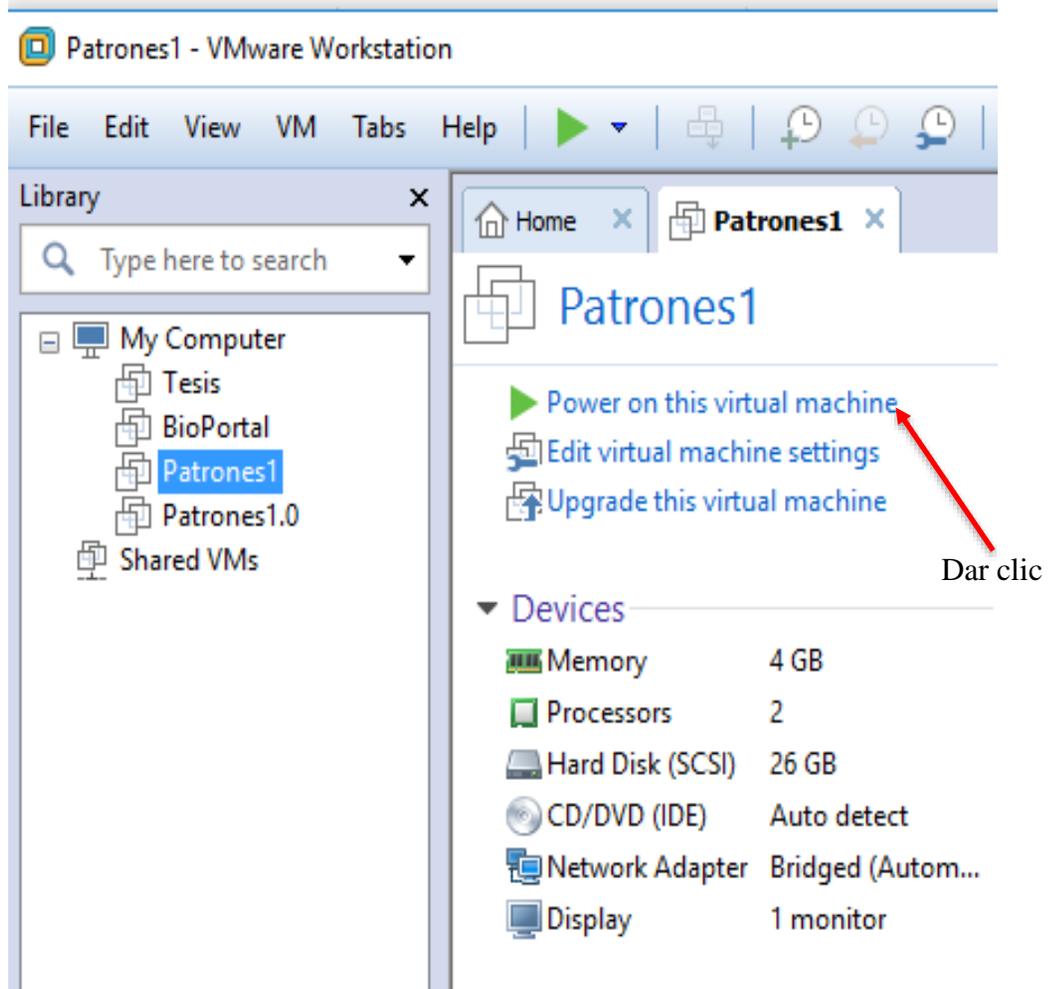


Figura B. 8 Inicio de la instalación de la aplicación NCBO.

9. En este paso se muestra una ventana en donde se selecciona por medio del teclado la opción *Quit*. En la siguiente ventana seleccionar la opción *Next* y finalmente seleccionar la opción *OK*.

10. Cuando el proceso de instalación ha finalizado se muestra la interfaz del servidor como se puede ver en la Figura B.9. En la interfaz del servidor se pide asignar una contraseña al usuario root, esta contraseña la solicitará el servidor cada vez que se desee abrir una sesión en el servidor y solicitará el nombre del usuario, que de forma predeterminada es root.

