



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Sistema de datawarehouse de repositorios
de proyectos de software libre

presentada por
Ing. Rebeca Medina Jerónimo

como requisito para la obtención del
grado de

**Maestra en Ciencias de la
Computación**

Director de tesis
Dr. Moisés González García

Cuernavaca, Morelos, México. Mayo de 2020.



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., 15/mayo/2020

OFICIO No. DCC/035/2020

Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **C. Ing. Rebeca Medina Jerónimo**, con número de control M17CE034, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"Sistema de datawarehouse de repositorios de proyectos de software libre"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

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

Dr. René Santaolaya Salgado
Doctor en Ciencias de la Computación
4454821
Revisor 1

Dr. Joaquín Pérez Ortega
Doctor en Ciencias Computacionales
4795984
Revisor 2

C.c.p. Depto. Servicios Escolares
Expediente / Estudiante
JGGS/lmz

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 3202, e-mail: dcc@cenidet.edu.mx

www.tecnm.mx | www.cenidet.tecnm.mx



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos **26/mayo/2020**

OFICIO No. SAC/151/2020

Asunto: Autorización de impresión de tesis

REBECA MEDINA JERÓNIMO
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 *"Sistema de datawarehouse de repositorios de proyectos de software libre"*, 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

Excelencia en Educación Tecnológica
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



**CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA**

C.c.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares
Expediente
CVGR/CHC



Lo importante es no dejar de hacerse preguntas.

Albert Einstein

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo económico recibido durante estos años permitiéndome así realizar mis estudios de posgrado.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por darme la oportunidad de formar parte de esta institución y por el apoyo brindado en mi estancia.

A mi director de tesis Dr. Moisés González García, por sus consejos, ayuda, orientación y paciencia brindada durante la realización de este proyecto y mi estancia en el CENIDET.

A mis padres Daniel y Elizabeth, a quienes amo profundamente, les dedico esta tesis por haberme brindado su comprensión y apoyo incondicional durante toda mi maestría, hoy puedo ver alcanzada mi meta.

A las personas que de alguna u otra manera han sido parte de mi formación tanto personal como profesional.

¡Gracias!

CONTENIDO

	PAG.
LISTA DE FIGURAS	V
LISTA DE TABLAS	VI
ACRÓNIMOS.....	VIII
RESUMEN.....	IX
ABSTRACT	X
Introducción.....	1
1.1 Introducción.....	2
1.2 Organización del documento.....	2
Marco de Referencia.....	4
2.1 Descripción del problema.....	5
2.2 Objetivos	5
2.2.1 Objetivo general.....	5
2.2.2 Objetivos específicos	6
2.3 Propuesta de solución	6
2.4 Alcances y limitaciones	7
2.4.1 Alcances	7
2.4.2 Limitaciones	7
2.5 Antecedentes.....	7
2.6 Trabajos relacionados	9
Marco Conceptual.....	13
3.1 Almacén de datos.....	14
3.2 Ontología.....	16
3.2.1 Componentes de una ontología.....	16
3.3 Extracción, Transformación y Carga (ETL)	16
3.4 Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R).....	17
3.5 Algoritmo de Levenshtein.....	18
3.6 Núcleo de Essence	19
3.7 Área de Minado de Repositorios de Software.....	20

3.8 Comparación de Factores MCPS-R.....	26
3.8.1 Metodologías Tradicionales	27
3.8.2 Metodologías Ágiles	30
3.8.3 Metodologías Híbridas	33
Diseño del almacén de datos	38
4.1 Repositorios FLOSSmole.....	39
4.2 Diseño conceptual del almacén de datos.....	41
4.2.1 Tarea1: Construir un glosario de términos	41
4.2.2 Tarea2: Construir una taxonomía de conceptos.....	55
4.2.3 Tarea3: Construir diagramas de relaciones binarias ad hoc	56
4.2.4 Tarea4: Construir el diccionario de conceptos	61
4.2.5 Tarea5: Describir en detalle las relaciones binarias.....	66
4.2.6 Formalización	67
4.3 Diseño lógico del almacén de datos.....	69
4.3.1 factors.....	70
4.3.2 alphas.....	71
4.3.3 milestones	71
4.3.4 mcpsr	71
4.3.5 competencias	72
4.3.6 mcpsr_competency	72
4.3.7 repositories	73
4.3.8 tables	73
4.3.9 columns_mcpsr	73
4.3.10 querys_metadata	74
4.3.11 operationals_metadata.....	74
4.3.12 state_alpha.....	75
4.3.13 Relaciones	75
4.4 Diseño físico del almacén de datos.....	75
Requerimientos y Diseño del sistema	85
5.1 Especificación de requerimientos	86

5.1.1 Notación de requerimientos de usuario	86
5.1.2 Perspectiva del producto	88
5.1.3 Funciones del producto	88
5.1.4 Características del usuario	88
5.1.5 Especificación URN.....	89
5.1.5.1 Diagrama GRL	89
5.1.5.2 Diagramas UCM.....	89
5.1.6 Requerimientos obtenidos	91
5.1.6.1 Requerimientos funcionales	92
5.1.6.2 Requerimientos no funcionales	92
5.2 Diseño del sistema	92
5.2.1 Diagramas caso de uso	92
5.2.2 Diagramas de secuencia	98
5.2.3 Diagrama de clases	101
Implementación.....	104
6.1 Cargar repositorios FLOSSmole hacia el dw_manager y Análisis de similitudes ...	105
6.2 Guardar metadatos	107
6.3 Consultar el dw_manager.....	108
6.4 Respalidar el dw_manager	110
6.5 Interfaz de usuario	110
Pruebas y Resultados	116
7.1 Plan de pruebas	117
7.1.1 Elementos de prueba	117
7.1.2 Requerimientos a probar.....	118
7.1.3 Matriz de Trazabilidad de pruebas	118
7.1.4 Criterio de aceptación de pruebas	119
7.1.5 Criterio de no aceptación de pruebas	119
7.1.6 Características del ambiente de pruebas.....	119
7.2 Diseño de Pruebas.....	119
7.2.1 Características a probar	119

7.2.2 Enfoque de pruebas	120
7.3 Casos de Prueba.....	120
7.3.1 Especificaciones de caso de prueba TC-01	120
7.4 Procedimiento de Pruebas	122
7.4.1 Procedimiento de prueba TPr-01	122
7.5 Registro de Pruebas	124
7.5.1 Registro de prueba TL-01.....	124
7.6 Reporte de Pruebas	128
7.6.1 Resultados generales	128
7.6.2 Resultados detallados	128
Conclusiones y Trabajos futuros	129
8.1 Conclusiones.....	130
8.2 Aportaciones.....	130
8.3 Trabajos Futuros	131
Referencias	132
Anexo A Atributos por factor del MCPS-R	136
Anexo B Estructuras de repositorios FLOSSmole a utilizar	142
Anexo C Metodología Methontology	147
Anexo D Diseño de base de datos.....	152
Anexo E Diccionario de datos MCPSR	156
Anexo F Documentación de pruebas	160

LISTA DE FIGURAS

Figura 2.1 Diseño de la propuesta de solución	6
Figura 3.1 Arquitectura típica de un almacén de datos.....	15
Figura 4.1 Taxonomía de la ontología MCPS-R.....	56
Figura 4.2 Relaciones binarias ad hoc de alfa oportunidad.	57
Figura 4.3 Relaciones binarias ad hoc de alfa requerimientos.....	57
Figura 4.4 Relaciones binarias ad hoc de alfa sistema de software.....	57
Figura 4.5 Relaciones binarias ad hoc de alfa interesados.....	58
Figura 4.6 Relaciones binarias ad hoc de alfa equipo.....	58
Figura 4.7 Relaciones binarias ad hoc de alfa forma de trabajar.....	58
Figura 4.8 Relaciones binarias ad hoc de alfa trabajo.	59
Figura 4.9 Relaciones binarias ad hoc de competencias.....	59
Figura 4.10 Relaciones binarias ad hoc de factores configurables.....	59
Figura 4.11 Relaciones binarias ad hoc de factores medibles en proyectos de software.....	60
Figura 4.12 Relaciones binarias ad hoc de factores sociológicos.....	60
Figura 4.13 Relaciones binarias ad hoc de factores del tipo de proyecto de software.	60
Figura 4.14 Relaciones binarias ad hoc de factores tecnológicos.	61
Figura 4.15 Relaciones binarias ad hoc de Milestone.	61
Figura 4.16 Ontología MCPS-R creada en Protegé.....	68
Figura 4.17 Taxonomía de la ontología MCPS-R.....	68
Figura 4.18 Esquema del modelo lógico dw_manager.....	70
Figura 4.19 Ejecución sin errores del script y estructura final de base de datos dw_manager .	84
Figura 5.1 Construcciones visuales de GRL herramienta JUCMNav v7.0.0.....	87
Figura 5.2 Construcciones visuales de UCM herramienta JUCMNav v7.0.0.....	87
Figura 5.3 GRL y UCM como una pieza faltante del rompecabezas UML.....	88
Figura 5.4 Diagrama GRL-Sistema administrador de almacén de datos.....	89
Figura 5.5 Diagrama UCM-Cargar repositorios FLOSSmole hacia el dw_manager.	90
Figura 5.6 Diagrama UCM-Guardar metadatos	90
Figura 5.7 Diagrama UCM-Respaldar el dw_manager.....	91
Figura 5.8 Diagrama UCM-Consultar el dw_manager.....	91
Figura 5.9 Diagrama de caso de uso CU00 Utilizar ADE-DS.	93
Figura 5.10 Diagrama de casos de uso del administrador.....	93
Figura 5.11 Diagrama de casos de uso del sistema ADE-DS.	94
Figura 5.12 Diagrama de secuencia Carga repositorios FLOSSmole hacia el dw_manager. ...	98
Figura 5.13 Diagrama de secuencia Análisis de similitudes con el MCPS-R.....	99
Figura 5.14 Diagrama de secuencia Consultar el dw_manager.	99
Figura 5.15 Diagrama de secuencia Respaldar el dw_manager.	100
Figura 5.16 Diagrama de secuencia Guardar metadatos.....	100

Figura C.1 Ejemplo de relaciones en un dominio.....	148
Figura C.2 Ejemplo de instancias	148
Figura C.3 Ejemplo de atributos de instancia.....	148
Figura C.4 Ejemplo de atributos de clase.....	148
Figura C.5 Modelado conceptual de una ontología	149

LISTA DE TABLAS

Tabla 2.1. Trabajos relacionados.	11
Tabla 3.1. Factores del MCPS-R.	17
Tabla 3.2. Ejemplo de atributos del factor configurable en proyectos de software (ocho atributos).....	18
Tabla 4.1. Selección de repositorios FLOSSmole	40
Tabla 4.2. Glosario de términos.....	41
Tabla 4.3. Diccionario de conceptos.....	61
Tabla 4.4. Relaciones binarias Ontología MCPS-R	67
Tabla 4.5. Descripción de la tabla “factors” de la base de datos dw_manager.	70
Tabla 4.6. Descripción de la tabla “alphas” de la base de datos dw_manager.....	71
Tabla 4.7. Descripción de la tabla “milestones” de la base de datos dw_manager.	71
Tabla 4.8. Descripción de la tabla “mcpsr” de la base de datos dw_manager.	71
Tabla 4.9. Descripción de la tabla “competencies” de la base de datos dw_manager.....	72
Tabla 4.10. Descripción de la tabla “mcpsr_competency” de la base de datos dw_manager..	72
Tabla 4.11. Descripción de la tabla “repositories” de la base de datos dw_manager.....	73
Tabla 4.12. Descripción de la tabla “tables” de la base de datos dw_manager.....	73
Tabla 4.13. Descripción de la tabla “columns_mcpsr” de la base de datos dw_manager.	73
Tabla 4.14. Descripción de la tabla “querys_metadata” de la base de datos dw_manager.....	74
Tabla 4.15. Descripción de la tabla “operationals_metadata” de la base de datos dw_manager.	74
Tabla 4.16. Descripción de la tabla “state_alpha” de la base de datos dw_manager.	75
Tabla 4.17. Descripción de relaciones entre las tablas de la base de datos dw_manager.	75
Tabla 4.18. Script SQL obtenido al transformar el modelo de la Figura 4.18.	76
Tabla 5.1. CU01 Cargar repositorios FLOSSmole hacia el dw_manager.	94
Tabla 5.2. CU02 Análisis de similitudes con el MPCPS-R.	95
Tabla 5.3. CU03 Consultar el dw_manager.	95
Tabla 5.4. CU04 Respalda el dw_manager.....	96
Tabla 5.5. CU05 Guardar metadatos.....	97
Tabla 5.6. CU05 Guardar metadatos.....	97
Tabla 7.1. Matriz de trazabilidad de pruebas.	118
Tabla 7.2. Características a probar.....	119

Tabla A.1. Factor tipo de proyecto de software	136
Tabla A.2. Factor atributos medibles	137
Tabla A.3. Factor atributos sociológicos.....	138
Tabla A.4. Factor atributos tecnológicos	139
Tabla A.5. Factor atributos configurables	139
Tabla B.1. Estructura del Repositorio Alioth.	142
Tabla B.2. Estructura del Repositorio RubyForge.....	144
Tabla B.3. Estructura del Repositorio Tigris.....	145

ACRÓNIMOS

AGD: *Architectural and Group Development*, en español Método de Desarrollo Arquitectónico en Grupo

ADE-DS: Almacén de Datos Esenciales del Desarrollo de Software

CSV: *Comma-Separated Values*, en español Valores Separados por Comas

CENIDET: Centro Nacional de Investigación y Desarrollo Tecnológico

FLOSS: *Free/Libre Open Source Software*, en español Software Libre y de Código abierto

GRL: *Goal-oriented Requirements Language*, en español Lenguaje de Requerimientos Orientado a Objetivos

IEEE: *Institute of Electrical and Electronics Engineers*, en español Insitituto de Ingenieros Eléctricos y Electrónicos.

ITU: *International Telecommunication Union*, en español Unión Internacional de Telecomunicaciones

MCPS-R: Marco para la Caracterización de Proyectos de Software Refinado

MSR: *Mining Software Repositories*.

UCM: *Use Case Maps*, en español Mapas de Casos de Uso

UML: *Unified Modeling Language*, en español Lenguaje de Modelado Unificado

URN: *User Requirements Notation*, en español Notación de Requerimientos de Usuario

RESUMEN

En los proyectos de desarrollo de software propietario, se obtiene gran cantidad de información durante sus diversos ciclos de vida, desde la obtención de requerimientos hasta el mantenimiento de software. Para dirigir la producción de esta documentación, destinada para los desarrolladores, los usuarios y los administradores de proyectos, existen diversos estándares para documentación del desarrollo de software, y existen herramientas que ayudan a obtener los productos importantes resultantes de las diversas actividades. Además, en los últimos años se ha presenciado un incremento de organizaciones que promueven el desarrollo de proyectos *Free Libre Open Source Software*. Esto ocasiona que se genere una gran cantidad de datos de proyectos de software. Sin embargo, contrario a lo que pudiera creerse, esta explosión de datos no está significando un aumento en el conocimiento.

Las organizaciones con proyectos de software abierto/libre, comparten los datos de sus proyectos, colocándolos en varios repositorios de software de acceso libre (p.ej. FLOSSmole), para ponerlos a disposición de la comunidad de usuarios. De esta información una vez limpia, se obtiene una versión operacional en formato valores separados por comas (CSV), que se requieren cargar a un almacén de datos (*data warehouse*) y administrarlos.

En nuestro grupo de investigación, con anterioridad, se propuso un Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R), formado por 90 características agrupadas en cinco factores y se estudió la relación existente entre dichas características y el núcleo de Essence, compuesto por alfas y competencias. Además, se desarrolló la herramienta DB2O-Cleaner que prepara (selecciona y limpia) los datos de FLOSSmole, hasta representarlos en formato CSV.

En esta investigación se especifican los requerimientos, se diseña el almacén de datos y el software, y se construye la herramienta Almacén de Datos Esenciales del Desarrollo de Software (ADE-DS). Que permite cargar un almacén de datos, características de proyectos de desarrollo de software, provenientes del repositorio FLOSSmole. El diseño del esquema del almacén se basó en: 1) el MCPS-R, y 2) las alfas y competencias del núcleo de Essence. Utilizando el manejador de base de datos MySQL para la creación del almacén de datos y para la implementación de la herramienta ADE-DS el IDE Netbeans.

Adicionalmente, al ADE-DS se le incluyó la funcionalidad para gestionar metadatos de operación y de consulta, permitir el respaldo de los datos del almacén, y presentar al usuario conocimiento sobre los repositorios y sus datos, mediante un análisis de similitudes con el MCPS-R.

ABSTRACT

In proprietary software development projects, a large amount of information is obtained during its various life cycles, from obtaining requirements to maintaining software. To direct the production of this documentation, intended for developers, users and project managers, there are various standards for software development documentation, and there are tools that are helpful in predicting the important products resulting from the various activities. In addition, in recent years there has been an increase in organizations that promote the development of Free Libre Open Source Software projects. This causes a large amount of data to be generated from software projects. However, contrary to what might be believed, this explosion of data does not mean an increase in knowledge.

Organizations with open / free software projects share the data of their projects, placing them in several open-access software repositories (eg FLOSSmole), to make them available to the user community. Once this information is clean, an operational version in comma separated values (CSV) format is obtained, which is required to be loaded into a data warehouse and managed.

In our research group, previously, the Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R) was proposed, consisting of 90 characteristics grouped into five factors and the relationship among these characteristics and the core of Essence was studied, composed of alphas and skills. In addition, the DB2O-Cleaner tool, which prepares (selects and cleans) the FLOSSmole data until it is represented in CSV format was developed.

In this investigation the requirements are specified, the data warehouse and the software are designed, and the Essential Data Warehouse Software Development Tool (ADE-DS) is built. This enables loading features of software development projects, coming from the FLOSSmole repository into a data warehouse. The design of the warehouse scheme was based on: 1) the MCPS-R, and 2) the alphas and competences of the Essence core. Using the MySQL database handler for the creation of the data warehouse and for the implementation of the ADE-DS tool the Netbeans IDE.

Additionally, the ADE-DS was included with the functionality to manage operation and query metadata, allow the backup of the warehouse data, and present the user with knowledge about the repositories and their data, through an analysis of similarities with the MCPS- R.

Capítulo 1

Introducción

En este capítulo se presenta la introducción y una descripción breve de la organización de la tesis.

1.1 Introducción

En los últimos años ha habido un incremento explosivo en lo referente al desarrollo de software, gracias a ello, grandes cantidades de datos del producto y del proceso de software se encuentran almacenadas en diferentes repositorios de datos de software(Gonzalez-Barahona, Izquierdo-Cortazar, & Squire, 2010). Esta gran cantidad de datos a su vez supone un problema cuando se desean consultar, usarlos como soporte en la toma de decisiones o bien analizarlos en un ambiente de minería de datos (de F. Farias et al., 2016).

A lo largo de las últimas décadas, el reto para las áreas de Tecnología de Información (TI), ha sido desde el almacenamiento de grandes volúmenes de datos y su procesamiento masivo(Gonzalez-Barahona, Izquierdo-Cortazar, & Squire, 2010). Una solución a este problema, es integrar estos datos de los repositorios en un almacén de datos de software. Sin embargo, cada repositorio se encuentra estructurado de forma diferente y para lograr este objetivo es necesario el diseño de un almacén de datos el cual emplea la representación de un modelo de datos común (Curto Díaz & Conesa Caralt, 2012).

1.2 Organización del documento

Este documento se organiza de la siguiente manera:

Capítulo 2. Marco de referencia. En este capítulo se presenta el planteamiento del problema que dio origen a esta investigación; el objetivo de la tesis; los alcances y limitaciones; la solución propuesta; así como los antecedentes; y trabajos relacionados a esta investigación.

Capítulo 3. Marco conceptual. En este capítulo se describen los conceptos a considerar para la identificación y síntesis de la solución del problema y el desarrollo de la investigación. Con el fin de facilitar el entendimiento de esta investigación.

Capítulo 4. Diseño del almacén de datos. En este capítulo se especifican los requerimientos y el diseño del almacén de datos (dw_manager) del sistema ADE-DS.

Capítulo 5. Requerimientos y Diseño del sistema. En este capítulo se especifican los requerimientos y el diseño para desarrollar la herramienta ADE-DS.

Capítulo 6. Implementación. En este capítulo se presentan las consideraciones para construir la herramienta ADE-DS y se describen sus funcionalidades.

Capítulo 7. Pruebas y Resultados. En este capítulo se describe la planificación, especificación, y ejecución de las pruebas y los resultados obtenidos.

Capítulo 8. Conclusiones y trabajos futuros. En este capítulo se incluyen las conclusiones a las que se llegaron después de haber realizado este trabajo, las aportaciones y los trabajos futuros que permitan la continuación de esta investigación.

Capítulo 2

Marco de Referencia

En este capítulo se presenta la descripción del problema que motivó a esta investigación; la solución propuesta para resolverlo; el objetivo a alcanzar. Además, se incluyen aquí los antecedentes y trabajos relacionados, que sirvieron de base para realizar la investigación.

A continuación, se presenta la estructura del capítulo:

- 2.1 Descripción del problema
- 2.2 Objetivos
 - 2.2.1 Objetivo general
 - 2.2.2 Objetivos específicos
- 2.3 Propuesta de solución
- 2.4 Alcances y limitaciones
 - 2.4.1 Alcances
 - 2.4.2 Limitaciones
- 2.5 Antecedentes
- 2.6 Trabajos relacionados

2.1 Descripción del problema

En CENIDET, se tienen datos de repositorios de software (FLOSSmole), que contienen datos heterogéneos acerca del desarrollo de software, extraídos y preparados para que los datos se carguen en un almacén. Estos datos se encuentran en un formato de valores separados por coma (CSV, *Comma-Separated Values*), contenidos en carpetas separadas para cada repositorio. Y no se ha realizado la carga de estos datos en un almacén, que permita la administración de la información.

El análisis de la situación anterior nos lleva a plantear el problema de que los datos de estos repositorios no están almacenados en una estructura de datos común que facilite su utilización para la toma de decisiones. Además, falta funcionalidad que asegure la consistencia de los datos y el registro de las operaciones realizadas con ellos.

2.2 Objetivos

2.2.1 Objetivo general

Desarrollar un sistema administrador de almacén de datos, que permita cargar y administrar los datos de proyectos de desarrollo de software FLOSSmole.

2.2.2 Objetivos específicos

- Diseñar el almacén de datos tomando como esquema el Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R) y la relación existente entre las características del MCPS-R y el núcleo de Essence, compuesto por alfas y competencias.
- Definir la estructura para los metadatos del almacén de datos
- Definir requerimientos, diseñar y construir un sistema administrador de un almacén de datos.

2.3 Propuesta de solución

Para resolver el problema planteado en la sección anterior, se propone el desarrollo de una herramienta que permita la carga e integración a partir de los archivos CSV de repositorios FLOSSmole.

En la Figura 2.1 se aprecia la solución propuesta. El primer paso, consiste en cargar en un almacén de datos los archivos .csv de repositorios FLOSSmole esto nos permitirá realizar el análisis de similitudes con el Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R) utilizando el algoritmo de Levenshtein. En el segundo paso, poder realizar consultas administrativas de los repositorios FLOSSmole cargados. El tercer paso se obtienen los metadatos operacionales y de consulta para poder gestionarlos. En el cuarto y último paso, se crea un respaldo total del almacén de datos.