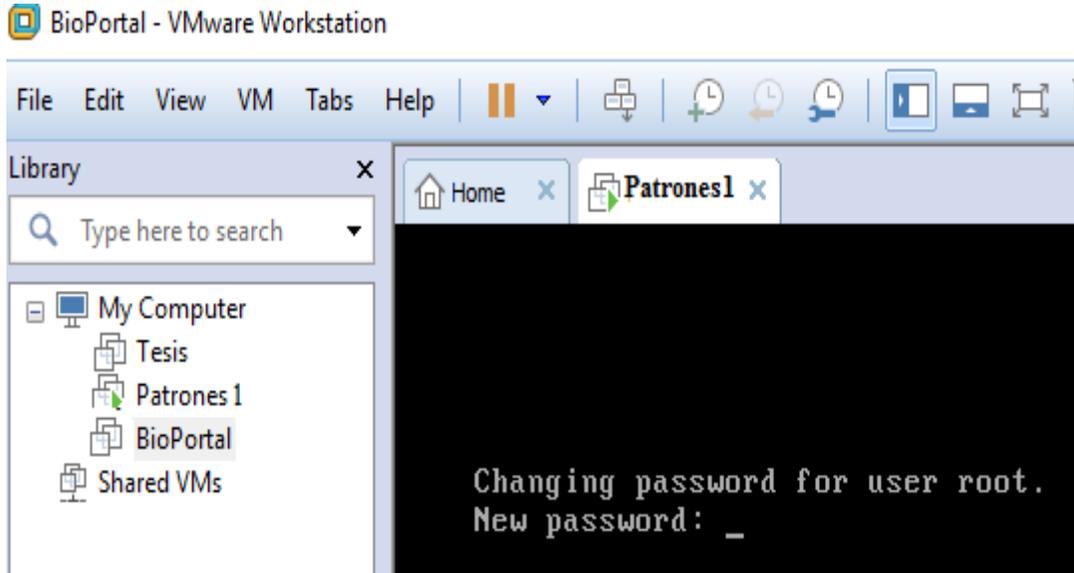


Figura B. 9 Crear contraseña para acceder al servidor.

Para iniciar una sesión en el servidor se enciende la máquina virtual como se señaló en el punto 8. Estando en la interfaz del servidor, se solicita ingresar el nombre del usuario que es root y la contraseña que se le asignó al usuario root (ver Figura B.10).

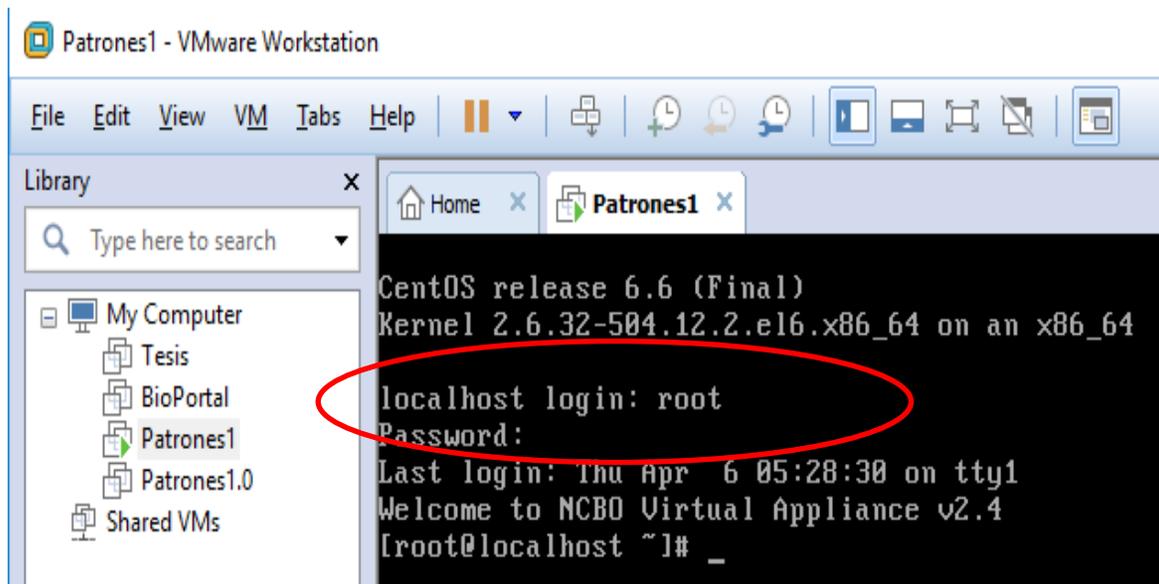


Figura B. 10 Ingreso a la aplicación virtual NCBO.

12 Obtener la IP local ingresando en la terminal el siguiente comando: `ip address show eth0 | awk '/inet/{printf $2}' | cut -d/ -f1`, como se muestra en la Figura B.11.

```
CentOS release 6.6 (Final)
Kernel 2.6.32-504.12.2.el6.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Apr 6 05:28:30 on tty1
Welcome to NCBO Virtual Appliance v2.4
[root@localhost ~]# ip address show eth0 | awk '/inet/{printf $2}' | cut -d/ -f1
10.160.122.127
[root@localhost ~]#
```

Figura B. 11 Ingresar comando para obtener la IP local.

13. La IP que devuelve la terminal se ingresa en un navegador para que se tenga acceso al portal, como se muestra en la Figura B.12.

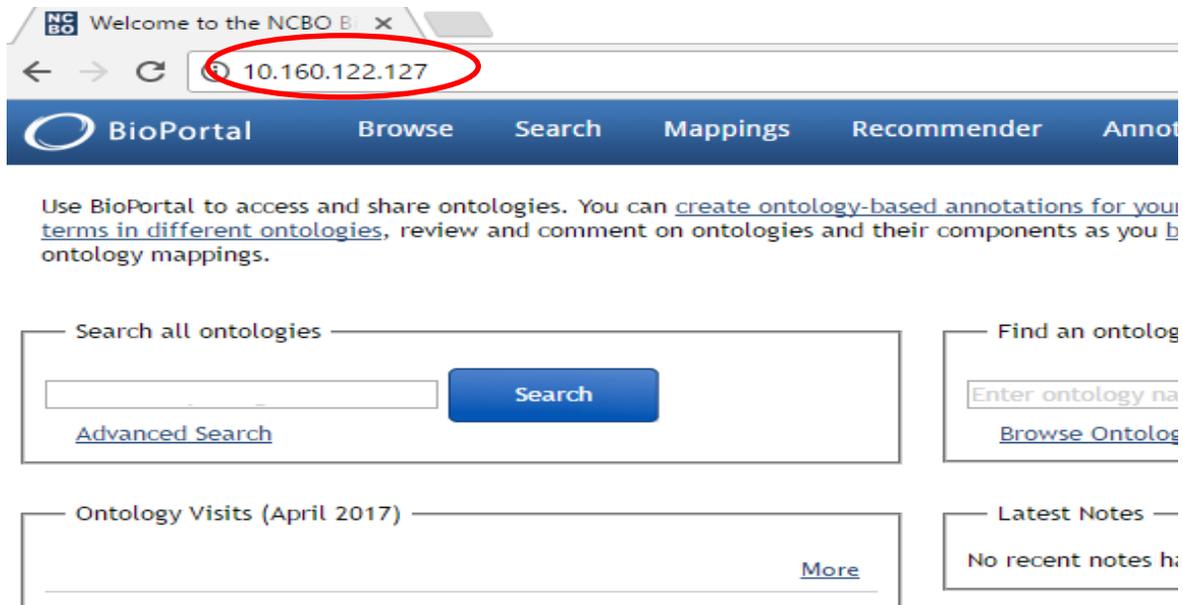


Figura B. 12 Interfaz del portal.

6.4 Ingresar como usuario Administrador en la aplicación NCBO BioPortal

El ingreso al portal como usuario administrador, por primera ocasión, es a través del nombre de la cuenta *admin* con contraseña *changeme*, posteriormente se puede hacer el cambio de contraseña en el apartado *Account*, ubicando el enlace *Edit Information*, ver Figura B.13.

FIRST NAME:	
LAST NAME:	
EMAIL ADDRESS:	admin@example.org
ACCOUNT NAME:	admin
Edit Information	

Figura B. 13 Editar la información del usuario administrador.

Al dar clic sobre el enlace *Edit Information*, que se muestra en la Figura B.13, se presenta una interfaz para el registro del nombre y apellido del administrador, en la parte inferior existe un enlace con el nombre de *Change Password*, como se muestra en la Figura B.14.

FIRST NAME	<input type="text"/>
LAST NAME	<input type="text"/>
EMAIL ADDRESS *	admin@example.org
<input type="button" value="Update"/>	Change Password

Figura B. 14 Registrar nombre y apellido del usuario administrador.

Haciendo clic en el enlace *Change Password* se observan los campos para el cambio de contraseña, como se nota en la Figura B.15.

PASSWORD: *	<input type="password"/>
RE-ENTER PASSWORD: *	<input type="password"/>
<input type="button" value="Update"/>	

Figura B. 15 Cambiar contraseña del usuario administrador.

Anexo C. Plan de pruebas

1. Introducción

En el presente documento se expone el plan de pruebas a cubrir para asegurar el funcionamiento del producto de software obtenido de la investigación “Almacenamiento y Uso de Patrones de Análisis Utilizando Ontologías”. Se define el alcance del plan de pruebas, detalles del plan de pruebas, los criterios de aceptación, suspensión y reanudación; y se nombra los elementos que se toman en cuenta para este plan de pruebas.