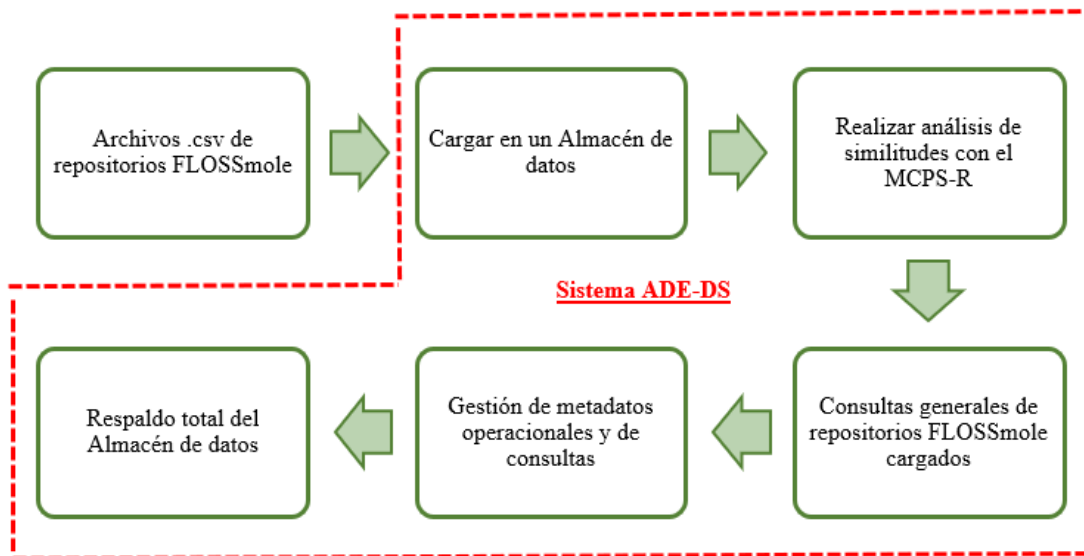


Figura 2.1 Diseño de la propuesta de solución

Fuente: creación propia

2.4 Alcances y limitaciones

2.4.1 Alcances

- Utilizar como esquema base para el almacén de datos la ontología MCPS-R (Cancino, 2020). Que incluye la relación existente entre las características del MCPS-R y el núcleo de Essence, compuesto por alfas y competencias.
- Se considerará como usuario del diseño del almacén a el sistema administrador de consultas realizadas por expertos en desarrollo de software.

2.4.2 Limitaciones

- Sólo se usarán repositorios de FLOSSmole ya pre-procesados.
- Se realizará solamente el diseño del almacén de datos para proyectos de desarrollo de software.
- Para la experimentación se establecerá un número máximo de 3 repositorios de proyectos de desarrollo de software FLOSSmole.

2.5 Antecedentes

A continuación, se presentan las partes relacionadas con esta investigación de los trabajos realizados por profesores y estudiantes del CENIDET.

- Tesis de maestría “**Caracterización de Proyectos de Software para Configurar su Desarrollo y Habilitar la Comparación entre Casos Almacenados en la Memoria Organizacional**” (Delgado Solis, 2008). En este trabajo se creó el Marco para la Caracterización de Proyectos de Software (MCPS), que fue el paso inicial para la implementación de la herramienta *Architectural and Group Development Environment* (AGDE). En el diseño del marco de características, se integraron 44 atributos relacionados con el desarrollo de un proyecto de software (el MCPS), que habilita diseñar una base de datos para el almacenamiento de la información de proyectos desarrollados. El marco MCPS, se revisó y modificó para obtener la versión refinada denominada MCPS-R, en (Sánchez-Santamaria, 2010).
- Tesis de maestría “**Evaluación de Técnicas de Comparación de Diferentes Grupos de Características de Proyectos de Software**” (Sánchez-Santamaria, 2010). En este trabajo se ampliaron los 44 atributos del MCPS de (Delgado-Solis, 2008), quedando 98 atributos que representan las características necesarias que tiene todo proyecto de software. Se evaluaron diferentes técnicas de comparación de proyectos de software y sus metadatos, para poder determinar cuáles son las más confiables en la obtención de similitud entre proyectos.

- Tesis de maestría “**Componentes de acoplamiento para la infraestructura de Soporte a Minería de Datos de Desarrollo de Software**” (Pérez Luna, 2016). En este trabajo se desarrolló el Ambiente para Minería de Datos de Desarrollo de Software (AMDADS), que está integrado por diferentes herramientas que soportan las necesidades de algunas de las etapas del proceso de minería de datos. Pero que se enfoca principalmente en la fase de preparación de datos. Esta investigación surgió, debido a que no existía una herramienta que integrara diferentes aplicaciones en un ambiente que soporte el proceso de preparación de datos, y que reduzca el tiempo y esfuerzo invertido.
- Tesis de maestría “**Preparación de Repositorios de Software usando Ontologías**” (Aguilar Viveros, 2016). En este trabajo se propuso la creación de un método que abarque la fase de preparación de datos, mediante el uso de ontologías, que se crean en base al repositorio a preparar. Las ontologías de los repositorios procesados hacen posible que su semántica se relacione con la ontología que representa la semántica del almacén destino de datos de software. Para soportar el método propuesto, se desarrolló la herramienta DB2O-Cleaner (*Database to Ontology-Cleaner*) que permite la limpieza de datos disponibles en repositorios de software utilizando ontologías que describan su estructura, y produce los datos preparados de los repositorios en archivos CSV.
- Tesis de maestría “**Priorización de características del desarrollo de software de acuerdo a la experiencia de los equipos de desarrollo de software orientado a la mejora de consultas a un almacén de datos históricos**” (Cancino Fuentes, 2020). En este trabajo se propuso desarrollar una metodología para representar de manera explícita el conocimiento de las capacidades en el desarrollo de software, para mejorar las consultas a datos históricos. Para lograrlo se agrega el Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R), las alfas y competencias del núcleo de Essence. Como base para establecer la importancia de las características del proyecto de desarrollo de software, mediante mapas cognitivos difusos, la experiencia de un grupo de profesionales.

De los cinco trabajos anteriormente mencionados, los de mayor utilidad para esta investigación, son cuatro. El primero es el realizado por (Delgado-Solis, 2008), el cual desarrolla el MCPS, que agrupa las características que todo proyecto de software tiene; El segundo es el realizado por (Sánchez-Santamaria, 2010), que refina el trabajo anterior y produce el MCPS-R, El tercero trabajo es de (Cancino Fuentes, 2020), que se utiliza para realizar la estructura del almacén de datos abarcando las alfas y competencias del núcleo de Essence en la ontología global MCPS-R; El cuarto es el realizado por (Aguilar Viveros, 2016), que realiza la preparación de los datos de proyectos de desarrollo de software FLOSSmole para utilizar los archivos .csv como entrada del sistema para el almacén de datos.

2.6 Trabajos relacionados

A continuación, se presentan los trabajos de investigación relacionados que sirvieron como base de esta investigación.

- Artículo **“Toward an ontology based approach for data warehousing”** (Elamin & Feki, 2014). En este artículo se presentó un nuevo enfoque para el diseño de un almacén de datos basado en una ontología, esto para disminuir los problemas semánticos (p.ej. terminología y semántica poco frecuentes). El diseño de un almacén de datos se considera una tarea compleja y tediosa, todavía no hay un enfoque común que los diseñadores puedan seguir para diseñar un almacén de datos. Tomando en cuenta que el uso de una ontología para diseñar un almacén ayuda a resolver problemas de heterogeneidad semántica entre las fuentes de datos y los requisitos del usuario.
- Artículo **“Generating Data Warehouse Schema”** (Abdalazis Ahmedl & Mohamed Ahmedl, 2014). En este artículo se presentó los componentes principales de un almacén de datos (sistemas operativos de origen, almacén de datos operacionales, área de preparación de datos, presentación de datos y herramientas de acceso a datos) y los principales pasos para construir un almacén de datos (recopilación de datos, transformación y limpieza, agregación y análisis, muestra de resultados para los usuarios finales en forma de informes). El método propuesto presenta una nueva estrategia para automatizar el diseño multidimensional de data warehouses; este algoritmo utiliza un esquema empresarial de una base de datos operativa como punto de partida para diseñar el esquema del depósito de datos.
- Artículo **“Research on Extract, Transform and Load(ETL) in Land and Resources Star Schema Data Warehouse”** (Qin, Jin, & Zhang, 2012). En este artículo se presentó las tecnologías clave de ETL incluyendo la extracción, transformación y carga incremental de datos. Incluye la extracción de los datos requeridos de la fuente de datos, el procesamiento y la carga de los datos en el almacén de datos de acuerdo con el modelo predefinido en el almacén. Asignando los atributos de las fuentes de datos a los atributos de las tablas del almacén de datos.
- Artículo **“Testing Extract-Transform-Load Process in Data Warehouse Systems”** (Homayouni, 2018). Este artículo presenta los procesos ETL en un sistema de almacén de datos con el fin de integrar los datos para que se ajusten al formato del modelo de un almacén. Evaluando dos tipos de enfoques de pruebas funcionales, a saber, la calidad de los datos y las pruebas de equilibrio. Las pruebas de calidad de datos validan los datos

en el almacén de datos objetivo de forma aislada y las pruebas de equilibrio verifican las discrepancias entre los datos de origen y destino.

- Artículo **“Datawarehouse: A Data Warehouse artista who have ability to understand data warehouse Schema pictures”**(Warnars & Randriatoamana, 2017). En este artículo se enumeró los requisitos de conocimientos básicos para convertirse en un artista de almacén de datos, entre los cuales menciona conocimientos básicos del software, bases de datos y data warehouse. Conocimientos para reconocer cada uno de los diseños de modelos de bases de datos y el diseño transaccional. Se menciona en que consisten el esquema de almacén de datos, los procesos ETL y la metodología del almacén de datos.
- Artículo **“Significance of Data Integration and ETL in Business Intelligence Framework for Higher Education”** (Rodzi, Othman, & Yusuf, 2016). En este artículo se utilizó el caso de la *Universiti Teknologi Malaysia (UTM)* como estudio de caso para comprender mejor la relación de la integración de datos con el proceso de extracción, transformación y carga (ETL). En el marco de inteligencia empresarial y su importancia para una mejor gestión de la educación superior. Combinando datos de diferentes fuentes de datos académicos e integrándolos antes de almacenarlos, para que sean compatibles con el sistema de destino específico.
- Artículo **“The challenges of Extract, Transform and Loading (ETL) system implementation for near real-time environment”** (Sabtu et al., 2017). En este artículo se identificó los desafíos de la implementación del sistema ETL para transmisión de datos en tiempo real, que deben evolucionar con los diferentes requisitos. La clasificación de los desafíos del sistema ETL se prepara en función de las características del entorno en tiempo real, que son: alta disponibilidad, baja latencia y escalabilidad horizontal, así como las etapas ETL, para fomentar diferentes perspectivas.

2.6.1 Comparación de trabajos relacionados

En la Tabla 2.1, se presenta de forma general el objetivo por el cual se llevaron a cabo las investigaciones en los trabajos relacionados, el método utilizado, el resultado para resolver dichos problemas; así como, el aporte para nuestra investigación.

Tabla 2.1. Trabajos relacionados.

Fuente: creación propia.

Investigación	Objetivo	Método	Resultado	Aporte
(Elamin & Feki, 2014)	Diseño de un almacén de datos.	Basado en una ontología.	Resuelve problemas de heterogeneidad semántica entre las fuentes de datos y requisitos de usuario.	Forma de definir la estructura del almacén de datos en base a una ontología.
(Abdalazis Ahmedl & Mohamed Ahmedl, 2014)	Construcción de un almacén de datos conociendo los conocimientos principales.	Utiliza un esquema empresarial de una base de datos operativa.	Automatiza el diseño multidimensional de un almacén de datos.	Conocer los componentes principales de un almacén de datos.
(Qin, Jin, & Zhang, 2012)	Proceso ETL extracción, transformación y carga.	Extracción de los datos de la fuente, procesamiento y carga con modelo predefinido del almacén.	Asigna los atributos de las fuentes de datos a los atributos de las tablas del almacén de datos.	Comprender proceso ETL.
(Homayouni, 2018)	Proceso ETL extracción, transformación y carga.	Integración de datos ajustándose al formato del modelo de un almacén.	Evaluación de dos tipos de enfoques pruebas funcionales (calidad de datos) y pruebas de equilibrio (verifican discrepancias entre datos de origen y destino).	Formas de realizar y probar la integración de datos en el almacén mediante el proceso ETL.
(Warnars & Randriatoamanana, 2017)	Conocimientos necesarios para ser un artista de almacén de datos.	Ninguno.	Requisitos de conocimientos básicos del software, bases de datos y almacén.	Detalles de un esquema de almacén de datos y de los procesos ETL.
(Rodzi, Othman, & Yusuf, 2016)	Importancia de la integración de datos y del ETL.	Utiliza <i>Universiti Teknologi Malaysia</i> como caso de estudio.	Combina datos de diferentes fuentes de datos académicos y los integra para que sean compatibles con el sistema destino específico.	Manera de combinar los datos para que sean compatibles con el almacén.
(Sabtu et al., 2017)	Identificar los desafíos de la implementación del sistema ETL para transmisión	Ninguno.	Clasificación de los desafíos ETL en diferentes etapas para fomentar	Conocimiento de los desafíos en el proceso ETL.

	de datos en tiempo real.		diferentes perspectivas.	
--	--------------------------	--	--------------------------	--

Los antecedentes y trabajos relacionados aclararon el panorama de esta investigación, proporcionando ideas y delimitando el alcance de la investigación. En este capítulo se incluyeron investigaciones que aportaron conocimiento acerca de conceptos y maneras de realizar esta investigación. En el siguiente capítulo se definen los conceptos relacionados al tema de investigación con el fin de tener una mejor comprensión del tema.

Capítulo 3

Marco Conceptual

En este capítulo se explican los conceptos y términos de las áreas relacionadas con esta investigación, tales como almacén de datos, ontología, ETL, MCPS-R entre otros. Esta información permite comprender la terminología utilizada en el desarrollo de la investigación, que se describe en los capítulos siguientes.

A continuación, se presenta la estructura del capítulo:

- 3.1 Almacén de datos
- 3.2 Ontología
 - 3.2.1 Componentes de una ontología
- 3.3 Extracción, Transformación y Carga (ETL)
- 3.4 Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R)
- 3.5 Algoritmo de Levenshtein
- 3.6 Núcleo de Essence
- 3.7 Área de Minado de Repositorios de Software
- 3.8 Comparación de Factores MCPS-R
 - 3.8.1 Metodologías Tradicionales
 - 3.8.2 Metodologías Ágiles
 - 3.8.3 Metodologías Híbridas

3.1 Almacén de datos

Un almacén de datos provee la facilidad de extraer la información que se encuentra en varios manejadores de bases de datos sin tener información duplicada. Unifica la forma de realizar las consultas de forma que todos los usuarios finales tengan acceso a la información de la forma más simple, permitiendo la toma de decisiones en tiempo y forma (Kimball & Caserta, 2004).

En la definición de Bill Inmon, un almacén de datos es una colección de datos orientada a temas específicos, integrada, variable en el tiempo y no volátil (Connolly & Begg, 2005). La estructura básica de un almacén de datos (véase la Figura 3.1), incluye:

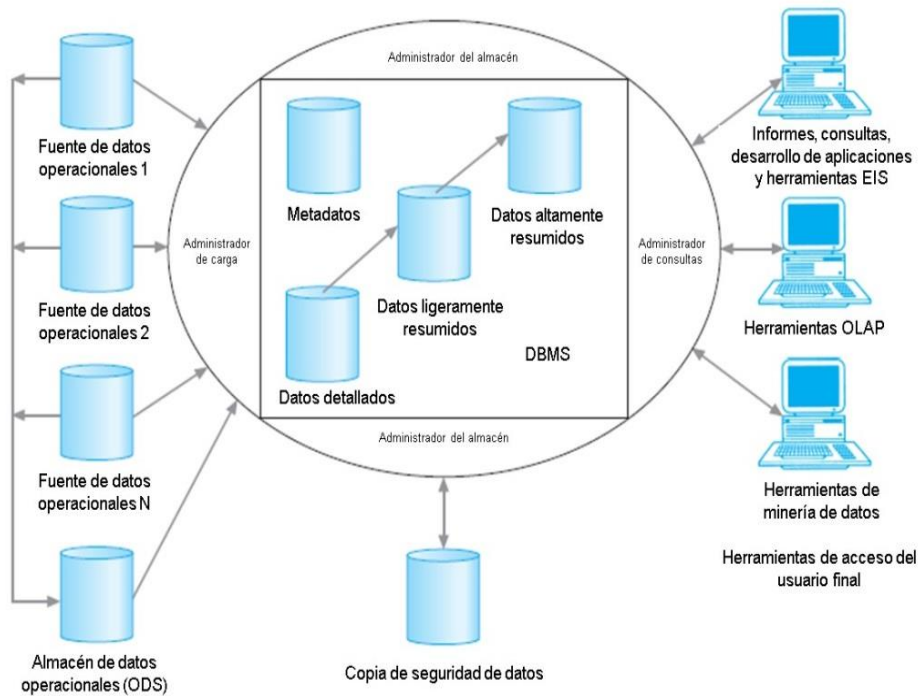


Figura 3.1 Arquitectura típica de un almacén de datos.

Fuente: traducida de (Connolly & Begg, 2005)

1. Datos operacionales. Origen de datos para el componente de almacenamiento físico del almacén de datos.
2. Almacén. Compuesto por metadatos que describen la estructura de los datos contenidos en el almacén, datos detallados actuales obtenidos directamente del procesado de los datos, datos ligeramente resumidos que es el primer nivel de agregación de los datos detallados actuales, datos muy resumidos son el nivel más alto de agregación y una copia de seguridad de los datos.
3. Herramientas de acceso. Son las herramientas que proveen acceso a los datos.
4. Administrador de carga. Realiza todas las operaciones asociadas con la extracción y carga de datos en el almacén. Los datos pueden ser extraídos directamente de la fuente de datos o desde el almacén operacional.
5. Administrador del almacén. Realiza operaciones como el análisis de datos para garantizar la coherencia, transformación y fusión de datos de origen, creación de índices y vistas en tablas base, generación de desnormalizaciones, generación de agregaciones y copias de seguridad.
6. Administrador de consultas. Realiza todas las operaciones asociadas con la gestión de consultas de los usuarios.

3.2 Ontología

Una ontología define los términos básicos y las relaciones entre términos, que comprende el vocabulario de una temática, además de las reglas para combinar términos y relaciones, para definir extensiones del vocabulario (El Idrissi Esserhrouchni et al. 2014).

Las ontologías proporcionan varios beneficios para representar y procesar conocimiento, así como para compartir conocimiento común entre humanos y agentes de software, y para el reúso de conocimiento de dominio para una variedad de aplicaciones.

3.2.1 Componentes de una ontología

Existen diferentes formalizaciones de las ontologías, y cada una cuenta con diferentes componentes. Sin embargo, comparten el siguiente conjunto de componentes (Calero et al. 2006):

Clases. Representan conceptos que se toman en un sentido amplio. Las clases en las ontologías se organizan usualmente en taxonomías, a través de las cuales se pueden aplicar los mecanismos de herencia.

Relaciones. Representan un tipo de asociación entre conceptos del dominio. Usualmente las ontologías contienen relaciones binarias, donde el primer argumento se conoce como el dominio de la relación y el segundo argumento como el rango.

Axiomas formales. Sirven para modelar sentencias que siempre son verdaderas. Normalmente se usan para representar conocimiento que no puede definirse formalmente por otros componentes. Los axiomas formales se usan para verificar la consistencia del conocimiento en sí o el conocimiento almacenado en la base de conocimiento.

Instancias. Se usan para representar elementos o individuos en una ontología.

3.3 Extracción, Transformación y Carga (ETL)

Un sistema ETL (Extract, Transform, Load) propiamente diseñado extrae datos de los sistemas fuente, aplica los estándares de calidad y consistencia en los datos, ajusta los datos de fuentes separadas para que puedan usarse juntos y finalmente entrega los datos en un formato de presentación listo para que los desarrolladores puedan construir aplicaciones y los usuarios finales puedan tomar decisiones (Kimball & Caserta, 2004).

El sistema ETL añade un valor significativo a los datos. Es mucho más que obtener los datos de las fuentes y depositarlos en el almacén de datos. Específicamente el sistema ETL remueve

errores y corrige datos faltantes, proporciona medidas documentadas de confidencia en los datos, captura el flujo de datos transaccionales para su resguardo, ajusta datos de múltiples fuentes para usarse juntos y estructura los datos para que los usen las herramientas utilizadas por el usuario final (Kimball & Caserta, 2004).

3.4 Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R)

El Marco para la Caracterización de Proyectos de Software Refinado está agrupado por factores, que son conjuntos de atributos de un proyecto de software que permiten la clasificación y comparación de los proyectos. Es un marco de características de proyectos, que contiene cinco conjuntos de atributos presentes en el desarrollo de un proyecto de software, que se muestran en la Tabla 3.1.

Tabla 3.1. Factores del MCPS-R.
Fuente: extraído de (González et al., 2017).

Factor	Descripción
Factor del tipo de proyecto de software	Las características incluidas en este factor describen el tipo de proyecto a partir de su naturaleza, contexto, clasificación a la que pertenece, criticidad y riesgos que puede presentar el proyecto.
Factor medible	Las características incluidas en este factor permiten cuantificar los proyectos en términos de tamaño, costo, complejidad y tiempo de desarrollo.
Factor sociológico	Las características incluidas en este factor corresponden a las características del equipo de desarrollo e identifican las cualidades individuales de cada desarrollador.
Factor tecnológico	Las características incluidas en este factor describen la plataforma de desarrollo, dominio de la aplicación, paradigma de lenguaje de programación y herramientas a aplicar.
Factor configurable	Las características incluidas en este factor se requieren para analizar y tomar decisiones para configurarlos adecuadamente al desarrollo de un proyecto de software.

El Marco para la Caracterización de Proyectos de Software Refinado incluye 90 características agrupadas en los cinco factores mostrados en la Tabla 3.1. Cada característica se presenta por un identificador “X_n” (donde n identifica el número que corresponde a cada uno de los 90 atributos). Como ejemplo, en la Tabla 3.2 se muestran 3 atributos de los 8 del factor configurable: Conjunto de resultados a obtener, que incluyen cinco de los bits de la característica

X_{66} ($X(f)_{66}$, $X(i)_{66}$, $X(o)_{66}$, $X(w)_{66}$, $X(x)_{66}$); Forma de realizar inspecciones, que incluye tres características (X_{74} , X_{75} , X_{76}); y Modelo de desarrollo, que incluye a la característica X_{77} . Los atributos y características incluidos en cada factor del MCPS-R se pueden revisar en el Anexo A.

Tabla 3.2. Ejemplo de atributos del factor configurable en proyectos de software (ocho atributos).

Fuente: creación propia.

Atributo	ID	Criterio para la asignación del valor
Conjunto de resultados a obtener	$X(f)_{66}$	Documentación de Revisión de requerimientos y prototipo del sistema.
	$X(i)_{66}$	Documentación de Revisión de requerimientos de software y especificación de pruebas de aceptación.
	$X(o)_{66}$	Documentación de Revisión de arquitectura de software.
	$X(w)_{66}$	Documentación de Prueba de integración.
	$X(x)_{66}$	Documentación de Prueba de aceptación.
Forma de realizar inspecciones	X_{74}	Tipo de inspección: técnica formal o técnica informal.
	X_{75}	Duración.
	X_{76}	Número promedio de asuntos o defectos identificados.
Modelo de desarrollo	X_{77}	Principal modelo de proceso de desarrollo.

3.5 Algoritmo de Levenshtein

El algoritmo de Levenshtein también conocido como distancia de edición, es un algoritmo que mide la distancia entre palabras y también se usa para la conversión de una palabra a otra. Debe su nombre al matemático ruso Vladimir Levenshtein, el algoritmo de Levenshtein funciona para alfabetos que en su composición solo tienen letras que consisten en un solo carácter como el idioma inglés (Rashiti, 2017).

Una forma para obtener el grado de similitud entre dos palabras distintas incluye el cálculo de la distancia de Levenshtein, que consiste en calcular el número mínimo de operaciones necesarias para transformar una cadena de caracteres en otra. Obtenida esta distancia se puede calcular fácilmente el grado de similitud porcentual (afinidad) entre las dos palabras utilizando la siguiente fórmula:

$$\text{Afinidad} = 1 - (D/L)$$

Donde:

D: Distancia de Levenshtein

L: Longitud de la palabra mas larga

3.6 Núcleo de Essence

El estándar *Essence* generaliza la ingeniería de software, identificando acciones y elementos universales; permite evaluar, comparar y medir las prácticas comunes de métodos existentes (Scrum, RUP, AGD, etc.), describiéndolos en un lenguaje sencillo y universal. Para dicha descripción se utiliza el núcleo del estándar, que contiene un pequeño número de “cosas con las que siempre se trabaja” (véase la Figura 3.2), “cosas que siempre se hacen” (véase la Figura 3.3) y “habilidades necesarias” (véase la Figura 3.4), cuando se desarrollan sistemas de software (Jacobson et al., 2013).