1.1 Identificador del Documento

A continuación en la Tabla C.1 se lista la nomenclatura utilizada para identificar los documentos en la etapa de pruebas.

Tabla C. 1 Nomenclatura para identificar los documentos generados.

Nomenclatura	Identificador	Descripción
AB	Tipo de documento	TP - Plan de Pruebas (<i>Test Plan</i>) TD - Diseño de Pruebas (<i>Test Desing</i>) TC – Casos de Prueba (<i>Test Case</i>) TPr – Procedimiento de Pruebas (<i>Test Log</i>) TL – Registro de Pruebas (<i>Test Report</i>) TR – Reporte de Pruebas (<i>Test Report</i>)
NM	Numeración	Representa el número correspondiente al documento.

1.2 Alcance

Verificar la funcionalidad y cumplimiento de los requerimientos funcionales que se especificaron para la implementación del almacén de ontologías de descripciones de patrones de análisis, siguiendo la jerarquía de ontologías propuesta.

1.3 Referencias

IEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Documentation.

2. Detalles del Plan de Pruebas del Sistema

De la sección 2.1.1 a 2.1.5 se definen los requerimientos funcionales a probar. Posteriormente se muestra la matriz de trazabilidad de pruebas con el fin de conocer que requerimientos son cubiertos acorde al caso de prueba asignado. Finalmente se muestran los criterios de

aceptación, suspensión y reanudación que se tomaron en cuenta para las pruebas y los respectivos entregables.

2.1 Requerimientos Funcionales a probar

El plan de pruebas consiste en evidenciar el cumplimiento de los requerimientos de software definidos en el documento SRS, los cuales son:

2.1.1 Crear cuentas de usuarios

Este requerimiento crear cuentas de usuario, a partir de solicitarle al usuario su nombre, primer apellido, dirección de correo electrónico, identificador de usuario y la contraseña que desea para creación de la cuenta.

2.1.2 Representar jerarquía de ontologías propuesta

Se requiere en primer lugar guardar la ontología de la plantilla, segundo lugar las ontologías de los individuos de la plantilla, tercer lugar las ontologías de los patrones de análisis, cuarto lugar las ontologías los individuos de cada ontología del patrón de análisis y en quinto lugar las ontologías de casos específicos de los individuos de cada patrón de análisis.

Nota: Este requerimiento no se conseguirá completo porque las ontologías que se requieren guardan en cuarto y quinto lugar no se tienen hasta este momento, en consecuencia solo se determinara el lugar donde estas se localizarán cuando se obtengan en futuras investigaciones.

2.1.3 Realizar consultas al repositorio de ontologías

El usuario producirá consultas ingresando el nombre de una clase de una ontología (palabra o concepto), para localizar una ontología en particular. La búsqueda a través de un texto breve y del nombre de alguna propiedad se abarcará en futuras investigaciones.

La descarga de archivos en formato *XMI* que son diagramas *GRL*, *UCM* y *UML* se pospone para futuras investigaciones.

2.1.4 Descargar archivos

Se requiere que se brinde la opción de descargar por separado el archivo de cada ontología en formato *owl*. Es preciso para el momento de generar otra ontología se basen en la estructura de acuerdo al nivel que se encuentren en la jerarquía.

2.1.5 Aportar ontologías por usuarios registrados

Guardar una aportación por parte del usuario en el repositorio, se requiere guardar el nombre de la ontología, una breve descripción acerca de la ontología y un archivo en formato *owl*.

2.1.6 Actualizar información de cada ontología

Se debe verificar que el usuario a actualizar la información de cierta ontología sea el autor de la aportación de la ontología.

2.1.7 Eliminar ontología

La eliminación de la ontología se ejecuta a través del encargado de administrar la aplicación, por lo tanto si un usuario que aporó una ontología desea eliminarla tiene que ponerse en contacto a través del correo electrónico con el administrador.

2.2 Matriz de Trazabilidad de Pruebas

En la Tabla B.2 se provee la lista de requerimientos que se establecieron para la aplicación y el identificador del caso de prueba para cada requerimiento.

Tabla C. 2 Matriz de trazabilidad de pruebas.

Requerimiento	Caso de Prueba
1. Crear cuentas de usuarios	TC-01
2. Representar jerarquía de ontologías propuesta	TC-02
3. Realizar consultas al repositorio de ontologías	TC-03
4. Descargar archivos	TC-04
5. Aportar ontologías por usuarios registrados	TC-05
6. Actualizar información de cada ontología	TC-06
7. Eliminar ontología	TC-07

2.3 Criterio de Aceptación

Los casos de prueba son aprobados si cumplen con cada una de las funcionalidades especificadas en los requerimientos sin ninguna clase de contratiempo.

2.4 Criterio para Suspensión y Reanudación de Pruebas

La suspensión de las pruebas se rige por la presencia de fallas en los elementos que impidan el cumplimiento de las funcionalidades. Al registrarse una falla se suspende la prueba y se reanuda al ser corregida.

2.5 Entregables

El resultado de las actividades correspondientes a las pruebas es la producción de los siguientes entregables:

1. Plan de Pruebas
2. Diseño de Pruebas
3. Casos de Pruebas
4. Procedimiento de Pruebas
5. Registro de Pruebas
6. Reporte de Pruebas

3. Administración de Pruebas

En esta sección se define las tareas de prueba a realizar. También se define la infraestructura utilizada para la etapa de pruebas.

3.1 Actividades Planificadas

Las actividades que son necesarias para preparar y realizar las pruebas son:

- Generación del plan de pruebas.
- Diseño de casos de pruebas.
- Ejecución de casos de pruebas.
- Generación de reporte de pruebas.

Las actividades son ejecutadas por el autor del proyecto.

3.2 Infraestructura

La infraestructura para las pruebas se especifica a continuación:

- Software
 - VMware Workstation 12 Pro 12.1.0 build-3272444

El servidor de la aplicación se ejecuta sobre el sistema operativo *CentOS* 6.6 de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
 - *Solr 4.10.4*
 - *Java 7*

- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

- Hardware
 - Procesador: Intel(R) Core(TM) i7-5500 CPU @2.40GHz 2.40 GHz
 - Memoria RAM en anfitrión: 8.00 GB
 - Memoria RAM en huésped: 4.00 GB
 - Disco Duro en anfitrión: 237 GB
 - Disco Duro en huésped: 26 GB

Anexo D. Diseño de pruebas

1. Introducción

En el presente documento se identifican las características a probar, se determina la estrategia para realizar las pruebas y el criterio de aceptación de las pruebas.

1.1 Identificador de documento

El identificador del documento pactado en la nomenclatura especificada en el Plan de Pruebas es: TD-01.

1.2 Alcance

Enfocar las características que permiten el cumplimiento de los requerimientos funcionales definidos en el documento de Especificación de Requerimientos de Software. Las pruebas utilizan el criterio de inspección visual para corroborar la existencia de los registros esperados.

1.3 Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Documentation.

2. Detalles del diseño de la prueba

Esta sección describe las características a probar, los casos de pruebas, los criterios de aceptación y entregables.

2.1 Características a probar

El diseño de pruebas engloba las características referentes a los siguientes requerimientos funcionales: crear cuentas de usuario, representar la jerarquía de ontologías propuesta, realizar consultas al repositorio de ontologías, descargar archivos, aportar ontologías por usuarios registrados, actualizar información de cada ontología y eliminar ontología. A continuación se muestra en la Tabla D.1 la matriz de trazabilidad de pruebas, en la cual se describen las características a probar, se menciona el nombre del caso y procedimiento de prueba correspondiente a cada requerimiento funcional.

Anexo D. Diseño de pruebas

Tabla D. 1 Matriz de trazabilidad

Requerimiento	Funcionalidad	Caso de prueba	Procedimiento de prueba
1. Crear cuentas de usuario (cualquier usuario puede ejecutar esta acción)	<ol style="list-style-type: none"> 1. Solicitar el nombre de pila del usuario. 2. Solicitar el primer apellido. 3. Solicitar la dirección del correo electrónico. 4. Solicitar el identificador de usuario. 5. Solicitar contraseña. 	TC-01	TPr-01
2. Representar la jerarquía de ontologías propuesta (es la estructura que se propuso como propuesta de solución)	<ol style="list-style-type: none"> 1. Probar la existencia de la categoría para almacenar la ontología de la plantilla. 2. Probar la existencia de la categoría para almacenar los individuos de la plantilla. 3. Probar la existencia de la categoría para almacenar ontologías del patrón de análisis. 4. Probar la existencia de la categoría para almacenar individuos de cada patrón de análisis. 5. Probar la existencia de la categoría para almacenar las ontologías de casos específicos de los individuos de cada patrón de análisis. 	TC-02	TPr-02
3. Realizar consultas al repositorio de ontologías (cualquier usuario puede ejecutar esta acción)	Realizar consultas a través del nombre de una clase (palabra o concepto) de la ontología, para localizar una ontología en particular.	TC-03	TPr-03
4. Descargar archivos (cualquier usuario puede ejecutar esta acción)	Descargar el archivo de la ontología en formato OWL.	TC-04	TPr-04
5. Aportar ontologías por usuarios registrados (esta opción solo los usuarios registrados pueden realizarla)	<ol style="list-style-type: none"> 1. Requerir el nombre de la ontología. 2. Requerir una breve descripción acerca de la ontología. 3. Requerir el archivo de la ontología con formato OWL. 4. Guardar los tres puntos anteriores en la aplicación. 	TC-05	TPr-05
6. Actualizar información de cada ontología (acción realizada por el usuario que ingreso la ontología)	Actualizar la información de cada ontología si el autor respectivo lo desea.	TC-06	TPr-06
7. Eliminar ontología (solo el administrador de la aplicación puede realizar esta acción)	Eliminar ontologías.	TC-07	TPr-07