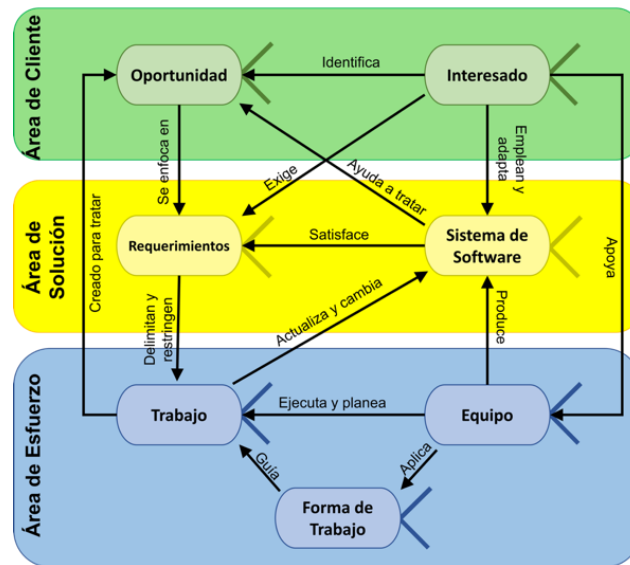


Figura 3.2 Relación entre las siete alfas.

Fuente: traducida de (OMG, 2015)

Essence estandariza la ingeniería de software, incluyendo un conjunto de elementos comunes que sirven de referencia para relacionar y diferenciar métodos de desarrollo de software. *Essence* se organiza en tres áreas de interés que se centran en aspectos específicos de la ingeniería de software: clientes (color verde), esta área contiene todo lo relacionado con el uso real y la explotación del sistema de software que se va a producir; solución (amarillo), esta área contiene todo para hacer la especificación y el desarrollo del sistema de software; y esfuerzo (azul), esta área contiene todo lo relacionado con el equipo, y la forma en que se acercan a su trabajo.

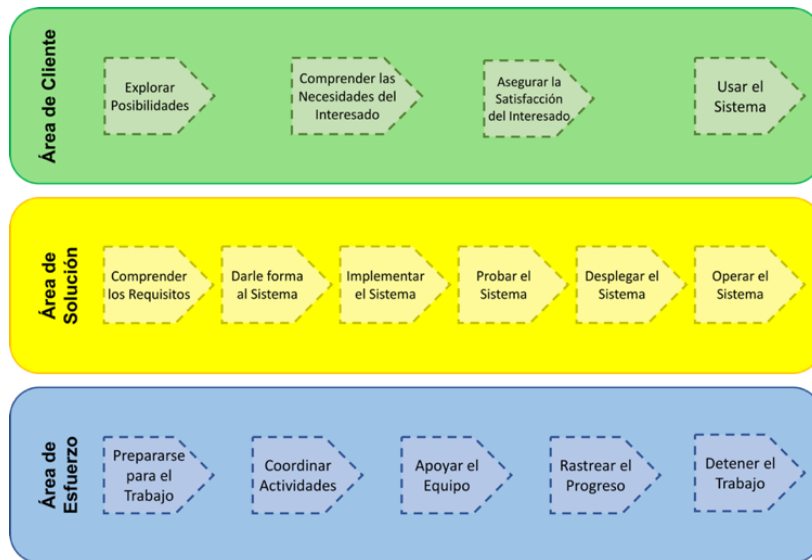


Figura 3.3 Espacios de actividad según su orden de aparición en el ciclo de vida de un proyecto de software.

Fuente: traducida de (OMG, 2015)

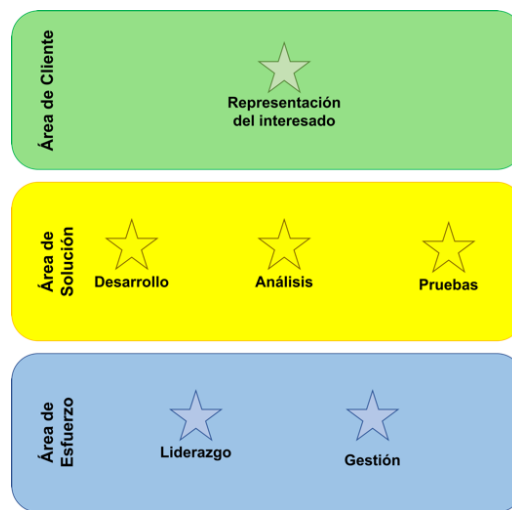


Figura 3.4 Representación de competencias del núcleo de Essence.

Fuente: traducida de (OMG, 2015)

3.7 Área de Minado de Repositorios de Software

La comunidad de MSR (*Mining Software Repositories*) analiza los abundantes datos disponibles en repositorios de software, para descubrir información interesante y procesable sobre sistemas de software y proyectos de software. Los repositorios de software como: los sistemas de administración de configuración de código fuente, las comunicaciones entre el personal del proyecto, archivadas y los sistemas de seguimiento de defectos. Se utilizan para ayudar a administrar el progreso de los proyectos de software.

Los profesionales del software y los investigadores están reconociendo los beneficios de utilizar esta información para apoyar el mantenimiento de los sistemas de software, mejorar el diseño / reutilización del software y validar empíricamente nuevas ideas y técnicas. Situación que causa que se investiguen las maneras en que la minería de estos repositorios puede ayudar a comprender el desarrollo de software y la evolución del software, para:

- Respalda las predicciones sobre el desarrollo del software y
- Aprovechar este conocimiento al planificar el desarrollo futuro.

Casos de estudio MSR

El propósito de analizar los siguientes casos de estudio del grupo MSR, es tomar en cuenta a la comunidad como usuarios potenciales del almacén de datos dw_manager, en un futuro. Y se consideró necesario identificar las necesidades específicas que tiene el MSR, con respecto al análisis de la información de proyectos de desarrollo de software libre.

1.- A Systematic Mapping Study on Mining Software Repositories, (de F. Farias et al., 2016), este trabajo tiene como objetivo investigar estudios recientes sobre los enfoques de repositorios de software de minería (MSR) recopilando evidencias sobre objetivos de análisis de software (propósito, enfoque y objeto de análisis), fuentes de datos, métodos de evaluación, herramientas y cómo el área está evolucionando. Se realizó un estudio de mapeo sistemático para identificar y analizar la investigación en repositorios de software de minería sobre los estudios publicados en cinco años de MSRConf (de 2010 a 2014).

El campo Mining Software Repositories (MSR) se centra en descubrir información interesante y útil sobre software y proyectos extrayendo y analizando datos disponibles de diferentes repositorios de software (Hassan, 2008).

Los repositorios de software contienen una gran cantidad de datos históricos de software que pueden incluir información valiosa sobre el código fuente, defectos y otros problemas, como nuevas características. De este modo, MSR se ha convertido en un área importante para las investigaciones de Ingeniería de Software y una herramienta popular para estudios empíricos en esta área (G. Robles, U. Re y, and J. Carlos, 2010).

Esto permitió investigar cómo se llevan a cabo las investigaciones en este campo y abordar la siguiente pregunta de investigación: "**¿Cuáles son los objetivos actuales abordados por los enfoques recientes en MSR?**". Al responder esta pregunta se ha identificado que los enfoques MSR se han utilizado para muchos objetivos diferentes, principalmente para la comprensión de

defectos, el análisis de la contribución y el comportamiento de los desarrolladores, y la comprensión de la evolución del software (de F. Farias et al., 2016).

En la Tabla 3.3 muestra la categoría de propósito que se encontró en el estudio de mapeo y el número de estudios por cada categoría.

Tabla 3.3. Clasificación de los propósitos de investigación de los estudios MSR.

Número	Categoría de propósito	Cantidad de estudios	Porcentaje	Porcentaje acumulado	Principales propósitos
1	Comprensión	58	47.9%	47.9%	
2	Predicción	20	16.5%	64.5%	
3	Identificación	18	14.9%	79.3%	
4	Evaluación	12	9.9%	89.3%	
5	Clasificación	6	5.0%	94.2%	
6	Mejora	4	3.3%	97.5%	
7	Caracterización	2	1.7%	99.2%	
8	Localización	1	0.8%	100%	
		121	100%		

En la Tabla 3.3 la categoría "comprensión" tiene el mayor número de estudios. De él, podemos ver que:

- 58 - 47.9% de los estudios presentan al menos un enfoque que pretende comprender algún tipo de artefactos de software;
- 20 - 16.5% fueron clasificados como "predicción";
- 18 - 14.9% como "identificación",
- y 12 - 9.9% como "evaluación".

Estos cuatro propósitos representan el 89.3% de los estudios que se analizaron en este mapeo. Controversialmente, los otros cuatro propósitos representan solo el 10.8% de todos los estudios (de F. Farias et al., 2016).

En la Tabla 3.4 muestra la categoría de enfoque que se encontró en el estudio de mapeo y el número de estudios por cada categoría.

Tabla 3.4. Categorías de enfoque de investigación de los estudios MSR.

Número	Categoría de enfoque	Porcentaje	Porcentaje acumulado	Principales enfoques
1	Defectos	31.2%	31.2%	
2	Contribución y comportamiento de desarrolladores	12.8%	44.0%	
3	Cambios	8.0%	52.0%	
4	Artefactos de calidad del software	7.2%	59.2%	

5	Evolución del software	6.4%	65.6%	
6	Procesos de desarrollo	6.4%	72.0%	
7	Clon de código	4.8%		
8	Reuso del software	4.0%		
9	Matenabilidad	3.2%		
10	Características del software	3.2%		
11	Consumo de energía	2.4%		
12	Licencias del software	2.4%		
13	Esfuerzo del desarrollo	1.6%		
14	Complejidad del software	1.6%		
15	CódigoSmell	1.6%		
16	Confiabilidad	0.8%		
17	Desempeño del software	0.8%		
18	Utilización del software y APIs	0.8%		
19	Refactorización	0.8%		
		100%		

En la Tabla 3.4 podemos notar que el enfoque:

- "defecto" con 39 - 31.2% (también relacionado con el tema de error),
- "cambios" con 10 - 8% (también relacionado con el cambio),
- y "evolución del software" con 8 - 6.4% (también relacionado con la evolución de errores)

Representan la mitad de los estudios analizados, siendo el enfoque más explorado por los estudios analizados. Revelaron que el área de MSR tiene una fuerte tendencia hacia el cambio y la evolución del software. Términos como "defectos", "cambio" y "evolución" aparecen casi todos los años y se clasificaron como uno de los temas de investigación más populares y como tendencia del campo MSR (de F. Farias et al., 2016).

En la Tabla 3.5 muestra la distribución de los estudios por tipo de objeto de análisis.

Tabla 3.5. Estudios por objetos de análisis de investigación de los estudios MSR.

Número	Objeto de análisis	Cantidad de estudios	Porcentaje	Porcentaje acumulado	Principales objetos
1	Código	41	28.5%	28.5%	
2	Datos compromiso	29	20.1%	48.6%	
3	Reportes de error (bug)	26	18.1%	66.7%	

4	Comentarios	13	9.0%	75.7%	
5	Microbloggeo	8	5.6%	81.3%	
6	E-mail	6	4.2%	85.4%	
7	Código binario	6	4.2%		
8	Datos del desarrollador	3	2.1%		
9	Datos de prueba	3	2.1%		
10	Biblioteca de tercera parte	3	2.1%		
11	Rastreo de la pila (Stack)	2	1.4%		
12	Métricas del software	1	0.7%		
13	Apps Móviles	1	0.7%		
14	Texto en bitácoras	1	0.7%		
15	Texto en documentos	1	0.7%		
		144	100%		

En la Tabla 3.5 podemos notar que:

- "Código" con 41 - 28.5%,
- "Datos de compromiso" con 29 - 20.1% ",
- " Informes de errores "con 26 - 18.1%,
- " Comentarios "con 13 - 9.0%,
- " Microblogging "con 8 - 5.6%,
- y" E-mail "con 6 - 4.2% son los seis objetos de análisis más significativos.

Representan el 85.4% de todos los estudios. Se encuentra un estudio que analiza microblogging en cada año con la excepción de 2010, destacando que la popularización de sistemas de preguntas y respuestas como Stackoverflow2, utilizado por los miembros del equipo de software, contribuyó para la propagación del "microblogging" como objeto de análisis (de F. Farias et al., 2016).

Analizando las tablas anteriores puedo notar que se utilizaron un gran conjunto de propósitos y enfoques para investigar artefactos de MSR, pero la "**comprensión de defectos**" y el "**código**" fue el objetivo, el enfoque y el objeto de análisis más utilizados en este campo.

Por lo tanto, concluyo que:

- La "**comprensión**" fue el propósito que se combinó con casi todas las facetas de enfoque.
- Ningún trabajo tiene la finalidad de comprender la refactorización y el rendimiento del software.

- El enfoque “defectos” es la única categoría que ha sido explorada por todos los propósitos analizados. Los propósitos más usados para esta categoría son “comprensión”, “predicción”, “identificación”, “evaluación”.
- Los enfoques “cambios” y “clones de código” usan como objeto de análisis “código”.

2.- The Road Ahead for Mining Software Repositories, (Hassan, 2008), la investigación empírica sobre el desarrollo de software basada en datos obtenidos de repositorios de proyectos está ganando cada vez más atención en la comunidad de investigación de ingeniería de software. Los estudios en esta área generalmente comienzan recuperando o monitoreando algunos de los datos encontrados en el repositorio, y estos datos son analizados posteriormente para encontrar patrones interesantes. Los desarrolladores utilizan su experiencia al agregar una nueva característica o corregir un error. Los probadores priorizan las pruebas de las características que se sabe que son propensas a errores en función de los informes de campo y errores. Los repositorios de software contienen una gran cantidad de información valiosa sobre proyectos de software. Utilizando la información almacenada en estos repositorios, los profesionales pueden depender menos de su intuición y experiencia, y dependen más de los datos históricos y de campo (Hassan, 2008).

La investigación MSR se centra principalmente en dos aspectos:

- La creación de técnicas para automatizar y mejorar la extracción de información de los repositorios.
- El descubrimiento y la validación de nuevas técnicas y enfoques para extraer información importante de estos repositorios.

En la Figura 3.5 se observa una gráfica de las publicaciones en la sede de MSR en los últimos cinco años a lo largo de varios tipos de repositorios. Muestra que una gran cantidad (80%) del trabajo publicado se centra en el código fuente y repositorios relacionados con errores (Hassan, 2008).

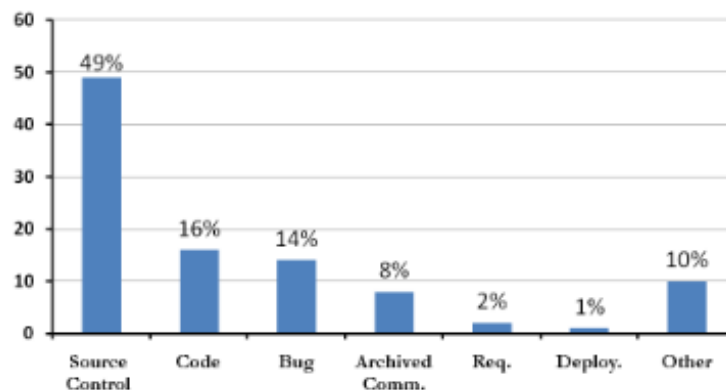


Figura 3.5 Repositorios en publicaciones de MSR

Fuente: Obtenida de (Hassan, 2008)

Analizando la figura anterior se muestra que los repositorios de documentación (por ejemplo, los requisitos) rara vez se estudian principalmente debido a la disponibilidad limitada de dichos repositorios. Estos datos se obtuvieron de examinar los artículos completos y breves publicados en los procedimientos de MSR de 2004 a 2008. En la Tabla 3.6, se describen los tipos de repositorios que se analizarán.

Tabla 3.6. Tipos de repositorios MSR.

Repositorios	Descripción
Source control	Estos repositorios registran el historial de desarrollo de un proyecto. Los repositorios de control de fuente son el repositorio más comúnmente disponible y usado en proyectos de software.
Bug	Estos repositorios rastrean el historial de resolución de informes de errores o peticiones de características que son reportados por usuarios y desarrolladores de grandes proyectos de software.
Archived communications	Estos repositorios rastrean las discusiones sobre varios aspectos de un proyecto de software a lo largo de su vida útil. Las listas de correo, los correos electrónicos, los chats de IRC y los mensajes instantáneos son ejemplos de comunicaciones archivadas sobre proyectos.
Deployment logs	Estos repositorios registran información sobre la ejecución de una única implementación de una aplicación de software o diferentes implementaciones de las mismas aplicaciones.
Code	Estos repositorios archivan el código fuente para una gran cantidad de proyectos. Sourceforge.net y el código de Google son ejemplos de grandes repositorios de código.

Los investigadores de MSR han propuesto técnicas que aumentan los datos, técnicas y herramientas tradicionales de ingeniería de software para resolver problemas importantes y desafiantes, como la identificación de errores y la reutilización de códigos, que los profesionales deben enfrentar y resolver a diario.

3.8 Comparación de Factores MCPS-R

Se realizó un reporte sobre la comparación de características del MCPS-R y los factores de éxito considerados en tres enfoques de desarrollo: metodologías ágiles, tradicionales e híbridas.

3.8.1 Metodologías Tradicionales

1.- Prioritization of Critical Success Factors In The Process of Software Development, (Ribeiro, Duarte, Salgado, & Castro, 2017), este estudio tuvo como objetivo identificar, analizar y priorizar los factores o áreas que cuando se manejan bien producen resultados favorables en el proceso de desarrollo del software, utilizando el método de toma de decisiones AHP (Analytic Hierarchy Process).

(Rockart & Sloan, 1978) fueron quienes introdujeron el término de factores críticos de éxito: *“Factores Críticos de Éxito son algunas áreas de actividades clave, cuyos resultados favorables son absolutamente necesarios para que los gerentes alcancen sus objetivos”*. Desarrolló un método empírico para la determinación de FCS (Factors Critical to Success), utiliza entrevistas estructuradas para identificar las prioridades gerenciales.

Después de la revisión de la literatura e identificación de los FCS, se elaboró la estructura para el AHP. La estructura final contó con 5 criterios y 29 subcriterios, como se muestra en la Fig. 3.6.

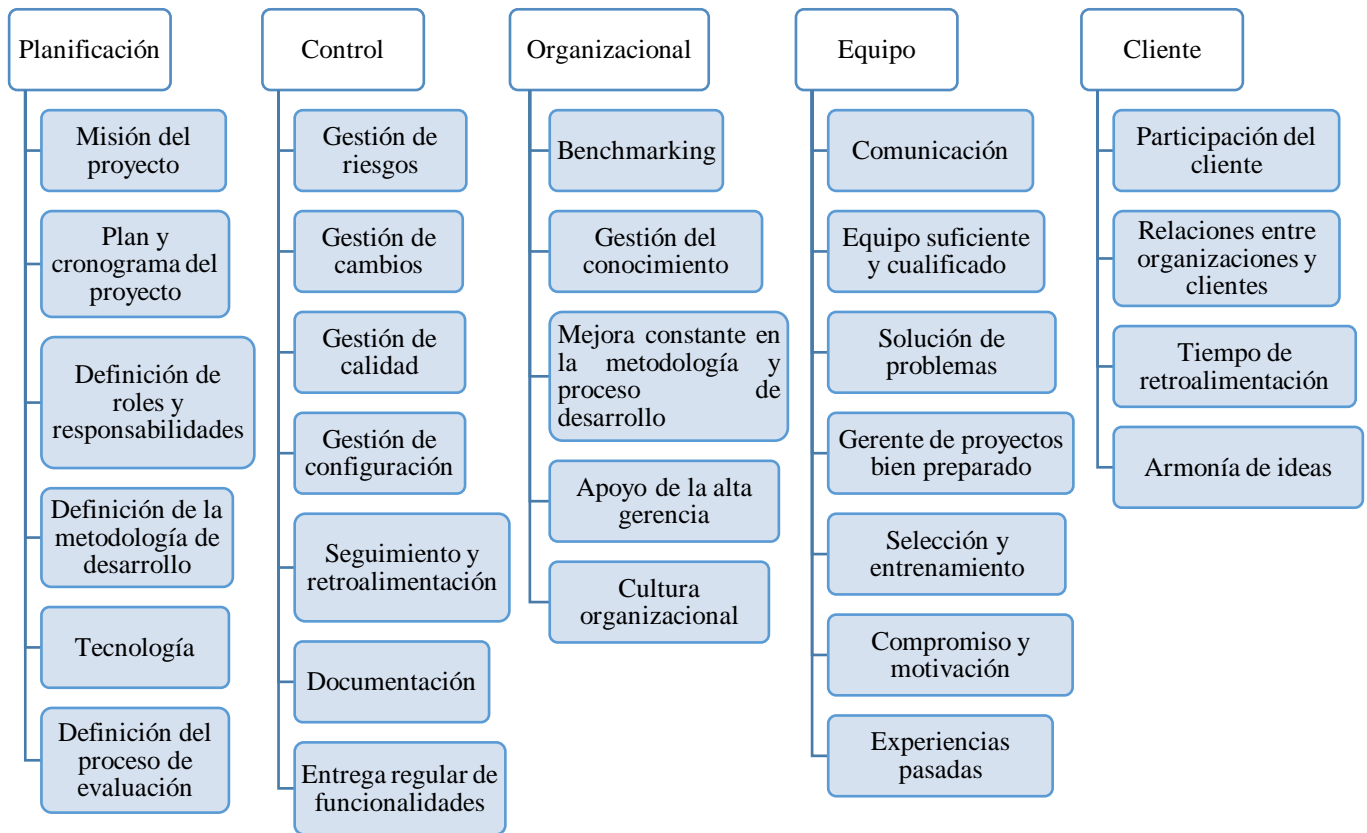


Figura 3.6 Estructuración de los FCS de forma jerárquica
Fuente: Obtenida de (Ribeiro et al., 2017)

De la estructura anterior de la Figura 3.6 los cinco principales factores son:

- Plan y cronograma del proyecto (6.47%),
- Participación del cliente (5.96%),
- Definición de la metodología para el desarrollo (4.43%),
- Compromiso y motivación (4.10%),
- Misión del proyecto (4.09%).

En el ranking de los factores de menor importancia se encuentran:

- Benchmarking (1,12%),
- Cultura Organizacional (1,52%),
- Apoyo de la Alta Gerencia (1,71%).

En la Tabla 3.7, se visualiza la comparación realizada del MCPS-R con los FCS obtenidos de (Ribeiro et al., 2017).