2.2 Estrategia de pruebas

La realización de las pruebas será de la siguiente manera: en cada etapa se realiza una o varias actividades para probar cada uno de los requerimientos funcionales establecidos. En algunas actividades se utilizarán datos ficticios para verificar la actividad. La Tabla D.2 expone las pruebas a realizar.

Tabla D. 2 Pruebas a realizar correspondiente a cada actividad de los requerimientos funcionales.

Prueba	Etapa	Descripción
1. Crear cuenta de usuario.	Crear cuentas de usuarios	Verificar con la creación de una cuenta de usuario en la aplicación, el registro incluye el primer nombre de pila del usuario, su primer apellido, una dirección de correo electrónico válido, identificador de usuario y una contraseña.
2. Zona para almacenar la ontología de la plantilla.	Representar la jerarquía propuesta.	Comprobar de manera visual la existencia de la ontología de la plantilla en la categoría I-O Ontología de la plantilla.
3. Zona para almacenar los individuos de la plantilla.	Representar la jerarquía propuesta.	Examinar de manera visual la existencia de la categoría I-I Individuos de la Plantilla para el alojamiento de los individuos de la plantilla. Ingresar en esta categoría al menos un individuo de los dos existentes.
4. Zona para almacenar ontologías del patrón de análisis.	Representar la jerarquía propuesta.	Probar el almacenamiento de la ontología de patrón de análisis de reservación de entidades o el de cuentas, en la categoría definida acorde al nivel de la jerarquía.
5. Zona para almacenar individuos de cada patrón de análisis.	Representar la jerarquía propuesta.	Para constatar este nivel ingresar al menos una ontología con datos ficticios que corresponda a la estructura de la ontología del patrón de análisis almacenado en el apartado anterior.
6. Zona para almacenar las ontologías de casos específicos de los individuos de cada patrón de análisis.	Representar la jerarquía propuesta.	Para la justificación de este nivel observar la presencia de la categoría III-O Ontologías Casos, por el momento esta categoría estará vacía, debido a que no se cuenta con al menos una ontología real en la categoría anterior en consecuencia es complicado realizar una ontología con datos ficticios.
7. Consultar a través del nombre de una clase (palabra o concepto) de la ontología.	Realizar consultas al repositorio de ontologías.	Verificar el ingreso de un nombre de una clase de la ontología (palabra o concepto), proporcionando el nombre de las ontologías que cuentan una clase con el nombre ingresado.
8. Descarga de un archivo.	Descargar archivos	Revisar que la descargar del archivo de cierta ontología se realice y que cuente con el formato OWL.
9. Realizar una aportación.	Aportar ontologías por usuarios registrados	Probar a partir del identificador de la cuenta que se creó en el punto 1, representar una aportación de una ontología con datos ficticios.
10. Actualizar una ontología.	Actualizar información de cada ontología.	Evidenciar una actualización a la ontología con datos ficticios.
11. Realizar la eliminación de una ontología	Eliminar una ontología	Confirmar la eliminación de una ontología con datos ficticios.

2.3 Criterio de aceptación del caso de prueba

Las pruebas serán aceptadas si cumplen con cada una de la funcionalidades especificadas en los requerimientos funcionales sin ningún contratiempo.

2.4 Entregables de pruebas

Los entregables resultados de las actividades de pruebas son las siguientes:

1. Plan de Pruebas
2. Diseño de Pruebas
3. Casos de Pruebas
4. Procedimiento de Pruebas
5. Registro de Pruebas
6. Reporte de Pruebas

Anexo E. Procedimiento de pruebas

1. Introducción

En el presente documento se describen los pasos a ejecutar para examinar el conjunto de pruebas especificadas en el documento Casos de prueba.

1.1 Alcance

Describir el procedimiento de los casos de prueba: TC-01, TC-02, TC-03, TC-04, TC-05, TC-06, TC-07.

1.2 Referencias

IEEE Computer Society. (2008) 829-2008 – IEEE Standard for Software and System Test Documentation.

2. Procedimiento de prueba TPr-01

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-01.

2.1 Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo CentOS 6.6 de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*

- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

2.2 Descripción de los pasos a seguir para ejecutar el caso de pruebas TC-01

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador Google Chrome, colocar la dirección *ip* que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: En la interfaz del portal, hacer clic en la etiqueta *Sign in*, en la siguiente interfaz que se muestra hacer clic en el enlace *Sign UP*.

Proceso: El usuario debe ingresar los datos que solicita la interfaz de aplicación.

Medición: Depende de tiempo que el usuario se tome en ingresar los datos requeridos.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina con la creación de la cuenta del usuario.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

3. Procedimiento de prueba TPr-02

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-02.

3.1 Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas

- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo CentOS 6.6 de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

3.2 Descripción de los pasos a seguir para ejecutar el caso de pruebas TC-02

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador Google Chrome, colocar la dirección ip que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: Situarse en la pestaña *Browse* del portal y ubicar la sección con la leyenda *Category*, observar de debajo de *Category* las siete categorías definidas para adaptar la jerarquía de ontologías propuesta.

Proceso: Comprobar de manera visual la existencia de la ontología de la plantilla en la categoría I-O Ontología de la plantilla, la existencia de los individuos de la plantilla en la categoría I-I Individuos de la plantilla, la existencia de la ontología del patrón de análisis para reservación de entidades en la categoría II-O Ontología de PARE, la existencia de la ontología

del patrón de análisis de cuenta en la categoría II-O Ontología de PAC, la existencia de una ontología con datos ficticios en la categoría II-I Individuos de PARE, la existencia de una ontología con datos ficticios en la categoría II-I Individuos de PAC y la presencia de la categoría III-O Ontologías de Casos Específicos.

Medición: Depende del tiempo que el usuario se lleve en observar las categorías.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina en momento que se realiza de examinar las categorías.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

4. Procedimiento de prueba TPr-03

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-03.

4.1 Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo CentOS 6.6 de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*

- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

4.2 Descripción de los pasos a seguir para ejecutar el caso de prueba TC-03

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador Google Chrome, colocar la dirección ip que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: Ubicar la pestaña *Search* en el portal.

Proceso: En el cuadro de texto que se muestra en la interfaz de la pestaña *Search* ingresar el nombre de una clase de una ontología (palabra o concepto) y hacer clic en el botón *Search*, visualizando el listado de las ontologías que contienen la clase con el nombre ingresado.

Medición: La búsqueda no pasa de 10 segundos, se tiene que tomar en cuenta la conexión a internet.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina al finalizar la búsqueda.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

5. Procedimiento de prueba TPr-04

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-04.

5.1 Entradas, salidas y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo CentOS 6.6 de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

5.2 Descripción de los pasos a seguir para ejecutar el caso de prueba TC-04

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador Google Chrome, colocar la dirección ip que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: Elegir la ontología que se desea descargar en un archivo con formato owl.

Proceso: Hacer clic sobre el nombre de la ontología, esta acción nos conduce a otra interfaz, en esa interfaz ubicar la sección de *Submissions*, en dicha sección situarse debajo de la etiqueta *DOWNLOADS* y hacer clic sobre el enlace *OWL*.

Medición: De forma automática se descarga el archivo de la ontología.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina en el momento que se localiza el archivo en la carpeta aginada para la descarga del archivo.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

6. Procedimientos de prueba TP-05

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-05.

6.1 Entrada, salida y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo *CentOS 6.6* de 64-bit y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*

- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

6.2 Descripción de los pasos a seguir para ejecutar el caso de prueba TC-05

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador Google Chrome, colocar la dirección *ip* que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: Ingresar a la aplicación con el identificador de la cuenta de usuario, para esta prueba se emplea el identificador martha2017.