Tabla 3.7. Comparación MCPS-R con los FCS

FCS	MCPS-R		Kernel de Essence	
	Factor	Característica	Alfas	Competencias
Planificación				
Misión del proyecto	Tipo de proyecto de software	X1 Clasificación del software. X2 Contexto o dominio del proyecto.	Oportunidad Interesados	Liderazgo
Plan y cronograma del proyecto	Medibles	X41-X43 Tiempo de desarrollo del proyecto.	Trabajo	Liderazgo Gestión
Definición de roles y responsabilidades	Configurables	X69 Estructura del equipo de trabajo.	Equipo	Liderazgo Gestión
Definición de la metodología de desarrollo	Configurables	X77 Modelo de desarrollo.	Forma de trabajar	Liderazgo Gestión
Tecnología	Tecnológicos	X57-X65	Oportunidad Requerimientos Forma de trabajar Sistema de software	Liderazgo Gestión Desarrollo Análisis

Definición del proceso de evaluación	Configurables	X71-X73 Forma de realizar el seguimiento y control del proceso.	Forma de trabajar	Gestión
Control				
Gestión de riesgos	Tipo de proyecto de software	X4-X15 Riesgos del proyecto.	Oportunidad	Liderazgo Análisis
Gestión de cambios	Configurables	X66-X77	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
Gestión de calidad				
Gestión de configuración				
Seguimiento y retroalimentación				
Documentación				
Entrega regular de funcionalidades				
Organizacional				
Benchmarking	Tipo de proyecto de software	X3 Naturaleza del proyecto.	Interesados	Liderazgo
Gestión del conocimiento	Sociológicos	X46-X56	Equipo	Liderazgo Gestión
Mejora constante en la metodología y proceso de desarrollo	Sociológicos	X52-X53 Experiencia del equipo en la metodología de desarrollo.	Equipo	Liderazgo Gestión
Apoyo de la alta gerencia	Configurables	X66-X77	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
Cultura organizacional	Sociológicos	X46-X56	Equipo	Liderazgo Gestión
Equipo				
Comunicación	Sociológicos	X46-X56	Equipo	Liderazgo Gestión
Equipo suficiente y cualificado				
Solución de problemas				
Gerente de proyectos bien preparado				
Selección y entrenamiento				
Compromiso y motivación				

Experiencias pasadas				
Cliente				
Participación del cliente	No se encuentra en ninguno de los factores ya que se enfoca al ámbito personal y no al desarrollo del software.			
Relaciones entre organizaciones y clientes				
Tiempo de retroalimentación				
Armonía de ideas				

Analizando la Tabla 3.7 de comparación puedo concluir que el MCPS-R incluye la mayoría de FCS para el desarrollo de software con excepción del factor cliente. Ya que el MCPS-R se enfoca en el proceso del desarrollo de software y no al ámbito personal.

3.8.2 Metodologías Ágiles

2.- Modelling the Critical Success Factors of Agile Software Development Projects in South África, (Tawanda B. & Mnkandla, 2017), este estudio tiene como objetivo identificar y proporcionar información sobre los factores críticos de éxito que influyen en el éxito de los proyectos de desarrollo de software utilizando metodologías ágiles en Sudáfrica. Existe la necesidad de que los administradores de desarrollo de software comprendan cuándo es apropiado usar metodologías ágiles y cuándo usar tradicionales (Taromirad & Ramsin 2008). Se utilizó un método de investigación cuantitativa en este estudio para comprender la experiencia práctica de profesionales ágiles con el éxito del proyecto ágil explorando los factores humanos, de proyecto, de proceso, organizativos, expectativa de rendimiento, éxito real y tecnológicos en organizaciones sudafricanas.

En la Tabla 3.8, se visualiza los factores de éxito de los proyectos ágiles obtenidos de (Tawanda B. & Mnkandla, 2017).

Tabla 3.8. Factores de éxito de los proyectos ágiles

Dimensión y/o intensidad	Factores
Organizativo	Instalaciones con un entorno de trabajo de estilo ágil adecuado.
	Colocación de todo el equipo.
	Cultura organizacional cooperativa en lugar de jerarquizada.
	Cultura oral que otorga un alto valor a la comunicación fluida y cara a cara.
	La metodología ágil es universalmente aceptada en la organización.
	Sistema de recompensa que es apropiado para ágil.
Éxito real	Calidad (entrega del resultado del proyecto).
	Alcance (que cumple todos los requisitos).
	Tiempo (entrega a tiempo).
	Costo (entrega dentro del costo estimado).

	Calidad (entrega del buen resultado del proyecto).
	Alcance (cumplir todos los objetivos).
	Costo (entrega dentro del esfuerzo estimado).
Proceso	Fuerte enfoque de comunicación con reuniones diarias cara a cara.
	Siguiendo procesos de gestión de requisitos orientados de forma ágil.
	El alcance del proyecto está bien definido.
	Los profesionales ágiles siguen un mecanismo ágil de seguimiento del progreso.
	Siguiendo el proceso de gestión de configuración orientado a ágiles.
Gente	Buena relación con el cliente.
	Trabajo en equipo coherente, auto organizado y equipo motivado.
	Los gerentes que tienen un toque ligero o un estilo de manejo adaptativo.
	Miembros del equipo con alta competencia y experiencia tecnológica.
Tecnológico	Proyecto que impone un estándar de codificación bien definido por adelantado.
	Proyecto que persigue un diseño simple.
	El proyecto persigue actividades de refactorización vigorosas.
	Proyecto mantiene la cantidad correcta de documentación.
Proyecto	Proyecto con un horario dinámico y acelerado.
	Proyecto sin equipos múltiples e independientes trabajando juntos.
	Proyectos con evaluación de costos inicial completa.
Expectativa de rendimiento	Los proyectos de desarrollo de software ágil son una esperanza útil.
	Los proyectos ágiles de desarrollo de software permiten realizar tareas.
	Los proyectos ágiles de desarrollo de software aumentan mi productividad.
	Aumentar las posibilidades de promoción.

En la Tabla 3.9, se visualiza la comparación realizada del MCPS-R con los factores de éxito de los proyectos ágiles obtenidos de (Tawanda B. & Mnkandla, 2017).

Tabla 3.9. Comparación MCPS-R con los factores de éxito de los proyectos ágiles

Dimensión y/o intensidad	Factor	MCPS-R		Kernel de Essence	
		Factor	Característica	Alfas	Competencias
Organizativo	Colocación de todo el equipo. La metodología ágil es universalmente aceptada en la organización.	Sociológicos	X49 Experiencia del equipo en el desarrollo del proyecto. X52-X53 Experiencia del equipo en la metodología de desarrollo.	Equipo	Liderazgo Gestión
	Sistema de recompensa que es apropiado para ágil. Cultura organizacional cooperativa en lugar de jerarquizada.	Configurables	X70 Forma de realizar el desarrollo.	Equipo	Liderazgo Gestión
Éxito real	Tiempo (entrega a tiempo). Costo (entrega dentro del costo estimado). Calidad (entrega del buen resultado del proyecto).	Medibles	X16-X45	Sistema de software Trabajo Oportunidad Interesados	Desarrollo Liderazgo Pruebas Análisis Gestión
Proceso	Siguiendo procesos de gestión de requisitos orientados de forma ágil. El alcance del proyecto está bien definido. Los profesionales ágiles siguen un mecanismo ágil de seguimiento del progreso. Siguiendo el proceso de gestión de configuración orientado a ágiles.	Configurables	X66 Documentación de requerimientos de sistema. Documentación de revisión de requerimientos y prototipo del sistema.	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
Gente		No se encuentra en ninguno de los factores ya que se enfoca al ámbito personal y no al desarrollo del software.			
Tecnológico	Proyecto que impone un estándar de codificación bien		X3 Naturaleza del proyecto.	Interesados	Liderazgo

	definido por adelantado. Proyecto que persigue un diseño simple.	Tipo de proyecto de software			
	El proyecto persigue actividades de refactorización vigorosas. Proyecto mantiene la cantidad correcta de documentación.	Configurables	X66 Conjunto de productos de trabajo a obtener.	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
Proyecto	Proyectos con evaluación de costos inicial completa.	Medibles	X36-X37 Costo del proyecto. X40 Complejidad de organización.	Trabajo Oportunidad Interesados	Liderazgo
	Proyecto con un horario dinámico y acelerado. Proyecto sin equipos múltiples e independientes trabajando juntos.	Configurables	X69 Estructura del equipo de trabajo.	Equipo	Liderazgo Gestión
Expectativa de rendimiento	Los proyectos ágiles de desarrollo de software permiten realizar tareas.	Configurables	X71-X73 Forma de realizar el seguimiento y control de proceso. X74-X76 Forma de realizar inspecciones. X77 Modelo de desarrollo.	Forma de trabajar	Gestión Liderazgo

Analizando la Tabla 3.9 de comparación puedo concluir que el MCPS-R incluye la mayoría de factores de éxito de los proyectos ágiles con excepción del factor gente. Ya que el MCPS-R se enfoca en el proceso del desarrollo de software y no al ámbito personal. El factor gente que maneja la metodología ágil se enfoca al cliente.

3.8.3 Metodologías Híbridas

3.- Prioritizing key success factors of software projects using fuzzy AHP, (Yaghoobi, 2018), este estudio tiene como objetivo identificar y clasificar los factores y aspectos clave que tienen efecto sobre el éxito de un proyecto de desarrollo de software mediante la incorporación de una técnica de AHP difusa. Los factores de éxito de un proyecto de software son aquellos elementos o actividades que se requieren para garantizar el logro del proyecto. La estructura jerárquica de AHP de este estudio se muestra en la Fig. 3.7, el objetivo general es el nivel superior, mientras que las perspectivas y los indicadores se encuentran en los niveles siguientes.

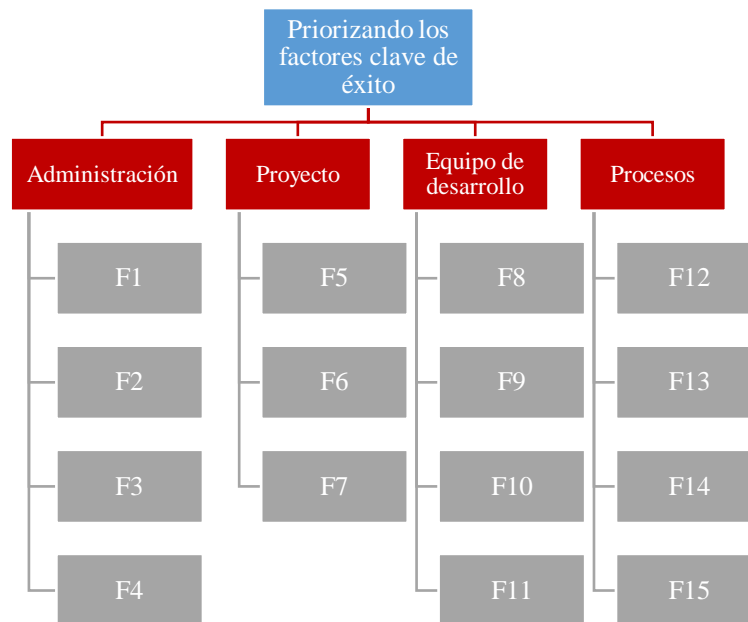


Figura 3.7 Estructura analítica del proceso de jerarquía AHP

Fuente: Traducida de (Yaghoobi, 2018)

En la Tabla 3.10, se visualiza los factores clave de éxito (KSFs) para proyectos de software en 4 perspectivas obtenidos de (Yaghoobi, 2018).

Tabla 3.10. Factores clave de éxito (KSFs) para proyectos de software

Perspectiva	Indicador (KSF)	Definición
Administración	F1	Presupuesto realista.
	F2	Horario realista.
	F3	Habilidades/Metodologías eficientes de gestión de proyectos.
	F4	Fuerte apoyo ejecutivo.
Proyecto	F5	Borrar requisitos y especificaciones.
	F6	Naturaleza/Tipo del proyecto (tamaño/complexidad).
	F7	Cambiando las demandas de la aplicación.
Equipo de desarrollo	F8	Miembros del equipo con alta competencia y experiencia.
	F9	Familiarizado con la tecnología/metodología de desarrollo.
	F10	Tamaño del equipo.
	F11	Comunicación eficiente con el gerente del proyecto.

Procesos	F12	Herramientas apropiadas y procesos/metodologías de desarrollo.
	F13	Informe de progreso actualizado.
	F14	Participación del usuario/cliente.
	F15	Planificación adecuada.

En la Tabla 3.11, se visualiza la comparación realizada del MCPS-R con los factores clave de éxito (KSFs) obtenidos de (Yaghoobi, 2018).

Tabla 3.11. Comparación MCPS-R con los factores clave de éxito (KSFs)

KSFs	MCPS-R		Kernel de Essence	
	Factor	Característica	Alfas	Competencias
F1 Presupuesto realista.	Medibles	X36-X37 Costo del proyecto.	Trabajo Oportunidad	Liderazgo
F2 Horario realista.	Medibles	X42 Tiempo de duración real en el desarrollo del proyecto.	Trabajo	Liderazgo Gestión
F3 Habilidades/Metodologías eficientes de gestión de proyectos.	Sociológicos	X52-X53 Experiencia del equipo en la metodología de desarrollo.	Equipo	Liderazgo Gestión
F4 Fuerte apoyo ejecutivo.	Configurables	X69 Estructura del equipo de trabajo.	Equipo	Liderazgo Gestión
F5 Borrar requisitos y especificaciones.	Configurables	X66 Documentación de requerimientos de sistema. Documentación de revisión de requerimientos y prototipo del sistema.	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
F6 Naturaleza/Tipo del proyecto (tamaño/complejidad).	Tipo de proyecto de software	X2 Contexto o dominio del proyecto. X3 Naturaleza del proyecto.	Interesados	Liderazgo
F7 Cambiando las demandas de la aplicación.	Configurables	X66 Documentación de revisión de	Forma de trabajar Interesados	Liderazgo Gestión Análisis

		requerimientos de software y especificación de pruebas de aceptación.	Sistema de software Requerimientos Trabajo	Desarrollo Pruebas
F8 Miembros del equipo con alta competencia y experiencia.	Sociológicos	X47 Nivel máximo de educación.	Equipo	Liderazgo Gestión
F9 Familiarizado con la tecnología/metodología de desarrollo.	Sociológicos	X48 Experiencia del equipo en el desarrollo del proyecto. X50 Experiencia del equipo en el lenguaje de programación.	Equipo	Liderazgo Gestión
F10 Tamaño del equipo.	Sociológicos	X46 Número de desarrolladores.	Equipo	Liderazgo Gestión
F11 Comunicación eficiente con el gerente del proyecto.	Configurables	X69 Estructura del equipo de trabajo.	Equipo	Liderazgo Gestión
F12 Herramientas apropiadas y procesos/metodologías de desarrollo.	Tecnológicos	X57-X65	Oportunidad Requerimientos Forma de trabajar Sistema de software	Liderazgo Gestión Desarrollo Análisis
F13 Informe de progreso actualizado.	Configurables	X66 Conjunto de productos de trabajo a obtener.	Forma de trabajar Interesados Sistema de software Requerimientos Trabajo	Liderazgo Gestión Análisis Desarrollo Pruebas
F14 Participación del usuario/cliente.	No se encuentra en ninguno de los factores ya que se enfoca al ámbito personal y no al desarrollo del software.			
F15 Planificación adecuada.	Tipo de proyecto de software	X1-X15	Oportunidad Interesados	Liderazgo Análisis

Analizando la Tabla 3.11 de comparación se concluye que el MCPS-R incluye la mayoría de KSFs con excepción del F14 Participación del usuario/cliente. Ya que el MCPS-R se enfoca en el proceso del desarrollo de software y no al ámbito personal.

En este capítulo se presentaron conceptos importantes para el objetivo de esta investigación y para que el enfoque de esta investigación sea comprendido por los lectores. En el siguiente capítulo se describe como se diseña el almacén de datos y como se seleccionan los repositorios para la realización de pruebas de esta investigación.

Capítulo 4

Diseño del almacén de datos

En el presente capítulo, se describe el diseño del almacén de datos para el cual se realizó un modelo conceptual, lógico y físico. También, se presenta un análisis de los repositorios de software de acceso libre, que cuentan con datos de proyectos reales. Y se incluyen las reglas especificadas para seleccionar los proyectos que se utilizarán para la creación de la estructura del almacén y la realización de pruebas.

A continuación, se presenta la estructura del capítulo:

- 4.1 Repositorios FLOSSmole
- 4.2 Diseño conceptual del almacén de datos
 - 4.2.1 Tarea1: Construir un glosario de términos
 - 4.2.2 Tarea2: Construir una taxonomía de conceptos
 - 4.2.3 Tarea3: Construir diagramas de relaciones binarias ad hoc
 - 4.2.4 Tarea4: Construir el diccionario de conceptos
 - 4.2.5 Tarea5: Describir en detalle las relaciones binarias
 - 4.2.6 Formalización
- 4.3 Diseño lógico del almacén de datos
 - 4.3.1 factors
 - 4.3.2 alphas
 - 4.3.3 milestones
 - 4.3.4 mcpsr
 - 4.3.5 competencias
 - 4.3.6 mcpsr_competency
 - 4.3.7 repositories
 - 4.3.8 tables
 - 4.3.9 columns_mcpsr
 - 4.3.10 queries_metadata
 - 4.3.11 operationals_metadata
 - 4.3.12 state_alpha
 - 4.3.13 Relaciones
- 4.4 Diseño físico del almacén de datos

4.1 Repositorios FLOSSmole

En la base de datos FLOSSmole se encuentran disponibles repositorios con una enorme cantidad de datos y únicamente se utilizarán los repositorios de software que cumplan con las reglas siguientes:

R1. El repositorio tiene que encontrarse activo (FLOSSmole conserva repositorios de proyectos de software que ya no se encuentran activos, por lo que en estos repositorios ya no se ingresan

nuevos datos. El repositorio tiene que estar activo, para que la herramienta tenga una mayor utilidad con el tiempo).

R2. El repositorio contenga sus versiones de datos (cada versión de datos corresponde a los datos recolectados de la fuente original en un periodo de tiempo).

R3. El repositorio contenga información considerable sobre el desarrollo de software.

R4. El repositorio contenga menos de cien mil registros por tabla.

Las reglas anteriores se tomaron en cuenta de la investigación(Aguilar, 2017) y se aplicaron a cada uno de los repositorios disponibles en la base de datos FLOSSmole, para seleccionar los que se usan en la investigación. En la Tabla 4.1 se muestra si un repositorio cumple (✓) o no (✗) con dichas reglas.

Tabla 4.1. Selección de repositorios FLOSSmole
Fuente: creación propia.

Repositorio	R1	R2	R3	R4
Alioth	✓	✓	✓	✓
Apache_irc	✓	✗	✗	✓
Debian	✓	✓	✗	✗
Freecode	✗	✗	✓	✗
Free Software Foundation	✓	✓	✗	✗
Github	✓	✗	✗	✗
Google Code	✗	✗	✓	✗
IRC	✗	✗	✗	✗
Launchpad	✓	✗	✓	✗
ObjectWeb	✗	✓	✓	✓
Opentask	✓	✗	✓	✗
Rubyforge	✓	✓	✓	✓
Rubygems	✓	✓	✗	✗
Savannah	✓	✗	✓	✓
Sourceforge	✓	✗	✓	✗
Tigris	✓	✓	✓	✓
Ultimate Debian Database	✗	✓	✓	✗
Wordpress	✓	✗	✗	✗

Los repositorios que cumplen con la totalidad de las reglas y que se usan en esta investigación son: **Alioth, RubyForge y Tigris**. La cantidad de tablas, atributos y versiones de datos que contienen los repositorios a utilizar pueden revisarse en el Anexo B.

4.2 Diseño conceptual del almacén de datos

Se utilizó como base para construir el modelo conceptual, de la estructura del almacén de datos, la ontología global MCPS-R (Cancino, 2020), que abarca el núcleo de Essence (OMG, 2015) y se construyó mediante el uso de la metodología Methontology. La descripción de la metodología Methontology para construir ontologías se incluye en el Anexo C. Para realizar el modelo conceptual para la presente investigación se utilizaron las tareas de conceptualización de la metodología Methontology, las cuales se dan a conocer a continuación:

4.2.1 Tarea1: Construir un glosario de términos

En esta tarea se identifican los conceptos fundamentales del dominio en que se trabaja y que se debe tener en consideración. Para la identificación se utiliza el marco MCPS-R y el núcleo de Essence compuesto por alfas y competencias, para facilitar la comparación de los términos con los repositorios FLOSSmole a utilizar fue necesario realizar una traducción de los términos al idioma inglés ya que los repositorios se encuentran en dicho idioma. A continuación, se dará a conocer el glosario de términos en la Tabla 4.2.

Tabla 4.2. Glosario de términos

Fuente: creación propia.

Nombre	Descripción	Tipo
Essence Kernel	The Essence Kernel captures the essential elements of software engineering, those that are integral to all software engineering methods.	Concepto
Alphas	The Alphas provide descriptions of the kind of things that a team will manage, produce, and use in the process of developing, maintaining, and supporting software and, as such, are relevant to assessing the progress and health of a software endeavor.	Concepto
Competencies	Representations of the key capabilities required to carry out the work of software engineering.	Concepto
MCPS-R	Feature framework of refined software projects.	Concepto
Opportunity	The set of circumstances that makes it appropriate to develop or change a software system.	Concepto

	<p>The opportunity articulates the reason for the creation of the new, or changed, software system. It represents the team's shared understanding of the stakeholders' needs, and helps shape the requirements for the new software system by providing justification for its development.</p>	
Requirements	<p>What the software system must do to address the opportunity and satisfy the stakeholders.</p> <p>It is important to discover what is needed from the software system, share this understanding among the stakeholders and the team members, and use it to drive the development and testing of the new system.</p>	Concepto
Software system	<p>A system made up of software, hardware, and data that provides its primary value by the execution of the software.</p> <p>The primary product of any software engineering endeavor, a software system can be part of a larger software, hardware, or business solution</p>	Concepto
Stakeholders	<p>The people, groups, or organizations who affect or are affected by a software system.</p> <p>The stakeholders provide the opportunity and are the source of the requirements and funding for the software system. The team members are also stakeholders. As much stakeholder involvement as possible throughout a software engineering endeavor is important to support the team and ensure that an acceptable software system is produced.</p>	Concepto
Team	<p>A group of people actively engaged in the development, maintenance, delivery, or support of a specific software system.</p> <p>One, or more, teams plan and perform the work needed to create, update, and/or change the software system.</p>	Concepto
Way of working	<p>The tailored set of practices and tools used by a team to guide and support their work.</p>	Concepto

	The team evolves their way of working alongside their understanding of their mission and their working environment. As their work proceeds they continually reflect on their way of working and adapt it as necessary to their current context.	
Work	Activity involving mental or physical effort done in order to achieve a result. In the context of software engineering, work is everything that the team does to meet the goals of producing a software system matching the requirements, and addressing the opportunity, presented by the stakeholders. The work is guided by the practices that make up the team's way-of-working.	Concepto
Configurable factors	Configurable factors are identified as, those that are required to analyze and make decisions in the development of a software.	Concepto
Measurable factors in software projects	Measurable factors allow you to quantify projects in terms of size, cost, complexity, and development time.	Concepto
Sociological factors	These factors correspond to the characteristics of the development team and identify the individual qualities of each developer.	Concepto
Software project type factors	This factor describes the type of project based on its nature, context, classification to which it belongs, criticality and risks that the project can present.	Concepto
Technological factors	The technological factors of the project describe the development platform, application domain, programming language paradigm and tools to be applied.	Concepto
Addressed	A solution has been produced that demonstrably addresses the opportunity. Phase 1 of 6 of the opportunity concept.	Concepto
Benefit accrued	The operational use or sale of the solution is creating tangible benefits. Phase 2 of 6 of the opportunity concept.	Concepto
Identified	A commercial, social or business opportunity has been identified that could be addressed by a software-based solution.	Concepto

	Phase 3 of 6 of the opportunity concept.	
Solution needed	The need for a software-based solution has been confirmed. Phase 4 of 6 of the opportunity concept.	Concepto
Value established	The value of a successful solution has been established. Phase 5 of 6 of the opportunity concept.	Concepto
Viable	It is agreed that a solution can be produced quickly and cheaply enough to successfully address the opportunity. Phase 6 of 6 of the opportunity concept.	Concepto
Acceptable	The requirements describe a system that is acceptable to the stakeholders. Phase 1 of 6 of the requirements concept.	Concepto
addressed	Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders. Phase 2 of 6 of the requirements concept.	Concepto
Bounded	The purpose and theme of the new system are clear. Phase 3 of 6 of the requirements concept.	Concepto
Coherent	The requirements provide a consistent description of the essential characteristics of the new system. Phase 4 of 6 of the requirements concept.	Concepto
Conceived	The need for a new system has been agreed. Phase 5 of 6 of the requirements concept.	Concepto
Fulfilled	The requirements that have been addressed fully satisfy the need for a new system. Phase 6 of 6 of the requirements concept.	Concepto
Architecture selected	An architecture has been selected that addresses the key technical risks and any applicable organizational constraints. Phase 1 of 6 of the software system concept.	Concepto