Proceso: En la aplicación, ubicarse en la pestaña *Browse* y hacer clic sobre el botón con la leyenda *Submit New Ontology*. Posterior a este paso se exhiben diferentes campos de texto, solicitando el nombre y acrónimo de la ontología, la categoría a la cual pertenecerá la ontología, estos datos son los más importantes en esta sección, a continuación hacer clic en el botón *Create ontology*. Se presenta una interfaz solicitando una descripción breve acerca de la ontología a ingresar, el formato del archivo en este caso elegir *OWL*, el estatus de la ontología, seleccionar la fecha de ingreso, el archivo *OWL* y finalmente el nombre y correo electrónico de la o las personas para ponerse en contacto otro usuario que necesite información detalla de la ontología. Inmediatamente del ingreso de todos los datos solicitados hacer clic en el botón *Add Submission*.

Medición: Este proceso toma pocos minutos.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina en el momento de hacer clic en el botón *Add Submission* y ver los datos correspondientes en el portal de la aplicación.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

7. Procedimientos de prueba TPr-06

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-06.

7.1 Entrada, salida y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo *CentOS 6.6* de *64-bit* y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de

análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

7.2 Descripción de los pasos a seguir para ejecutar el caso de prueba TC-06

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador *Google Chrome*, colocar la dirección ip que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario. Tener presente que el usuario puede actualizar la información de la ontología, si la aportación fue realiza por él, de no ser así la aplicación no muestra la opción para actualizar.

Inicio: Ingresar a la aplicación con el identificador de la cuenta de usuario, para esta prueba se emplea el identificador martha2017.

Proceso: Localizar el nombre de la ontología que se va a actualizar, hacer clic sobre el nombre de la ontología, en la interfaz que se muestra ubicar la etiqueta *Add submission*, posterior a este paso se observa una interfaz con los datos que tiene la ontología hasta este momento, se procede a actualizar los datos de la ontología y al terminar hacer clic en el botón *Add submission*.

Medición: Este proceso toma pocos minutos.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina en el momento de hacer clic en el botón *Add Submission* y ver los datos correspondientes en el portal de la aplicación.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.

8. Procedimientos de prueba TP-07

En esta sección se mencionan los datos que se requieren como entrada y el procedimiento para realizar el caso de prueba TC-07.

8.1 Entrada, salida y requerimientos especiales

Documentos:

- Plan de pruebas
- Diseño de pruebas

Aplicaciones:

- *Google Chrome*
- *VMware Workstation 12 Pro 12.1.0 build-3272444*

El servidor de la aplicación se ejecuta sobre el sistema operativo *CentOS 6.6* de *64-bit* y cuenta con los siguientes entornos:

- *Tomcat 6.0.26*
- *Solr 4.10.4*
- *Java 7*
- *MySQL 5.1.x*
- *Rails 3.2.x*
- *Ruby 2.1.5*
- *memcached*
- *redis 2.8.18*
- *nginx 1.6.3*
- *4store*
- *passenger/Apache*

Datos:

Archivo de la ontología de la plantilla estable, archivos de las ontologías de individuos de la plantilla, archivo de la ontología del patrón de análisis reservación de entidades y del patrón de análisis de cuentas, archivo de las ontologías de los individuos de la ontología del patrón de análisis de reservación de entidades y el de cuentas (contienen datos ficticios).

8.2 Descripción de los pasos a seguir para ejecutar el caso de prueba TC-07

Reporte: Su elaboración es manualmente observando los eventos que ocurren durante la ejecución de pruebas. El registro de los sucesos se realiza en el documento de pruebas.

Preparación: Verificar que el servidor de la aplicación este en ejecución, en el navegador *Google Chrome*, colocar la dirección *ip* que muestra el servidor después de ejecutar el comando correspondiente. Contar con los datos requeridos para la creación de la cuenta de usuario.

Inicio: En la consola del servidor ingresar a la carpeta */srv/ncbo/ncbo_cron*.

Proceso: Escribir el comando `./bin/ncbo_cron --console` para el ingreso a la consola de *ruby*, ubicándose en la consola de *ruby* escribir `ontology = LinkedData::Models::Ontology.find("RV")` y hacer clic en la tecla *Intro* del teclado del servidor, después ingresar el comando `ontology.delete`, a continuación presionar las teclas *Ctrl* + *d* para salir de la consola de *ruby* y finalmente reiniciar el servidor.

Medición: Depende del tiempo que el usuario se lleve en entrar a la consola y escribir los comandos necesarios.

Suspender: La operación se puede cancelar en el momento que el usuario lo desee, la interfaz de la aplicación no le afecta esta acción.

Reiniciar: La actividad se puede realizar en el momento que el usuario lo requiera.

Cierre: La prueba termina después de eliminar la ontología y reiniciar el servidor.

Contingencias: La presencia de anomalías, obliga a suspender la prueba hasta que se corrija el error. Solucionado el problema se puede reiniciar la prueba.