Demonstrable	An executable version of the sytem is available that demonstrates the architecture is fit for purpose and supports testing. Phase 2 of 6 of the software system concept.	Concepto
Operational	The system is use in a live environment. Phase 3 of 6 of the software system concept.	Concepto
Ready	The system (as a whole) has been accepted for deployment in a live environment. Phase 4 of 6 of the software system concept.	Concepto
Retired	The system is no longer supported. Phase 5 of 6 of the software system concept.	Concepto
Usable	The system is usable and demonstrates all of the quality characteristics required of an operational system. Phase 6 of 6 of the software system concept.	Concepto
In agreement	The stakeholder representatives are in agreement. Phase 1 of 6 of the stakeholders concept	Concepto
Involved	The stakeholder representatives are actively involved in the work and fulfilling their responsibilities. Phase 2 of 6 of the stakeholders concept	Concepto
Recognized	The stakeholders have been identified. Phase 3 of 6 of the stakeholders concept	Concepto
Represented	The mechanisms for involving the stakeholders are agreed and the stakeholder representatives have been appointed. Phase 4 of 6 of the stakeholders concept	Concepto
Satisfied for deployment	The minimal expectations of the stakeholder representatives have been achieved. Phase 5 of 6 of the stakeholders concept	Concepto
Satisfied in use	The system meets or exceeds the minimal stakeholder expectations. Phase 6 of 6 of the stakeholders concept	Concepto

Adjoumed	The team is no longer accountable for carrying out its mission Phase 1 of 5 of the team concept	Concepto
Collaborating	The team members are working together as one unit. Phase 2 of 5 of the team concept	Concepto
Formed	The team has been populated with enough committed people to start the mission. Phase 3 of 5 of the team concept	Concepto
Performing	The team is working effectively and efficiently. Phase 4 of 5 of the team concept	Concepto
Seeded	The team's mission is clear and the know-how needed to grow the team is in place. Phase 5 of 5 of the team concept	Concepto
Foundation established	The key practices, and tools, that form the foundation of the way of working are selected and ready for use. Phase 1 of 6 of the way of working concept	Concepto
In place	All team members are using the way-of-working to accomplish their tasks. Phase 2 of 6 of the way of working concept	Concepto
In use	Some members of the team are using and adapting, the way-of-working. Phase 3 of 6 of the way of working concept	Concepto
Principles established	The principles, and constraints, that shape the way-of-working are established. Phase 4 of 6 of the way of working concept	Concepto
retired	The way-of-working is no longer in use by the team. Phase 5 of 6 of the way of working concept	Concepto
Working well	The way-of-working is working well for the team. Phase 6 of 6 of the way of working concept	Concepto

Closed	All remaining housekeeping tasks have been completed and the work has been officially closed. Phase 1 of 6 of the work concept	Concepto
Concluded	The work to produce the results has been concluded. Phase 2 of 6 of the work concept	Concepto
Initiated	Work has been requested. Phase 3 of 6 of the work concept	Concepto
Prepared	All pre-conditions for starting the work have been met. Phase 4 of 6 of the work concept	Concepto
Started	The work is proceeding. Phase 5 of 6 of the work concept	Concepto
Under control	The work is going well, risks are under control and productivity levels are sufficient to achieve a satisfactory result. Phase 6 of 6 of the work concept	Concepto
Analysis	This competence encapsulates the ability to understand the opportunities, the needs of their stakeholders and transform them into a set of consistent requirements.	Concepto
Development	This competence encapsulates the ability to design and program software systems efficiently, following the rules and regulations agreed by the team.	Concepto
Leadership	This competency enables a person to inspire and motivate a group of people to achieve a successful conclusion to their work and to meet their objectives.	Concepto
Management	This competency encapsulates the ability to coordinate, plan, and track the work done by a team. The management competency is the administrative and organizational ability that enables the right things to be done at the right time to maximize a team's chances of success.	Concepto

Test	This competition encapsulates the possibility of testing a system, verifying that it is usable and that it meets the requirements.	Concepto
Configurables factors	Configurable factors are identified as, those that are required to analyze and make decisions in the development of a software.	Concepto
Project development model	Main model of development process: Cascade, Evolutionary, Incremental, Spiral, Win win, Agile or Other.	Concepto
Set of subprocess	Name of the subprocesses included in the process pattern: launch & strategy, planning, specification of requirements, analysis & design, implementation, testing or post-mortem evaluation.	Concepto
Set of work products to get	Documentation to obtain.	Concepto
Business model	Documentation business model 1 of 26 of the concept set of work product to get.	Concepto
Detailed design of work units	Documentation detailed design of work units 2 of 26 of the concept set of work product to get.	Concepto
Detailed design review and specification of unit tests	Documentation detailed design review and specification of unit tests 3 of 26 of the concept set of work product to get.	Concepto
Direct engineering	Documentation direct engineering 4 of 26 of the concept set of work product to get.	Concepto
Division of work structure	Documentation division of work structure 5 of 26 of the concept set of work product to get.	Concepto
Drive code review	Documentation drive code review 6 of 26 of the concept set of work product to get.	Concepto
HLD software architecture	Documentation HLD software architecture 7 of 26 of the concept set of work product to get.	Concepto
Identification of components	Documentation identification of components 8 of 26 of the concept set of work product to get.	Concepto
Integration test documentation	Integration test documentation 9 of 26 of the concept set of work product to get.	Concepto
Planning	Documentation planning 10 of 26 of the concept set of work product to get.	Concepto
Platform definition and requirements restrictions	Documentation platform definition and requirements restrictions 11 of 26 of the concept set of work product to get.	Concepto
Post-mortem report of the development process	Documentation post-mortem report of the development process 12 of 26 of the concept set of work product to get.	Concepto

Requirements review and system prototype	Documentation requirements review and system prototype 13 of 26 of the concept set of work product to get.	Concepto
Reverse engineering	Documentation reverse engineering 14 of 26 of the concept set of work product to get.	Concepto
Review of software requirements and specification of acceptance tests	Documentation review of software requirements and specification of acceptance tests 15 of 26 of the concept set of work product to get.	Concepto
Software architecture review	Documentation software architecture review 16 of 26 of the concept set of work product to get.	Concepto
Software prototype	Documentation software prototype 17 of 26 of the concept set of work product to get.	Concepto
Software requirements	Documentation software requirements 18 of 26 of the concept set of work product to get.	Concepto
Specification of acceptance tests	Documentation specification of acceptance tests 19 of 26 of the concept set of work product to get.	Concepto
Specification of integration tests	Documentation specification of integration tests 20 of 26 of the concept set of work product to get.	Concepto
System prototype	Documentation system prototype 21 of 26 of the concept set of work product to get.	Concepto
System requirements	Documentation system requirements 22 of 26 of the concept set of work product to get.	Concepto
Test and acceptance documentation	Test and acceptance documentation 23 of 26 of the concept set of work product to get.	Concepto
Unit code	Documentation unit code 24 of 26 of the concept set of work product to get.	Concepto
Unit test documentation	Unit test documentation 25 of 26 of the concept set of work product to get.	Concepto
Unit test specification	Documentation unit test specification 26 of 26 of the concept set of work product to get.	Concepto
Structure of the work team	Name of roles included in the role pattern: team leader, customer interface manager, design administrator, deployment administrator, test administrator, planning administrator, process administrator, quality manager, support manager.	Concepto
Type of software product assembly	Name the types of assembly-part included in the structure pattern: system, subsystem, product, component, module or object.	Concepto
Way to conduct inspections	Type of inspection: formal technique or informal technique. Duration. Average number of identified issues or defects.	Concepto

Average number of issues or defects identified	Average number of issues or defects identified 1 of 3 of the concept way to conduct inspections	Concepto
Duration	Duration 2 of 3 of the concept way to conduct inspections	Concepto
Type of inspection	Type of inspection 3 of 3 of the concept way to conduct inspections	Concepto
Way to perform the development	Choice of collaborative or individual work: individual or collaborative.	Concepto
Way to track and control the process	Periodicity: daily, weekly, biweekly, monthly or other. Schedule: pre-established or variable. Duration.	Concepto
Measurable factors in software projects	Measurable factors allow you to quantify projects in terms of size, cost, complexity, and development time.	Concepto
Cost of the project	Estimated cost of the project (expressed in pesos). Actual cost of the project (expressed in pesos).	Concepto
Actual cost	Part 1 of 2 of the concept cost of project	Concepto
Estimated cost	Part 2 of 2 of the concept cost of project	Concepto
Defects in the project	Total, defects in the final product. Total defects removed in the product before delivery.	Concepto
Total defects inserted in the final product	Part 1 of 2 of the concept defects in the project	Concepto
Total defects removed in the product before delivery	Part 2 of 2 of the concept defects in the project	Concepto
Effort in the project	Estimated effort of project development. Expressed in Minutes / Man. Real effort of the development of the project. Expressed in Minutes / Man.	Concepto
Actual project development effort	Part 1 of 2 of the concept effort in the project	Concepto
Estimated project development effort	Part 2 of 2 of the concept effort in the project	Concepto
Project complexity	Programming complexity: low, medium or high. Complexity in the data: low, medium or high. Complexity of organization: a person, a group of a department, several groups of a department, several groups of several departments, several groups of several sites not co-located or development of subcontracts by third parties.	Concepto
Complexity in the data	Part 1 of 3 of the concept project complexity	Concepto

Organizational complexity	Part 2 of 3 of the concept project complexity	Concepto
Schedulling complexity	Part 3 of 3 of the concept project complexity	Concepto
Project development time	Estimated duration for the development of the project. Expressed in minutes. Real duration time in the development of the project. Expressed in minutes. Development time per work product. Expressed in minutes.	Concepto
Actual duration time	Part 1 of 3 of the concept project development time	Concepto
Development time per work product	Part 2 of 3 of the concept project development time	Concepto
Estimated length of time	Part 3 of 3 of the concept project development time	Concepto
Project size	Actual number of total lines of code (LOC-R) (not counting comments and blank lines) of the system. Estimated amount of total code lines (LOC-E) (not counting comments and blank lines) of the system. Actual number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse. Estimated number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse. Number of actual function points of the final product. Amount of estimated function points of the final product.	Concepto
Actual amount of end-product code lines	Part 1 of 6 of the concept project size	Concepto
Actual amount of total code lines	Part 2 of 6 of the concept project size	Concepto
Estimated amount of total code lines	Part 3 of 6 of the concept project size	Concepto
Estimated quantity of lines of code of the final product	Part 4 of 6 of the concept project size	Concepto
Number of actual function points in the final product	Part 5 of 6 of the concept project size	Concepto
Number of estimated function points of the final product	Part 6 of 6 of the concept project size	Concepto
Sociological factors	These factors correspond to the characteristics of the development team and identify the individual qualities of each developer	Concepto

Experience of the team in the development methodology	Experience in years in the application of the methodology to be measured. Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.	Concepto
Experience in the methodology to develop	Part 1 of 2 of the concept experience of the team in the development methodology .	Concepto
Way to acquire experience in the methodology	Part 2 of 2 of the concept experience of the team in the development methodology .	Concepto
Maximum level of education	Identify the level of education of each developer: technical, engineering or bachelor, master, doctorate or other.	Concepto
Number of developers	Number of full-time participants.	Concepto
Team experience in the development of the project	Experience in years of software development. Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.	Concepto
Experience in software development	Part 1 of 2 of the concept team experience in the development of the project	Concepto
Way to acquire the experience by developer	Part 2 of 2 of the concept team experience in the development of the project	Concepto
Team experience in the programming language	Experience in years of implementations in the programming language to be measured. Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.	Concepto
Experience in programming language implementations	Part 1 of 2 of the concept team experience in the programming language	Concepto
Way to acquire the language experience	Part 2 of 2 of the concept team experience in the programming language	Concepto
Team experience in the project domain	Experience in years of software development. Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.	Concepto
Domain experience	Part 1 of 2 of the concept team experience in the project domain	Concepto
Experience in the domain	Part 2 of 2 of the concept team experience in the project domain	Concepto
Team experience in using support tools	Locate the experience in the use of the support tools: none, low, medium, high or very high.	Concepto

Software project type factors	This factor describes the type of project based on its nature, context, classification to which it belongs, criticality and risks that the project can present.	Concepto
Nature of the project	Identify the nature of the project according to its background: it has background, development of new application, improvement of application, maintenance of application or reengineering.	Concepto
Project domain	Indicate the professional environment where the system will be used: administrative, banking, accounting, education, engineering, finance, medicine, science, sales or other.	Concepto
Project risks	Risks that may exist in the project.	Concepto
Estimation risk level	Identify the level of estimation risk for the project: very low, low, moderate, high or very high.	Concepto
Organizational risk level	Identify the level of organizational risk for the project: very low, low, moderate, high or very high.	Concepto
People risk level	Identify the level of risk of people for the project: very low, low, moderate, high or very high.	Concepto
Possible impact of estimation risk	Identify the impact of estimation risks, should they occur: none, insignificant, tolerable, serious or catastrophic.	Concepto
Possible impact of organizational risk	Identify the impact of organizational risks, if any, insignificant, tolerable, serious or catastrophic.	Concepto
Possible impact of people risk	Identify the impact of the risks of people, in case of happening: none, insignificant, tolerable, serious or catastrophic.	Concepto
Possible impact of requirement risk	Identify the impact of the requirements risks, in case of happening: none, insignificant, tolerable, serious or catastrophic.	Concepto
Possible impact of technological risk	Identify the impact of technology risks, if any: none, insignificant, tolerable, serious or catastrophic.	Concepto
Possible impact of tool risk	Identify the impact of tool risks, should they occur: none, insignificant, tolerable, serious or catastrophic.	Concepto
Requirement risk level	Identify the level of risk of requirement for the project: very low, low, moderate, high or very high.	Concepto
Technological risk level	Identify the level of technology risk for the project: very low, low, moderate, high or very high.	Concepto
Tools risk level	Identify the level of technology risk for the project: very low, low, moderate, high or very high.	Concepto

Software classification	Locate the project in any of the general types: system software, software for commerce, information systems, military software, real-time software, Web application or other.	Concepto
Technological factors	The technological factors of the project describe the development platform, application domain, programming language paradigm and tools to be applied.	Concepto
Database manager	Manipulator names: Acces, Informix, MySQL, Oracle, PostgreSQL, SQL Server, Sybase or other.	Concepto
Modeling tools	Names of the modeling tools: Rational RosE, Enterprise Architect, Altova, Eclipse Modelig framework or other.	Concepto
Operating systems running the application	Names of the operating systems in which the application will run: Apple-Macintosh, Linux, Microsoft Windows, Unix or other.	Concepto
Paradigm of the main language	Paradigm of the main language: procedural, object-oriented, functional, logical, event-driven, imperative, arrangement oriented, stack oriented, aspect oriented, data flow, visual or assembly.	Concepto
Planning tools	Names of planning tools: Rational Rose, Microsoft Office Visio, Together, Microsoft Office Project or other.	Concepto
Programming languages	Names of programming languages used: Java, C ++, C, C #, Cliper, Cobol, JavaScript, Visual Basic, Pascal, Delphi or other.	Concepto
Programming tools	Names of the programming tools: IDE-General, IDE-Programming, NetBeans, Eclipse or other.	Concepto
Project application domain	Name of the technological domain where it applies: web, medical, administrative or engineering.	Concepto
Types of machine to be used	Names of the types of machines in which it will be used: mainframe, mini computer, pc, mobile device or other.	Concepto
Milestone	When presented with an important part of the development work, the teams may want to identify some intermediate milestones to help improve their estimates and plans.	Concepto
Before approval	Explore the possibilities presented by the creation of a new improved system or software. This includes analyzing the opportunity to be addressed and identifying the interested parties.	Concepto
Decision to go live	Verify that the system produced meets the requirements of the interested parties.	Concepto

Usable system available	Build a system by implementing, testing and integrating one or more elements of the system. This includes the correction of errors and unit tests.	Concepto
In development	Shape the system so that it is easy to develop, modify and maintain. This includes the general design.	Concepto
Skinny system available	Understand what the system produced should do.	Concepto
Decision to fund	Ensure the satisfaction of the interested parties.	Concepto
In production	Support the use of the software system in the production environment.	Concepto

El resultado de esta tarea es de utilidad para darnos un panorama general sobre el dominio MCPs-R el cuál se utilizó durante esta investigación. Identificando los conceptos que componen dicho marco y su significado.

4.2.2 Tarea2: Construir una taxonomía de conceptos

Una taxonomía es un tipo de vocabulario controlado en que todos los términos están conectados mediante un modelo estructural (jerárquico, arbóreo, facetado, etc.) y especialmente orientado a los sistemas de navegación, organización y búsqueda de contenidos de los sitios web (Centelles, 2005). De acuerdo con (Abed, Barbato, Duarte, Montes, Hoepers, Stedding, 2007), definen una taxonomía como: “Un proceso científico (o un sistema particular) de categorizar entidades, es decir, de organizarlas en grupos. Estos mismos autores plantean además que la base para el desarrollo de una buena taxonomía son las características taxonómicas, que son las propiedades o atributos de los objetos que se categorizarán, y deberán satisfacer los requisitos. Las características taxonómicas son las siguientes:

- **Objetividad:** Cuando la característica se identifica sobre la base de un conocimiento objetivo exclusivamente.
- **Determinación:** Cuando existe un proceso claro que se pueda seguir para identificar la característica.
- **Reproducibilidad:** Cuando varias personas describen de forma independiente las características de un mismo objeto y coinciden con el valor observado.
- **Mutuamente excluyente:** Cuando la inclusión de un grupo en una categoría lo excluye de la categorización en cualquier otra.
- **Exhaustivos:** Cuando los grupos incluyen todas las posibilidades.
- **Aceptable:** Cuando la característica es lógica e intuitiva de forma que la comunidad acepte las categorías.
- **Útil:** Cuando la característica puede utilizarse para obtener conocimiento. El cumplimiento de los atributos en una taxonomía aplicada a un sitio web favorece la navegabilidad y facilita además acceder al capital intelectual de las organizaciones.

A continuación, se dará a conocer la taxonomía de conceptos para la ontología MCPS-R como se muestra en la Figura 4.1.

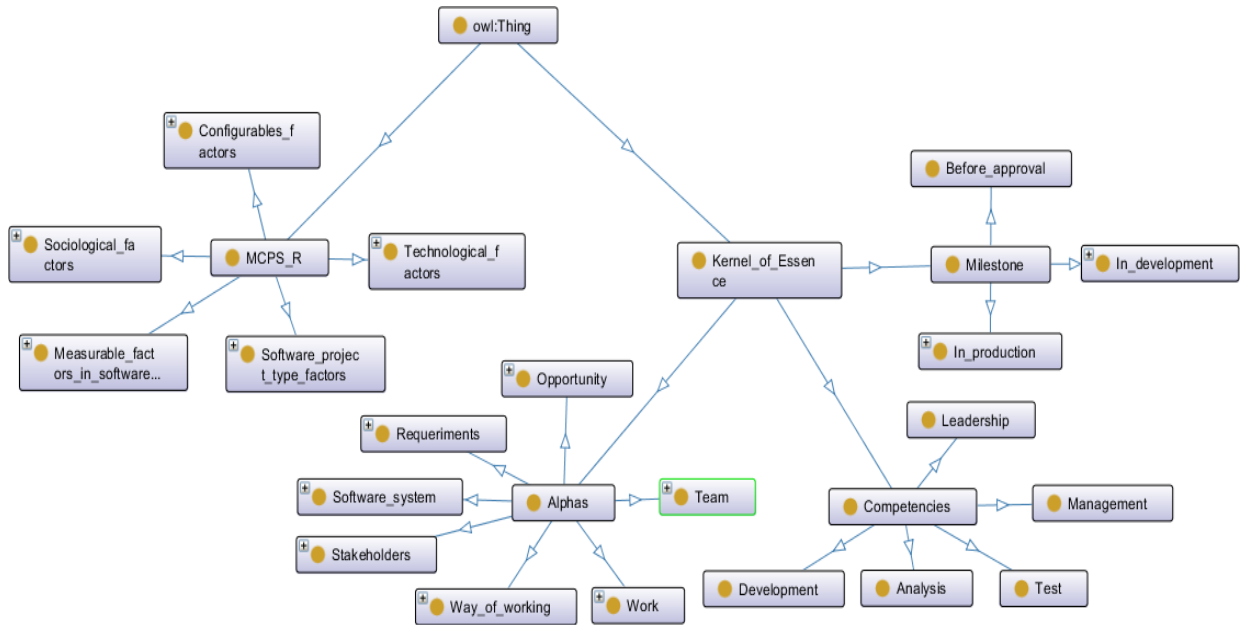


Figura 4.1 Taxonomía de la ontología MCPS-R.
Fuente: creación propia

El resultado de esta taxonomía nos permite identificar la jerarquía general de como se encuentran los conceptos que componen al MCPS-R y el núcleo de Essence.

4.2.3 Tarea3: Construir diagramas de relaciones binarias ad hoc

Ad hoc es una locución latina, que significa literalmente “para esto”. Generalmente se refiere a una solución elaborada específicamente para un problema o fin preciso y, por tanto, no es generalizable ni utilizable para otros propósitos. Se usa para referirse a algo que es adecuado solo para un determinado fin. En sentido amplio, ad hoc puede traducirse como “especifico” o “específicamente”.

Los diagramas Ad-Hoc especifican las relaciones entre los conceptos y clases del dominio de la investigación. Permitiendo ver cuales conceptos se relacionan con el MCPS-R y el núcleo de Essence compuesto por alfas y competencias. A continuación, se da a conocer el diagrama de relaciones binarias de las alfas del núcleo de Essence como se muestra en las Figuras 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8.

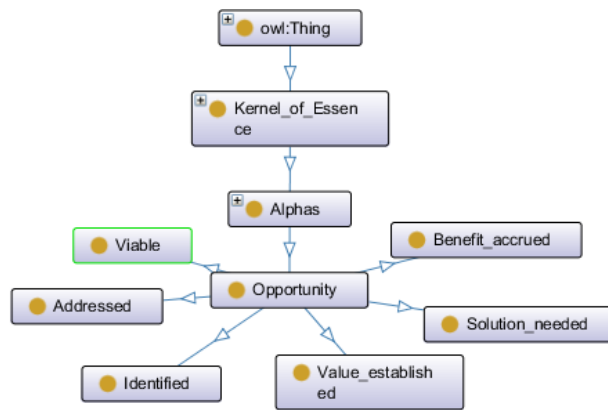


Figura 4.2 Relaciones binarias ad hoc de alfa oportunidad.
Fuente: creación propia

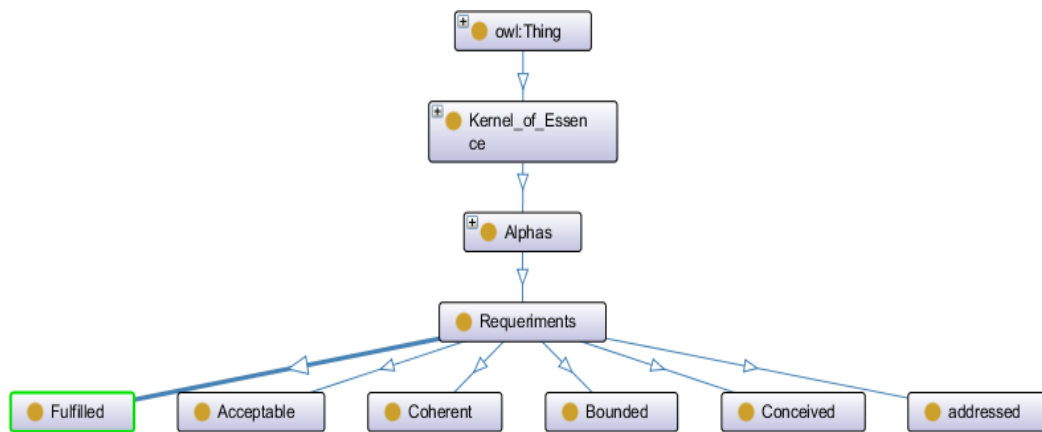


Figura 4.3 Relaciones binarias ad hoc de alfa requerimientos.
Fuente: creación propia

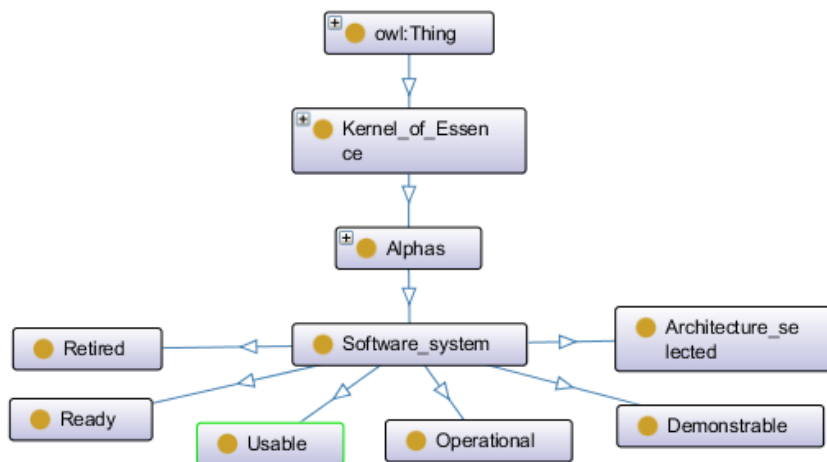


Figura 4.4 Relaciones binarias ad hoc de alfa sistema de software.
Fuente: creación propia

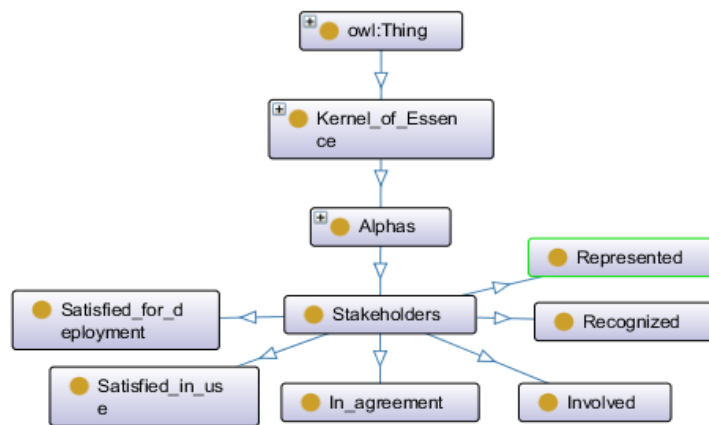


Figura 4.5 Relaciones binarias ad hoc de alfa interesados.
Fuente: creación propia

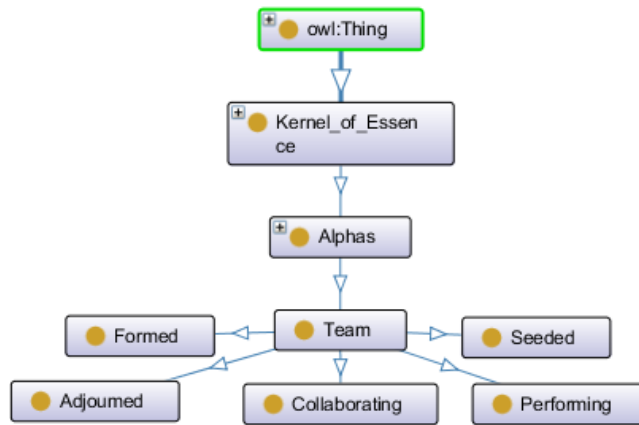


Figura 4.6 Relaciones binarias ad hoc de alfa equipo.
Fuente: creación propia

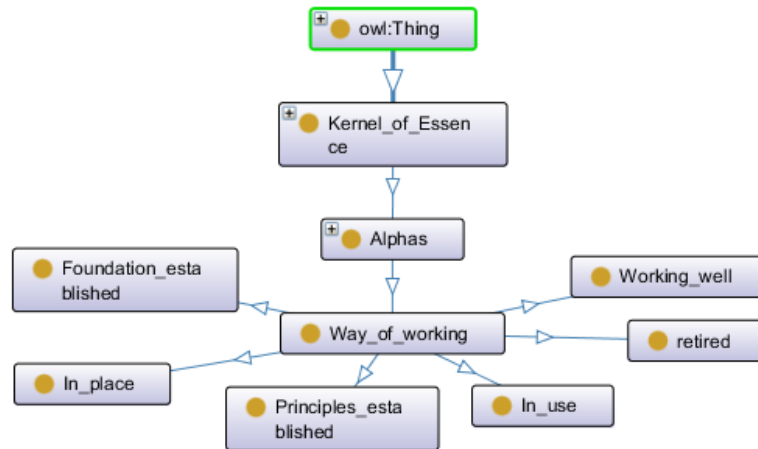


Figura 4.7 Relaciones binarias ad hoc de alfa forma de trabajar.
Fuente: creación propia

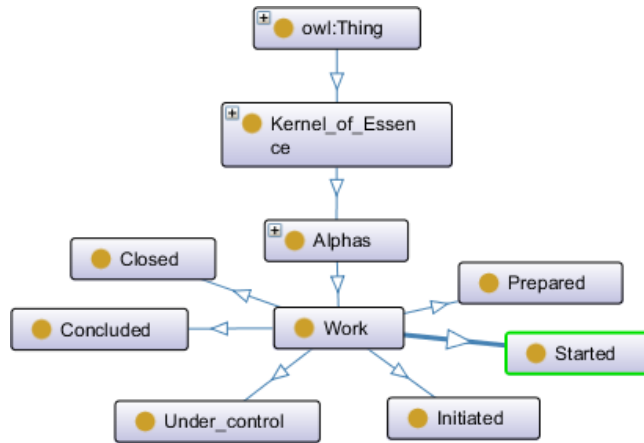


Figura 4.8 Relaciones binarias ad hoc de alfa trabajo.
Fuente: creación propia

A continuación, se da a conocer el diagrama de relaciones binarias de las competencias del núcleo de Essence como se muestra en la Figura 4.9.

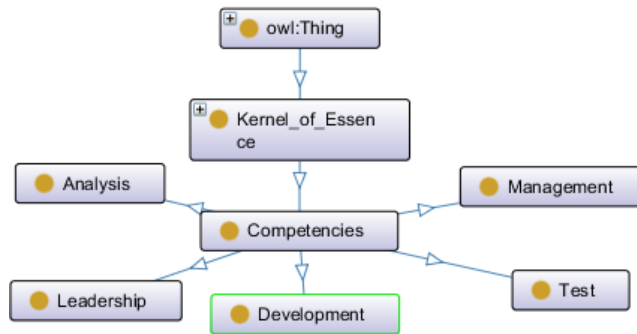


Figura 4.9 Relaciones binarias ad hoc de competencias.
Fuente: creación propia

A continuación, se da a conocer el diagrama de relaciones binarias del MCPS-R como se muestra en las Figuras 4.10, 4.11, 4.12, 4.13 y 4.14.

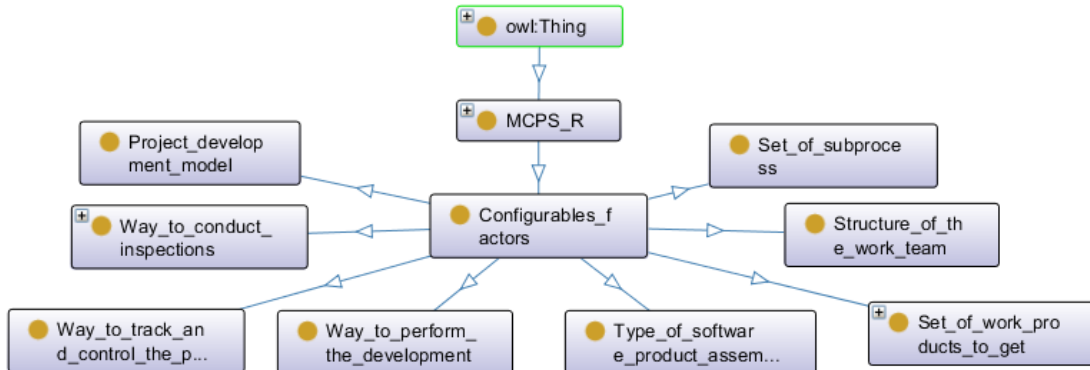


Figura 4.10 Relaciones binarias ad hoc de factores configurables.
Fuente: creación propia

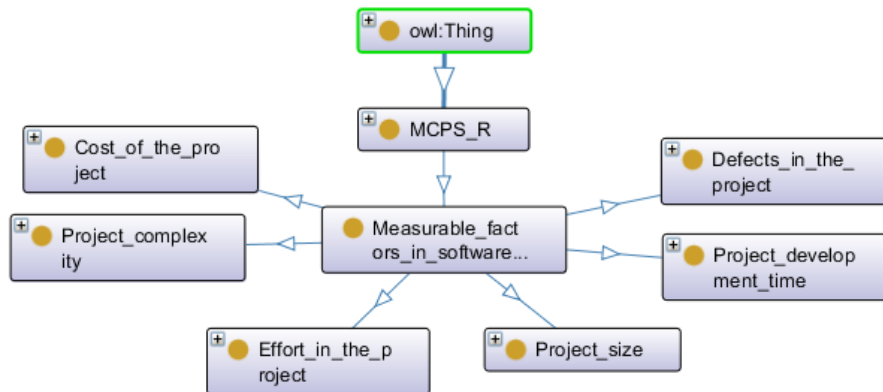


Figura 4.11 Relaciones binarias ad hoc de factores medibles en proyectos de software.
Fuente: creación propia

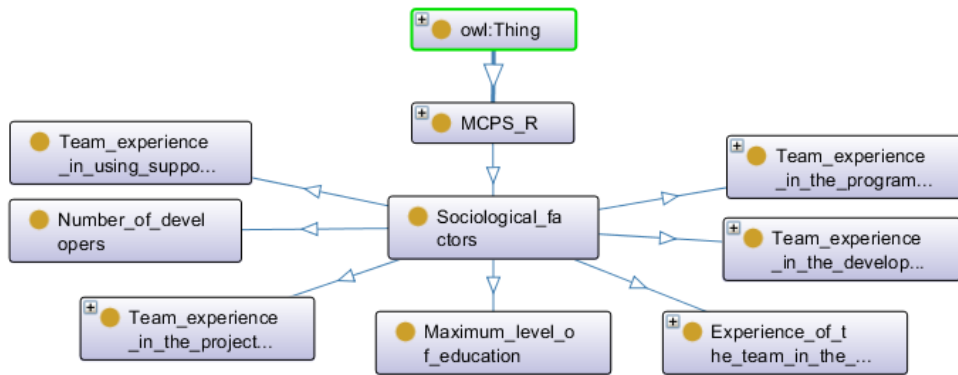


Figura 4.12 Relaciones binarias ad hoc de factores sociológicos.
Fuente: creación propia

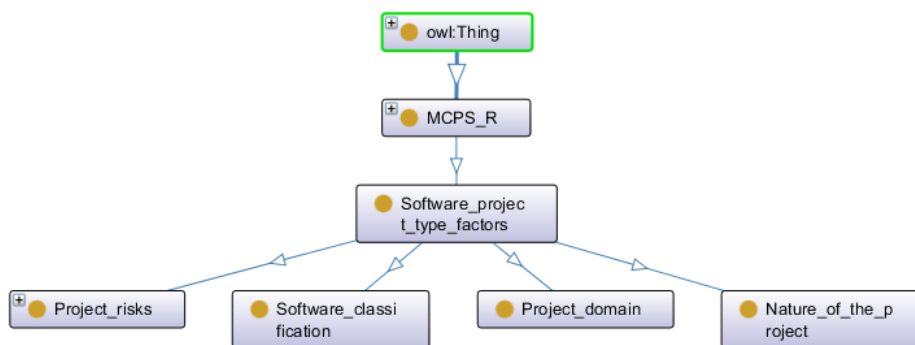


Figura 4.13 Relaciones binarias ad hoc de factores del tipo de proyecto de software.
Fuente: creación propia

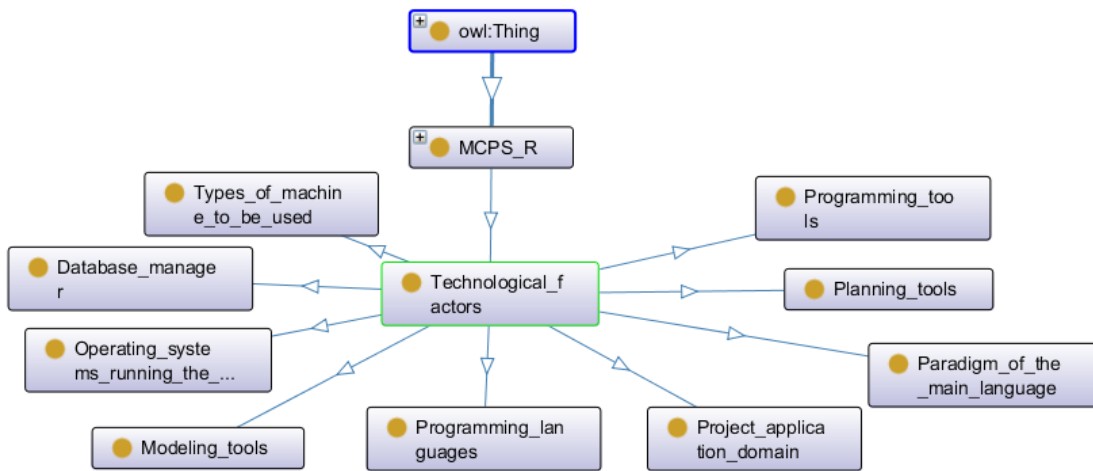


Figura 4.14 Relaciones binarias ad hoc de factores tecnológicos.

Fuente: creación propia

A continuación, se da a conocer el diagrama de relaciones binarias de los milestone del núcleo de Essence como se muestra en la Figura 4.15.

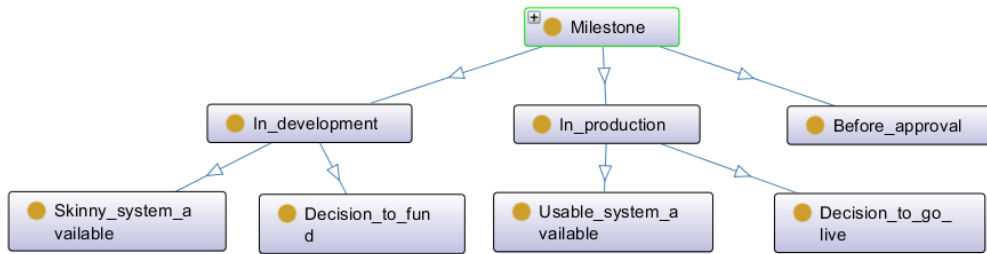


Figura 4.15 Relaciones binarias ad hoc de Milestone.

Fuente: creación propia

4.2.4 Tarea4: Construir el diccionario de conceptos

En este diccionario se incluyen todos los conceptos del vocabulario basado en la ontología, así como los subconceptos. A continuación, se muestra el diccionario de conceptos en la Tabla 4.3.

Tabla 4.3. Diccionario de conceptos

Fuente: creación propia

Nombre del concepto	Subconcepto
Kernel of Essence	<ul style="list-style-type: none"> Alphas Competencies
MCPS-R	<ul style="list-style-type: none"> Configurables factors Measurable factors in software projects Sociological factors Software project type factors

	<ul style="list-style-type: none"> • Technological factors
Alphas	<ul style="list-style-type: none"> • Opportunity • Requirements • Software system • Stakeholders • Team • Way of working • Work
Opportunity	<ul style="list-style-type: none"> • Addressed • Benefit accrued • Identified • Solution needed • Value established • Viable
Requeriments	<ul style="list-style-type: none"> • Acceptable • addressed • Bounded • Coherent • Conceived • Fulfilled
Software sytem	<ul style="list-style-type: none"> • Architecture_selected • Demonstrable • Operational • Ready • Retired • Usable
Stakeholders	<ul style="list-style-type: none"> • In agreement • Involved • Recognized • Represented • Satisfied for deployment • Satisfied in use
Team	<ul style="list-style-type: none"> • Adjoined • Collaborating • Formed • Performing • Seeded

Way of working	<ul style="list-style-type: none"> • Foundation established • In place • In use • Principles established • retired • Working well
Work	<ul style="list-style-type: none"> • Closed • Concluded • Initiated • Prepared • Started • Under control
Competencies	<ul style="list-style-type: none"> • Analysis • Development • Leadership • Management • Test
Configurables factors	<ul style="list-style-type: none"> • Project development model • Set of subprocess • Set of work products to get • Structure of the work team • Type of software product assembly • Way to conduct inspections • Way to perform the development • Way to track and control the process
Set of work products to get	<ul style="list-style-type: none"> • Business model • Detailed design of work units • Detailed design review and specification of unit tests • Direct engineering • Division of work structure • Drive code review • HLD software architecture • Identification of components • Integration test documentation • Planning • Platform definition and requirements restrictions

	<ul style="list-style-type: none"> • Post-mortem report of the development process • Requirements review and system prototype • Reverse engineering • Review of software requirements and specification of acceptance tests • Software architecture review • Software prototype • Software requirements • Specification of acceptance tests • Specification of integration tests • System prototype • System requirements • Test and acceptance documentation • Unit code • Unit test documentation • Unit test specification
Way to conduct inspections	<ul style="list-style-type: none"> • Average number of issues or defects identified • Duration • Type of inspection
Measurable factors in software projects	<ul style="list-style-type: none"> • Cost of the project • Defects in the project • Effort in the project • Project complexity • Project development time • Project size
Cost of the project	<ul style="list-style-type: none"> • Actual cost • Estimated cost
Defects in the project	<ul style="list-style-type: none"> • Total defects inserted in the final product • Total defects removed in the product before delivery
Effort in the project	<ul style="list-style-type: none"> • Actual project development effort • Estimated project development effort
Project complexity	<ul style="list-style-type: none"> • Complexity in the data • Organizational complexity • Scheduling complexity

Project development time	<ul style="list-style-type: none"> • Actual duration time • Development time per work product • Estimated length of time
Project size	<ul style="list-style-type: none"> • Actual amount of end-product code lines • Actual amount of total code lines • Estimated amount of total code lines • Estimated quantity of lines of code of the final product • Number of actual function points in the final product • Number of estimated function points of the final product
Sociological factors	<ul style="list-style-type: none"> • Experience of the team in the development methodology • Maximum level of education • Number of developers • Team experience in the development of the project • Team experience in the programming language • Team experience in the project domain • Team experience in using support tools
Experience of the team in the development methodology	<ul style="list-style-type: none"> • Experience in the methodology to develop • Way to acquire experience in the methodology
Team experience in the development of the project	<ul style="list-style-type: none"> • Experience in software development • Way to acquire the experience by developer
Team experience in the programming language	<ul style="list-style-type: none"> • Experience in programming language implementations • Way to acquire the language experience
Team experience in the project domain	<ul style="list-style-type: none"> • Domain experience • Experience in the domain
Software project type factors	<ul style="list-style-type: none"> • Nature of the project • Project domain • Project risks • Software classification
Project risks	<ul style="list-style-type: none"> • Estimation risk level

	<ul style="list-style-type: none"> • Organizational risk level • People risk level • Possible impact of estimation risk • Possible impact of organizational risk • Possible impact of people risk • Possible impact of requeriment risk • Possible impact of technological risk • Possible impact of tool risk • Requeriment risk level • Technological risk level • Tools risk level
Technological factors	<ul style="list-style-type: none"> • Database manager • Modeling tools • Operating systems running the application • Paradigm of the main language • Planning tools • Programming languages • Programming tools • Project application domain • Types of machine to be used
Milestone	<ul style="list-style-type: none"> • Before approval • In development • In production
In development	<ul style="list-style-type: none"> • Decision to fund • Skinny system available
In production	<ul style="list-style-type: none"> • Usable system available • Decision to go live

4.2.5 Tarea5: Describir en detalle las relaciones binarias

Para cada relación binaria se debe especificar: nombre, nombre de sus conceptos origen y destino, cardinalidad y relación inversa, si existe. En esta tarea se detalla las relaciones existentes en el vocabulario ontológico como se muestra en la Tabla 4.4.

Tabla 4.4. Relaciones binarias Ontología MCPS-R
Fuente: creación propia

Nombre Relación	Concepto Origen	Concepto destino	Cardinalidad	Relación inversa
incluye	Factor (Nota: los factores son 5)	MCPS-R	(1,N)	integra
es descrito por	Alfa (Nota: 7 alfas)	Estado del alfa	(1,N)	es etapa de
habilitan	Estado del alfa	MCPS-R	(1,N)	verifican
desarrolla	Competencia	MCPS-R	(N,N)	es realizada

4.2.6 Formalización

Para formalizar la ontología se utilizó Protégé, un editor para la construcción de la ontología, con el fin de generar su respectiva estructura. Protégé es un software libre de código abierto implementado en java, desarrollado en la Universidad de Stanford, que permite la construcción de ontologías. Es capaz de operar como una plataforma para acceder a otros sistemas basados en conocimiento o aplicaciones integradas, o como una librería que puede ser usada por otras aplicaciones para acceder y visualizar bases de conocimiento. Esta herramienta ofrece una interfaz gráfica que permite al desarrollador de ontologías enfocarse en el modelado conceptual sin que requiera de conocimientos de la sintaxis de los lenguajes de salida, que se almacenan permanentemente.

A continuación, se muestra en la Figura 4.16 y 4.17 la ontología MCPS-R creada para utilizarse como modelo conceptual del almacén de datos. En la Figura 4.16 se muestra la jerarquía de clases de como esta compuesta la ontología MCPS-R. En la Figura 4.17 se muestra la taxonomía de la ontología MCPS-R, permitiendo la navegación visual e interactiva de las relaciones en la ontología.

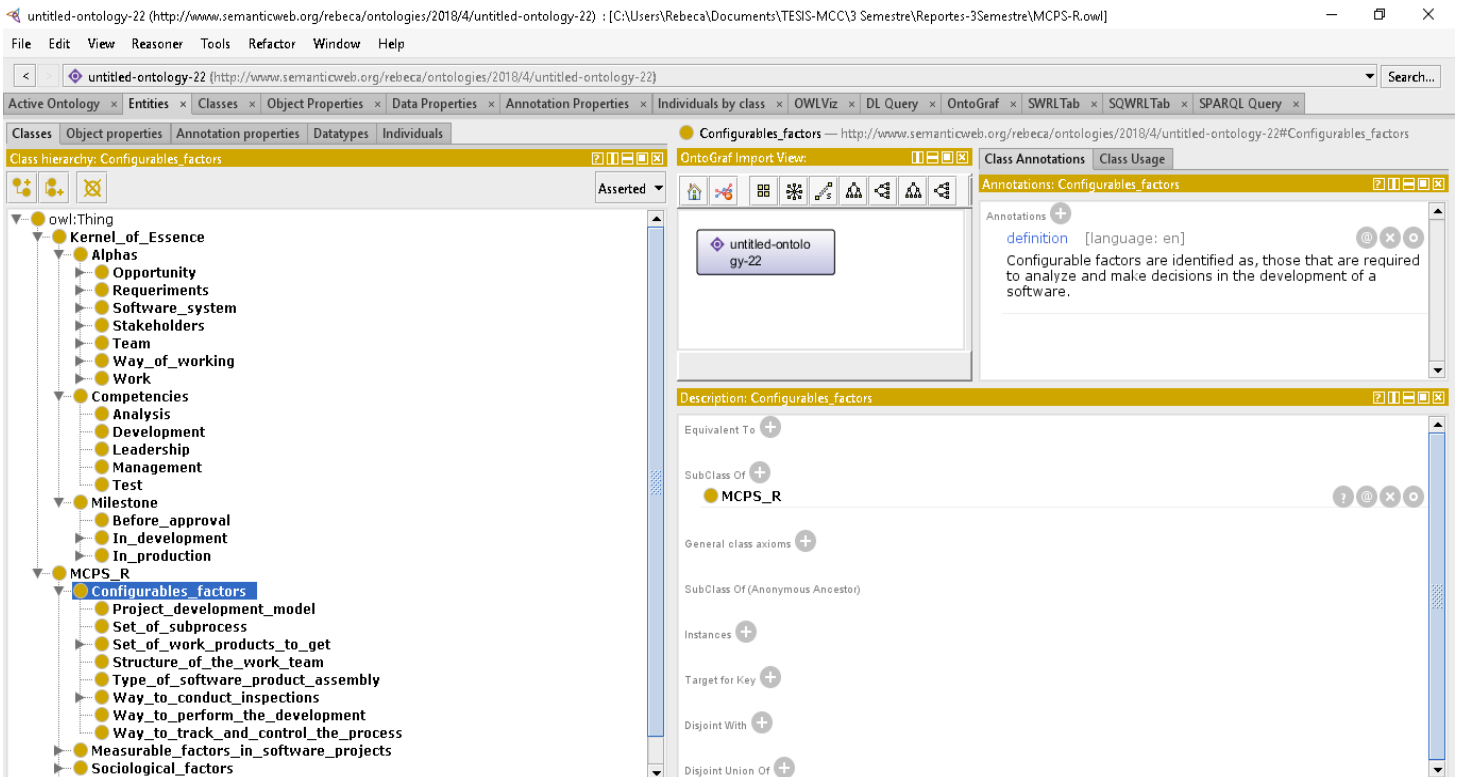


Figura 4.16 Ontología MCPS-R creada en Protegé.
Fuente: creación propia

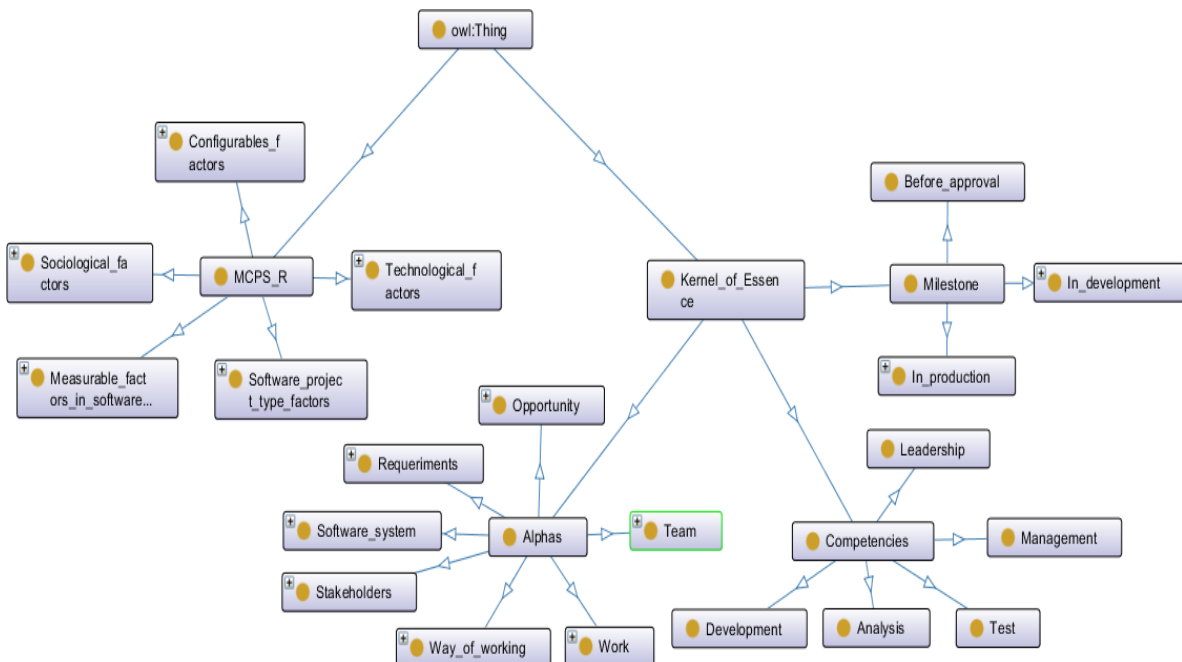


Figura 4.17 Taxonomía de la ontología MCPS-R
Fuente: creación propia

El Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R) contiene conjuntos de características presentes en el desarrollo de un proyecto de software. Los conjuntos de características se conocen también como factores: Factores del tipo de proyecto de software, Factores medibles, Factores sociológicos, Factores tecnológicos y Factores configurables. Así, para aclarar esta clasificación cada factor es un conjunto de características (o sub-factores).

En la figura vemos también, que el núcleo de Essence se conforma por alfas, competencias y contadores (*milestone*). Las alfas representan las cosas esenciales para trabajar, es decir la clase de cosas que un equipo debe gestionar, usar y producir en cualquier esfuerzo de desarrollo de software. Las competencias representan las habilidades clave que se requieren para el desarrollo de software y los contadores (*milestone*) son hitos para ayudar a mejorar estimaciones y planes en el trabajo de desarrollo de software.

4.3 Diseño lógico del almacén de datos

Para la construcción del almacén de datos fue necesario identificar el diseño de base de datos más apropiado, para lo cual se realizó una comparación sobre las ventajas y desventajas de las bases de datos relacionales y bases de datos basadas en ontologías, que se muestra en el Anexo D.

A continuación, en la Figura 4.18 se presenta el modelo lógico (esquema general del modelo entidad-relación) de la base de datos que tiene como nombre *dw_manager* que se realizó con la herramienta MySQL Workbench.

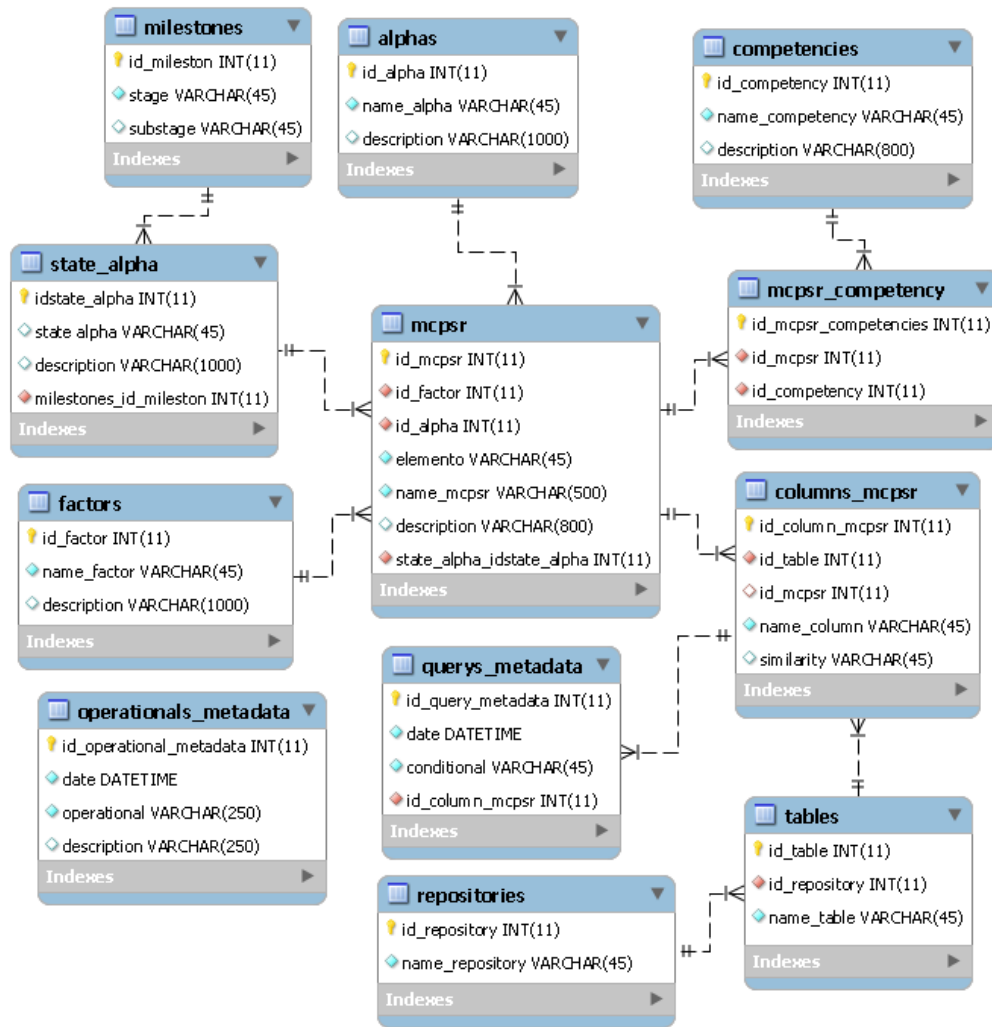


Figura 4.18 Esquema del modelo lógico dw_manager.
Fuente: creación propia

A continuación, se procede a definir cada tabla del esquema mostrado en la Figura 4.18.

4.3.1 factors

La tabla “factors” de la base de datos *dw_manager* es descrita en la Tabla 4.5.

Tabla 4.5. Descripción de la tabla “factors” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_factor	int (11)	PK	No	Si	Identificador para diferenciar el tipo de factor de agrupación de los conceptos del MCPS-R

name_factor	varchar (45)		No		Nombre del tipo de factor de agrupación de los conceptos del MCPS-R
description	varchar (45)		Si		Breve descripción del tipo de factor de agrupación

4.3.2 alphas

La tabla “alphas” de la base de datos *dw_manager* es descrita en la Tabla 4.6.

Tabla 4.6. Descripción de la tabla “alphas” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_alpha	int (11)	PK	No	Si	Identificador para diferenciar el tipo de alfa (que maneja el estándar Essence) para agrupar los conceptos del MCPS-R
name_alpha	varchar (45)		No		Nombre del tipo de alfa para agrupar los conceptos del MCPS-R
description	varchar (1000)		Si		Breve descripción del alfa

4.3.3 milestones

La tabla “milestones” de la base de datos *dw_manager* es descrita en la Tabla 4.7.

Tabla 4.7. Descripción de la tabla “milestones” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_mileston	int (11)	PK	No	Si	Identificador para diferenciar el hito del proyecto (que maneja el estándar Essence)
stage	varchar (45)		No		Nombre de la fase del hito del proyecto
substage	varchar (45)		Si		Nombre de la sub fase del hito del proyecto, si es que tiene alguno

4.3.4 mcpsr

La tabla “mcpsr” de la base de datos *dw_manager* es descrita en la Tabla 4.8.

Tabla 4.8. Descripción de la tabla “mcpsr” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_mcpsr	int (11)	PK	No	Si	Identificador para diferenciar a cada concepto del MCPS-R
id_factor	int (11)	FK	No		Identificador del factor en el que se agrupa dicho concepto MCPS-R

id_alpha	int (11)	FK	No		Identificador del estado del alfa en el que se agrupa dicho concepto MCPS-R
elemento	varchar (45)		No		Identificador para cada concepto del MCPS-R
name_mcpsr	varchar (500)		No		Nombre del concepto del MCPS-R
description	varchar (800)		Si		Breve descripción del concepto del MCPS-R
state_alpha_idstate_alpha	int (11)	FK	No		Identificador del estado de alpha al que pertenece dicho concepto MCPS-R

4.3.5 competencias

La tabla “competencies” de la base de datos *dw_manager* es descrita en la Tabla 4.9.

Tabla 4.9. Descripción de la tabla “competencies” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_competency	int (11)	PK	No	Si	Identificador para diferenciar a cada competencia (rol en el desarrollo de software) que maneja el estándar Essence
name_competency	varchar (45)		No		Nombre de la competencia que maneja el estándar Essence
description	varchar (800)		Si		Breve descripción de la competencia

4.3.6 mcpsr_competency

La tabla “mcpsr_competency” de la base de datos *dw_manager* es descrita en la Tabla 4.10.

Tabla 4.10. Descripción de la tabla “mcpsr_competency” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_mcpsr_competency	int (11)	PK	No	Si	Identificador para diferenciar a los conceptos del MCPS-R que están a cargo de cada competencia (rol en el desarrollo de software)
id_mcpsr	int (11)	FK	No		Identificador para diferenciar el concepto del MCPS-R
id_competency	int (11)	FK	No		Identificador para diferenciar a cada competencia que maneja el estándar Essence

4.3.7 repositories

La tabla “repositories” de la base de datos *dw_manager* es descrita en la Tabla 4.11.

Tabla 4.11. Descripción de la tabla “repositories” de la base de datos *dw_manager*.

Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_repository	int (11)	PK	No	Si	Identificador para diferenciar a los diferentes repositorios de proyectos que se encuentran en el sitio de FlossMole
name_repository	varchar (45)		No		Nombre del repositorio de proyectos de software

4.3.8 tables

La tabla “tables” de la base de datos *dw_manager* es descrita en la Tabla 4.12.

Tabla 4.12. Descripción de la tabla “tables” de la base de datos *dw_manager*.

Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_table	int (11)	PK	No	Si	Identificador para diferenciar a las diferentes tablas de cada repositorio de proyecto de software
id_repository	int (11)	FK	No		Identificador para diferenciar a los diferentes repositorios de proyectos que se encuentran en el sitio de FlossMole
name_table	varchar (45)				Nombre de la tabla del repositorio

4.3.9 columns_mcpsr

La tabla “columns_mcpsr” de la base de datos *dw_manager* es descrita en la Tabla 4.13.

Tabla 4.13. Descripción de la tabla “columns_mcpsr” de la base de datos *dw_manager*.

Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_column_mcpsr	int (11)	PK	No	Si	Identificador para diferenciar las concordancias entre los conceptos del MCPSR y las columnas de las tablas de cada repositorio de proyectos de software que se encuentran en el sitio de FlossMole
id_table	int (11)	FK	No		Identificador para diferenciar a las diferentes tablas de cada repositorio de proyecto de software

id_mcpsr	int (11)	FK	Si		Identificador para diferenciar el concepto del MCPS-R
name_column	varchar (45)		No		Nombre de la columna según la tabla y su repositorio repositorio
similarity	varchar (45)		Si		Nivel de concordancia al usar el algoritmo que permita comparar el MCPS-R y las columnas de los repositorios

4.3.10 *querys_metadata*

La tabla “*querys_metadata*” de la base de datos *dw_manager* es descrita en la Tabla 4.14.

Tabla 4.14. Descripción de la tabla “*querys_metadata*” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_query_metadata	int (11)	PK	No	Si	Identificador para diferenciar cada una de las consultas que se realicen, generando datos históricos de búsquedas
date	datetime		No		Fecha y hora en la que se realizó la consulta
conditional	varchar (45)		Si		Especificaciones usadas para las consultas generales
id_column_mcpsr	int (11)	FK	No		Identificador para <i>columns_mcpsr</i>

4.3.11 *operationals_metadata*

La tabla “*operationals_metadata*” de la base de datos *dw_manager* es descrita en la Tabla 4.15.

Tabla 4.15. Descripción de la tabla “*operationals_metadata*” de la base de datos *dw_manager*.
Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
id_operational_metadata	int (11)	PK	No	Si	Identificador para diferenciar cada una de las operaciones que realice el usuario final en el sistema DW Manager
date	datetime		No		Fecha y hora en la que se realizó la funcionalidad
operational	varchar (250)		No		Especifica el tipo de operación realizada con el sistema: carga, respaldo, consulta, etc.
description	varchar (250)		Si		Una breve descripción de la operación realizada en el sistema

4.3.12 state_alpha

La tabla “state_alpha” de la base de datos *dw_manager* es descrita en la Tabla 4.16.

Tabla 4.16. Descripción de la tabla “state_alpha” de la base de datos *dw_manager*.

Fuente: creación propia.

Columna	Tipo (longitud)	Llave	Nulo	AI	Descripción
idstate_alpha	int (11)	PK	No	Si	Identificador para diferenciar el tipo de estado de alfa (que maneja el estándar Essence)
state_alpha	varchar(45)		No		Nombre del estado de alfa
description	varchar (1000)		No		Descripción del estado del alfa
milestones_id_mileston	int (11)	FK	No		Identificador para el milestone al que pertenece

4.3.13 Relaciones

A continuación, en la Tabla 4.17 se procede a definir las relaciones entre las tablas del esquema mostrado en la Figura 4.18. Donde en la primera columna se describe la tabla y la columna con la llave primaria (PK), separadas por un guión bajo; en la segunda columna se describe la cardinalidad entre la tabla con la llave primaria y la tabla con la llave foránea, con la sintaxis PK:FK; y en la tercera columna se describe la tabla y la columna con la llave foránea (FK).

Tabla 4.17. Descripción de relaciones entre las tablas de la base de datos *dw_manager*.

Fuente: creación propia.

tabla.columna (pk)	cardinalidad	tabla.columna (fk)
alphas.id_alpha	1:N	mcpsr.id_alpha
factors.id_factor	1:N	mcpsr.id_factor
mcpsr.id_mcpsr	1:N	mcpsr_competency.id_mcpsr
competencies.id_competency	1:N	mcpsr_competency.id_competency
repositories.id_repository	1:N	tables.id_repository
tables.id_table	1:N	columns_mcpsr.id_table
mcpsr.id_mcpsr	1:N	columns_mcpsr.id_mcpsr
columns_mcpsr.id_column_mcpsr	1:N	querys_metadata.id_column_mcpsr
milestones.id_mileston	1:N	state_alpha.milestones_id_milestone
state_alpha.idstate_alpha	1:N	mcpsr.state_alpha_idstate_alpha

4.4 Diseño físico del almacén de datos

A continuación, en la Tabla 4.18 se muestra el script SQL que pertenece al diseño de la base de datos *dw_manager* mostrada en la Figura 4.18 de la sección 4.3. En la cual se describe cada una de las tablas a detalle, y se ejecutó en el manejador de base de datos MySQL.

Tabla 4.18. Script SQL obtenido al transformar el modelo de la Figura 4.18.

Fuente: creación propia.

```

CREATE DATABASE IF NOT EXISTS `dw_manager`
/*!40100 DEFAULT CHARACTER SET utf8 */;
USE `dw_manager`;
-- MySQL dump 10.13 Distrib 5.7.12, for Win64
(x86_64)
-- Host: localhost Database: dw_manager
-----
-- Server version 5.7.17-log
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLI
ENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RE
SULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECT
ION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET
@OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES,
SQL_NOTES=0 */;
-- Table structure for table `alphas`
DROP TABLE IF EXISTS `alphas`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `alphas` (
  `id_alpha` int(11) NOT NULL AUTO_INCREMENT,
  `name_alpha` varchar(45) CHARACTER SET utf8
NOT NULL,
  `description` varchar(1000) CHARACTER SET
utf8 DEFAULT NULL,
  PRIMARY KEY (`id_alpha`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `alphas`
LOCK TABLES `alphas` WRITE;
/*!40000 ALTER TABLE `alphas` DISABLE KEYS */;
INSERT INTO `alphas` VALUES
(1,'Opportunity','The set of circumstances
that makes it appropriate to develop or change
a software system.The opportunity articulates
the reason for the creation of the new, or
changed, software system.It represents the
team's shared understanding of the
stakeholders'needs, and helps shape the
requirements for the new software system by
providing justification for its
development. '),(2,'Requirements','What the
software system must do to address the
opportunity and satisfy the stakeholders.It is

```

```

important to discover what is needed from the
software system, share this understanding
among the stakeholders and the team members,
and use it to drive the development and testing
of the new system. '),(3,'Software system','A
system made up of software, hardware, and data
that provides its primary value by the
execution of the software. The primary product
of any software engineering endeavor, a
software system can be part of a larger
software, hardware, or business
solution. '),(4,'Stakeholders','The people,
groups, or organizations who affect or are
affected by a software system.The people,
groups, or organizations who affect or are
affected by a software system.The stakeholders
provide the opportunity and are the source of
the requirements and funding for the software
system.The team members are also stakeholders.
As much stakeholder involvement as possible
throughout a software engineering endeavor is
important to support the team and ensure that
an acceptable software system is
produced. '),(5,'Team','A group of people
actively engaged in the development,
maintenance, delivery, or support of a
specific software system.One, or more, teams
plan and perform the work needed to create,
update, and/or change the software
system. '),(6,'Way of working','The tailored
set of practices and tools used by a team to
guide and support their work.The team evolves
their way of working alongside their
understanding of their mission and their
working environment. As their work proceeds
they continually reflect on their way of
working and adapt it as necessary to their
current context. '),(7,'Work ', 'Activity
involving mental or physical effort done in
order to achieve a result.In the context of
software engineering, work is everything that
the team does to meet the goals of producing
a software system matching the requirements,
and addressing the opportunity, presented by
the stakeholders. The work is guided by the
practices that make up the team's way-of-
working. ');
/*!40000 ALTER TABLE `alphas` ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table `columns_mcpsr`
DROP TABLE IF EXISTS `columns_mcpsr`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `columns_mcpsr` (
  `id_column_mcpsr` int(11) NOT NULL
AUTO_INCREMENT,
  `id_table` int(11) NOT NULL,
  `id_mcpsr` int(11) DEFAULT NULL,
  `name_column` varchar(45) CHARACTER SET utf8
NOT NULL,
  `similarity` varchar(45) CHARACTER SET utf8
DEFAULT NULL,

```

```

PRIMARY KEY (`id_column_mcpsr`),
KEY `id_table_idx` (`id_table`),
KEY `id_mcpsr_idx` (`id_mcpsr`),
CONSTRAINT `FK_id_mcpsr_column` FOREIGN KEY
(`id_mcpsr`) REFERENCES `mcpsr` (`id_mcpsr`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `FK_id_table` FOREIGN KEY
(`id_table`) REFERENCES `tables` (`id_table`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `columns_mcpsr`
LOCK TABLES `columns_mcpsr` WRITE;
/*!40000 ALTER TABLE `columns_mcpsr` DISABLE
KEYS */;
/*!40000 ALTER TABLE `columns_mcpsr` ENABLE
KEYS */;
UNLOCK TABLES;
-- Table structure for table `competencies`
DROP TABLE IF EXISTS `competencies`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `competencies` (
`id_competency` int(11) NOT NULL
AUTO_INCREMENT,
`name_competency` varchar(45) CHARACTER SET
utf8 NOT NULL,
`description` varchar(800) CHARACTER SET
utf8 DEFAULT NULL,
PRIMARY KEY (`id_competency`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `competencies`
LOCK TABLES `competencies` WRITE;
/*!40000 ALTER TABLE `competencies` DISABLE
KEYS */;
INSERT INTO `competencies` VALUES
(1,'Analysis','This competence encapsulates
the ability to understand the opportunities,
the needs of their stakeholders and transform
them into a set of consistent
requirements. '),(2,'Development','This
competence encapsulates the ability to design
and program software systems efficiently,
following the rules and regulations agreed by
the team. '),(3,'Leadership','This competency
enables a person to inspire and motivate a
group of people to achieve a successful
conclusion to their work and to meet their
objectives. '),(4,'Management','This
competency encapsulates the ability to
coordinate, plan, and track the work done by
a team.The management competency is the
administrative and organizational ability that
enables the right things to be done at the
right time to maximize a team's chances of
success. '),(5,'Test','This competition
encapsulates the possibility of testing a
system, verifying that it is usable and that
it meets the requirements. ');

```

```

/*!40000 ALTER TABLE `competencies` ENABLE
KEYS */;
UNLOCK TABLES;
-- Table structure for table `factors`
DROP TABLE IF EXISTS `factors`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `factors` (
`id_factor` int(11) NOT NULL AUTO_INCREMENT,
`name_factor` varchar(45) CHARACTER SET utf8
NOT NULL,
`description` varchar(1000) CHARACTER SET
utf8 DEFAULT NULL,
PRIMARY KEY (`id_factor`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `factors`
LOCK TABLES `factors` WRITE;
/*!40000 ALTER TABLE `factors` DISABLE KEYS
*/;
INSERT INTO `factors` VALUES (1,'Configurables
factors','Configurable factors are identified
as, those that are required to analyze and make
decisions in the development of a
software. '),(2,'Measurable_factors_in
software_projects','Measurable factors allow
you to quantify projects in terms of size,
cost, complexity, and development
time. '),(3,'Sociological_factors','These
factors correspond to the characteristics of
the development team and identify the
individual qualities of each
developer. '),(4,'Software_project_type
factors','This factor describes the type of
project based on its nature, context,
classification to which it belongs,
criticality and risks that the project can
present. '),(5,'Technological_factors','The
technological factors of the project describe
the development platform, application domain,
programming language paradigm and tools to be
applied. ');
/*!40000 ALTER TABLE `factors` ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table `mcpsr`
DROP TABLE IF EXISTS `mcpsr`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `mcpsr` (
`id_mcpsr` int(11) NOT NULL AUTO_INCREMENT,
`id_factor` int(11) NOT NULL,
`id_alpha` int(11) NOT NULL,
`elemento` varchar(45) CHARACTER SET utf8
NOT NULL,
`name_mcpsr` varchar(500) CHARACTER SET utf8
NOT NULL,
`description` varchar(800) CHARACTER SET
utf8 DEFAULT NULL,
`state_alpha_idstate_alpha` int(11) NOT
NULL,
PRIMARY KEY (`id_mcpsr`),
KEY `id_alpha_idx` (`id_alpha`),

```

```

KEY `id_factor_idx` (`id_factor`),
KEY `fk_mcpsr_state_alpha1_idx`
(`state_alpha_idstate_alpha`),
CONSTRAINT `FK_id_alpha_mcpsr` FOREIGN KEY
(`id_alpha`) REFERENCES `alphas` (`id_alpha`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `FK_id_factor` FOREIGN KEY
(`id_factor`) REFERENCES `factors`
(`id_factor`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
CONSTRAINT `fk_mcpsr_state_alpha1` FOREIGN
KEY (`state_alpha_idstate_alpha`) REFERENCES
`state_alpha` (`idstate_alpha`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=103 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `mcpsr`
LOCK TABLES `mcpsr` WRITE;
/*!40000 ALTER TABLE `mcpsr` DISABLE KEYS */;
INSERT INTO `mcpsr` VALUES
(1,4,1,'X1','Software classification','Locate
the project in any of the general types: system
software, software for commerce, information
systems, military software, real-time
software, Web application or
other.',3),(2,4,4,'X2','Project
domain','Indicate the professional
environment where the system will be used:
administrative, banking, accounting,
education, engineering, finance, medicine,
science, sales or
other.',21),(3,4,4,'X3','Nature of the
project','Identify the nature of the project
according to its background: it has
background, development of new application,
improvement of application, maintenance of
application or
reengineering.',21),(4,4,1,'X4','Technologica
l risk level','Identify the level of
technology risk for the project: very low,
low, moderate, high or very
high.',6),(5,4,1,'X5','Possible impact of
technological risk','Identify the impact of
technology risks, if any: none, insignificant,
tolerable, serious or
catastrophic.',6),(6,4,1,'X6','People risk
level','Identify the level of risk of people
for the project: very low, low, moderate, high
or very high.',6),(7,4,1,'X7','Possible
impact of people risk','Identify the impact of
the risks of people, in case of happening:
none, insignificant, tolerable, serious or
catastrophic.',6),(8,4,1,'X8','Organizational
risk level','Identify the level of
organizational risk for the project: very low,
low, moderate, high or very
high.',6),(9,4,1,'X9','Possible impact of
organizational risk','Identify the impact of
organizational risks, if any, insignificant,
tolerable, serious or
catastrophic.',6),(10,4,1,'X10','Tools risk
level','Identify the level of technology risk
for the project: very low, low, moderate, high
or very high.',6),(11,4,1,'X11','Possible

```

```

impact of tool risk','Identify the impact of
tool risks, should they occur: none,
insignificant, tolerable, serious or
catastrophic.',6),(12,4,1,'X12','Requeriment
risk level','Identify the level of risk of
requirement for the project: very low, low,
moderate, high or very
high.',6),(13,4,1,'X13','Possible impact of
requeriment risk','Identify the impact of the
requirements risks, in case of happening:
none, insignificant, tolerable, serious or
catastrophic.',6),(14,4,1,'X14','Estimation
risk level','Identify the level of estimation
risk for the project: very low, low, moderate,
high or very
high.',6),(15,4,1,'X15','Possible impact of
estimation risk','Identify the impact of
estimation risks, should they occur: none,
insignificant, tolerable, serious or
catastrophic.',6),(16,2,3,'X16','Actual
amount of total code lines','Actual number of
total lines of code (LOC-R) (not counting
comments and blank lines) of the
system.',16),(17,2,7,'X17','Estimated amount
of total code lines','Estimated amount of
total code lines (LOC-E) (not counting
comments and blank lines) of the
system.',39),(18,2,3,'X18','Actual amount of
end-product code lines','Actual number of
lines of code of the final product of type:
base, deleted, reused, added, modified, new
and changed, and new for
reuse.',16),(19,2,7,'X19','Estimated quantity
of lines of code of the final
product','Estimated number of lines of code of
the final product of type: base, deleted,
reused, added, modified, new and changed, and
new for reuse.',39),(20,2,3,'X20','Actual
amount of end-product code lines','Actual
number of lines of code of the final product
of type: base, deleted, reused, added,
modified, new and changed, and new for
reuse.',16),(21,2,7,'X21','Estimated quantity
of lines of code of the final
product','Estimated number of lines of code of
the final product of type: base, deleted,
reused, added, modified, new and changed, and
new for reuse.',39),(22,2,3,'X22','Actual
amount of end-product code lines','Actual
number of lines of code of the final product
of type: base, deleted, reused, added,
modified, new and changed, and new for
reuse.',16),(23,2,7,'X23','Estimated quantity
of lines of code of the final
product','Estimated number of lines of code of
the final product of type: base, deleted,
reused, added, modified, new and changed, and
new for reuse.',39),(24,2,3,'X24','Actual
amount of end-product code lines','Actual
number of lines of code of the final product
of type: base, deleted, reused, added,
modified, new and changed, and new for
reuse.',16),(25,2,7,'X25','Estimated quantity
of lines of code of the final
product','Estimated number of lines of code of
the final product of type: base, deleted,

```

reused, added, modified, new and changed, and new for reuse.',39),(26,2,3,'X26','Actual amount of end-product code lines','Actual number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',16),(27,2,7,'X27','Estimated quantity of lines of code of the final product','Estimated number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',39),(28,2,3,'X28','Actual amount of end-product code lines','Actual number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',16),(29,2,7,'X29','Estimated quantity of lines of code of the final product','Estimated number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',39),(30,2,3,'X30','Actual amount of end-product code lines','Actual number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',16),(31,2,7,'X31','Estimated quantity of lines of code of the final product','Estimated number of lines of code of the final product of type: base, deleted, reused, added, modified, new and changed, and new for reuse.',39),(32,2,3,'X32','Number of actual function points in the final product','Number of actual function points of the final product.',16),(33,2,7,'X33','Number of estimated function points of the final product','Amount of estimated function points of the final product.',39),(34,2,3,'X34','Total defects inserted in the final product','Total defects in the final product',18),(35,2,3,'X35','Total defects removed in the product before delivery','Total defects removed in the product before delivery.',16),(36,2,7,'X36','Estimated cost','Estimated cost of the project (expressed in pesos).',39),(37,2,1,'X37','Actual cost','Actual cost of the project (expressed in pesos).',5),(38,2,7,'X38','Schedulling complexity','Programming complexity: low, medium or high.',39),(39,2,7,'X39','Complexity in the data','Complexity in the data: low, medium or high.',39),(40,2,4,'X40','Organizational complexity','Complexity of organization: a person, a group of a department, several groups of a department, several groups of several departments, several groups of several sites not co-located or development of subcontracts by third parties.',20),(41,2,7,'X41','Estimated length of time','Estimated duration for the development of the project. Expressed in minutes.',39),(42,2,7,'X42','Actual duration time','Real duration time in the development

of the project. Expressed in minutes.',41),(43,2,7,'X43','Development time per work product','Development time per work product. Expressed in minutes.',40),(44,2,7,'X44','Estimated project development effort','Estimated project development effort',39),(45,2,7,'X45','Actual project development effort','Actual project development effort',39),(46,3,5,'X46','Number of developers','Number of full-time participants.',29),(47,3,5,'X47','Maximum level of education','Identify the level of education of each developer: technical, engineering or bachelor, master, doctorate or other.',27),(48,3,5,'X48','Experience in software development','Experience in years of software development.',27),(49,3,5,'X49','Way to acquire the experience by developer','Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.',27),(50,3,5,'X50','Experience in programming language implementations','Experience in years of implementations in the programming language to be measured.',27),(51,3,5,'X51','Way to acquire the language experience','Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.',27),(52,3,5,'X52','Experience in the methodology to develop','Experience in years in the application of the methodology to be measured.',27),(53,3,5,'X53','Way to acquire experience in the methodology','Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.',27),(54,3,5,'X54','Domain experience','Experience in years of software development.',27),(55,3,5,'X55','Experience in the domain','Way to acquire the experience by developer. Locate it in any of the alternatives: in educational institution, in industry or independently.',27),(56,3,5,'X56','Team experience in using support tools','Locate the experience in the use of the support tools: none, low, medium, high or very high.',27),(57,5,1,'X57','Project application domain','Name of the technological domain where it applies: web, medical, administrative or engineering.',3),(58,5,2,'X58','Paradigm of the main language','Paradigm of the main language: procedural, object-oriented, functional, logical, event-driven, imperative, arrangement oriented, stack oriented, aspect oriented, data flow, visual or assembly.',9),(59,5,2,'X59','Programming languages','Names of programming languages used:Java, C ++, C, C #, Cliper, Cobol, JavaScript, Visual Basic, Pascal, Delphi or other.',9),(60,5,6,'X60','Planning tools','Names of planning tools: Rational Rose, Microsoft Office Visio, Together,

```

Microsoft Office Project or
other.',30),(61,5,6,'X61','Modeling
tools','Names of the modeling tools: Rational
RosE, Enterprise Architect, Altova, Eclipse
Modelig framework or
other.',30),(62,5,6,'X62','Programming
tools','Names of the programming tools: IDE-
General, IDE-Programming, NetBeans, Eclipse or
other.',30),(63,5,3,'X63','Database
manager','Manipulator names: Acces, Informix,
MySQL, Oracle, PostgresQL, SQL Server, Sybase
or other.',18),(64,5,2,'X64','Operating
systems ','Names of the operating systems in
which the application will run: Apple-
Macintosh, Linux, Microsoft Windows, Unix or
other.',10),(65,5,2,'X65','Types of machine
','Names of the types of machines in which it
will be used: mainframe, mini computer, pc,
mobile device or
other.',10),(66,1,6,'X66a','Planning','Docume
ntation
planning.',30),(67,1,4,'X66b','Business
model','Documentation business
model.',22),(68,1,3,'X66c','Identification of
components','Documentation identification of
components.',14),(69,1,3,'X66d','System
prototype','Documentation system
prototype.',14),(70,1,2,'X66e','System
requirements','Documentation system
requirements.',7),(71,1,6,'X66f','Requirement
s review and system prototype','Documentation
requirements review and system
prototype.',35),(72,1,2,'X66g','Reverse
engineering','Documentation reverse
engineering.',1),(73,1,3,'X66h','Direct
engineering','Documentation direct
engineering.',18),(74,1,3,'X66i','Software
prototype','Documentation software
prototype.',14),(75,1,2,'X66j','Software
requirements','Documentation software
requirements.',7),(76,1,4,'X66k','Specificati
on of acceptance tests','Documentation
specification of acceptance
tests.',19),(77,1,6,'X66l','Review
of
software requirements and specification of
acceptance tests','Documentation review of
software requirements and specification of
acceptance
tests.',17),(78,1,2,'X66m','Platform
definition and requirements
restrictions','Documentation platform
definition and requirements
restrictions.',9),(79,1,3,'X66n','HLD
software architecture','Documentation HLD
software
architecture.',13),(80,1,3,'X66ñ','Specificat
ion of integration tests','Documentation
specification of integration
tests.',14),(81,1,6,'X66o','Software
architecture review','Documentation software
architecture
review.',35),(82,1,7,'X66p','Division of work
structure','Documentation division of work
structure.',39),(83,1,2,'X66q','Detailed
design of work units','Documentation detailed
design
of
work

```

```

units.',10),(84,1,3,'X66r','Unit
test
specification','Documentation unit
test
specification.',14),(85,1,6,'X66s','Detailed
design review and specification of unit
tests','Documentation detailed design review
and
specification
of
unit
tests.',35),(86,1,3,'X66t','Unit
code','Documentation
unit
code.',18),(87,1,6,'X66u','Drive
code
review','Documentation drive
code
review.',35),(88,1,3,'X66v','Unit
test
documentation','Documentation
unit
test.',16),(89,1,3,'X66w','Integration
test
documentation','Documentation
integration
test.',16),(90,1,4,'X66x','Test
and
acceptance documentation','Documentation test
and
acceptance.',23),(91,1,7,'X66y','Post-
mortem report of the development
process','Documentation post-mortem report of
the
development
process.',37),(92,1,2,'X67','Type of software
product assembly','Name the types of assembly-
part
included in the structure pattern:
system, subsystem, product, component, module
or
object.',11),(93,1,6,'X68','Set
of
subprocess','Name of the subprocesses included
in
the process pattern: launch & strategy,
planning, specification of requirements,
analysis & design, implementation, testing or
post-mortem
evaluation.',33),(94,1,5,'X69','Structure of
the work team','Name of roles included in the
role
pattern: team leader, customer interface
manager, design administrator, deployment
administrator, test administrator, planning
administrator, process administrator, quality
manager,
support
manager.',29),(95,1,5,'X70','Way to perform
the development','Choice of collaborative or
individual
work: individual or
collaborative.',26),(96,1,6,'X71','Periodicit
y','Daily, weekly, biweekly, monthly or
other.',32),(97,1,6,'X72','Schedule','Pre-
established
or
variable.',32),(98,1,6,'X73','Way to track
and
control
the
process','Duration.',32),(99,1,6,'X74','Type
of inspection','Formal technique or informal
technique.',31),(100,1,6,'X75','Duration','Du
ration.',31),(101,1,6,'X76','Way to conduct
inspections','Average number of identified
issues
or
defects.',31),(102,1,6,'X77','Project
development model','Main model of development
process: Cascade, Evolutionary, Incremental,
Spiral, Win win, Agile or Other.',33);
/*140000 ALTER TABLE `mcpsr` ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table
`mcpsr_competency`
DROP TABLE IF EXISTS `mcpsr_competency`;
/*!40101 SET @saved_cs_client
=
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `mcpsr_competency` (

```

```

`id_mcpsr_competencies` int(11) NOT NULL
AUTO_INCREMENT,
`id_mcpsr` int(11) NOT NULL,
`id_competency` int(11) NOT NULL,
PRIMARY KEY (`id_mcpsr_competencies`),
KEY `id_competency_idx` (`id_competency`),
KEY `id_mcpsr_idx` (`id_mcpsr`),
CONSTRAINT `FK_id_competency` FOREIGN KEY
(`id_competency`) REFERENCES `competencies`
(`id_competency`) ON DELETE NO ACTION ON
UPDATE NO ACTION,
CONSTRAINT `FK_id_mcpsr_competency` FOREIGN
KEY (`id_mcpsr`) REFERENCES `mcpsr`
(`id_mcpsr`) ON DELETE NO ACTION ON UPDATE NO
ACTION
) ENGINE=InnoDB AUTO_INCREMENT=139 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `mcpsr_competency`
LOCK TABLES `mcpsr_competency` WRITE;
/*!40000 ALTER TABLE `mcpsr_competency`
DISABLE KEYS */;
INSERT INTO `mcpsr_competency` VALUES
(1,1,3),(2,2,3),(3,3,3),(4,4,3),(5,4,1),(6,5,
3),(7,5,1),(8,6,3),(9,6,1),(10,7,3),(11,7,1),
(12,8,3),(13,8,1),(14,9,3),(15,9,1),(16,10,3)
,(17,10,1),(18,11,3),(19,11,1),(20,12,3),(21,
12,1),(22,13,3),(23,13,1),(24,14,3),(25,14,1)
,(26,15,3),(27,15,1),(28,16,2),(29,17,3),(30,
18,2),(31,19,3),(32,20,2),(33,21,3),(34,22,2)
,(35,23,3),(36,24,2),(37,25,3),(38,26,2),(39,
27,3),(40,28,2),(41,29,3),(42,30,2),(43,31,3)
,(44,32,2),(45,33,3),(46,34,2),(47,34,5),(48,
35,2),(49,35,5),(50,36,3),(51,37,3),(52,38,3)
,(53,38,1),(54,39,3),(55,39,1),(56,40,3),(57,
41,3),(58,41,4),(59,42,3),(60,43,4),(61,44,3)
,(62,44,4),(63,45,3),(64,45,4),(65,46,3),(66,
46,4),(67,47,3),(68,47,4),(69,48,3),(70,48,4)
,(71,49,3),(72,49,4),(73,50,3),(74,50,4),(75,
51,3),(76,51,4),(77,52,3),(78,52,4),(79,53,3)
,(80,53,4),(81,54,3),(82,54,4),(83,55,3),(84,
55,4),(85,56,3),(86,56,4),(87,57,3),(88,58,3)
,(89,59,3),(90,60,4),(91,61,4),(92,62,4),(93,
63,2),(94,64,1),(95,65,1),(96,66,3),(97,66,4)
,(98,67,1),(99,68,1),(100,69,2),(101,70,1),(1
02,71,2),(103,72,1),(104,73,2),(105,74,2),(10
6,75,1),(107,76,5),(108,77,2),(109,78,1),(110
,79,1),(111,80,5),(112,81,2),(113,82,3),(114,
82,4),(115,83,1),(116,84,5),(117,85,2),(118,8
6,2),(119,87,2),(120,88,2),(121,89,2),(122,90
,2),(123,91,2),(124,92,3),(125,93,3),(126,93,
4),(127,94,3),(128,94,4),(129,95,3),(130,95,4
),(131,96,4),(132,97,4),(133,98,4),(134,99,4)
,(135,100,4),(136,101,4),(137,102,3),(138,102
,4);
/*!40000 ALTER TABLE `mcpsr_competency` ENABLE
KEYS */;
UNLOCK TABLES;
-- Table structure for table `milestones`
DROP TABLE IF EXISTS `milestones`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `milestones` (

```

```

`id_mileston` int(11) NOT NULL
AUTO_INCREMENT,
`stage` varchar(45) CHARACTER SET utf8 NOT
NULL,
`substage` varchar(45) CHARACTER SET utf8
DEFAULT NULL,
PRIMARY KEY (`id_mileston`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table `milestones`
LOCK TABLES `milestones` WRITE;
/*!40000 ALTER TABLE `milestones` DISABLE KEYS
*/;
INSERT INTO `milestones` VALUES (1,'Before
approval',''),(2,'In development','Decision
to fund'),(3,'In development','Skinny system
available'),(4,'In development','Usable
system available'),(5,'In
production','Decision to go live');
/*!40000 ALTER TABLE `milestones` ENABLE KEYS
*/;
UNLOCK TABLES;
-- Table structure for table
`operational_metadata`
DROP TABLE IF EXISTS `operational_metadata`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `operational_metadata` (
`id_operational_metadata` int(11) NOT NULL
AUTO_INCREMENT,
`date` datetime NOT NULL,
`operational` varchar(250) CHARACTER SET
utf8 NOT NULL,
`description` varchar(250) CHARACTER SET
utf8 DEFAULT NULL,
PRIMARY KEY (`id_operational_metadata`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!40101 SET character_set_client =
@saved_cs_client */;
-- Dumping data for table
`operational_metadata`
LOCK TABLES `operational_metadata` WRITE;
/*!40000 ALTER TABLE `operational_metadata`
DISABLE KEYS */;
/*!40000 ALTER TABLE `operational_metadata`
ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table `query_metadata`
DROP TABLE IF EXISTS `query_metadata`;
/*!40101 SET @saved_cs_client =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `query_metadata` (
`id_query_metadata` int(11) NOT NULL
AUTO_INCREMENT,
`date` datetime NOT NULL,
`conditional` varchar(45) CHARACTER SET utf8
NOT NULL,
`id_column_mcpsr` int(11) NOT NULL,
PRIMARY KEY (`id_query_metadata`),
KEY `id_column_mcpsr` (`id_column_mcpsr`),

```



```

CONSTRAINT `id_column_mcpsr` FOREIGN KEY
(`id_column_mcpsr`) REFERENCES
`columns_mcpsr` (`id_column_mcpsr`) ON DELETE
NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!140101 SET character_set_client =
@savед_cs_client */;
-- Dumping data for table `querys_metadata`
LOCK TABLES `querys_metadata` WRITE;
/*!140000 ALTER TABLE `querys_metadata` DISABLE
KEYS */;
/*!140000 ALTER TABLE `querys_metadata` ENABLE
KEYS */;
UNLOCK TABLES;
-- Table structure for table `repositories`
DROP TABLE IF EXISTS `repositories`;
/*!140101 SET @savед_cs_client =
@@character_set_client */;
/*!140101 SET character_set_client = utf8 */;
CREATE TABLE `repositories` (
  `id_repository` int(11) NOT NULL
  AUTO_INCREMENT,
  `name_repository` varchar(45) CHARACTER SET
  utf8 NOT NULL,
  PRIMARY KEY (`id_repository`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!140101 SET character_set_client =
@savед_cs_client */;
-- Dumping data for table `repositories`
LOCK TABLES `repositories` WRITE;
/*!140000 ALTER TABLE `repositories` DISABLE
KEYS */;
/*!140000 ALTER TABLE `repositories` ENABLE
KEYS */;
UNLOCK TABLES;
-- Table structure for table `state_alpha`
DROP TABLE IF EXISTS `state_alpha`;
/*!140101 SET @savед_cs_client =
@@character_set_client */;
/*!140101 SET character_set_client = utf8 */;
CREATE TABLE `state_alpha` (
  `idstate_alpha` int(11) NOT NULL,
  `state_alpha` varchar(45) CHARACTER SET utf8
  DEFAULT NULL,
  `description` varchar(1000) CHARACTER SET
  utf8 DEFAULT NULL,
  `milestones_id_mileston` int(11) NOT NULL,
  PRIMARY KEY (`idstate_alpha`),
  KEY `fk_state_alpha_milestones1_idx`
  (`milestones_id_mileston`),
  CONSTRAINT `fk_state_alpha_milestones1`
  FOREIGN KEY (`milestones_id_mileston`)
  REFERENCES `milestones` (`id_mileston`) ON
  DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!140101 SET character_set_client =
@savед_cs_client */;
-- Dumping data for table `state_alpha`
LOCK TABLES `state_alpha` WRITE;
/*!140000 ALTER TABLE `state_alpha` DISABLE
KEYS */;
INSERT INTO `state_alpha` VALUES
(1,'Addressed','A solution has been produced

```

that demonstrably addresses the opportunity.',2),(2,'Benefit accrued','The operational use or sale of the solution is creating tangible benefits.',5),(3,'Identified','A commercial, social or business opportunity has been identified that could be addressed by a software-based solution.',1),(4,'Solution needed','The need for a software-based solution has been confirmed.',1),(5,'Value established','The value of a successful solution has been established.',1),(6,'Viable','It is agreed that a solution can be produced quickly and cheaply enough to successfully address the opportunity',1),(7,'Acceptable','The requirements describe a system that is acceptable to the stakeholders.',3),(8,'Addressed','Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders.',3),(9,'Bounded','The purpose and theme of the new system are clear.',1),(10,'Coherent','The requirements provide a consistent description of the essential characteristics of the new system.',2),(11,'Conceived','The need for a new system has been agreed.',1),(12,'Fulfilled','The requirements that have been addressed fully satisfy the need for a new system.',4),(13,'Architecture selected','An architecture has been selected that addresses the key technical risks and any applicable organizational constraints.',1),(14,'Demonstrable','An executable version of the system is available that demonstrates the architecture is fit for purpose and supports testing.',2),(15,'Operational','The system is use in a live environment.',5),(16,'Ready','The system (as a whole) has been accepted for deployment in a live environment.',4),(17,'Retired','The system is no longer supported.',5),(18,'Usable','The system is usable and demonstrates all of the quality characteristics required of an operational system.',3),(19,'In_agreement','The stakeholder representatives are in agreement.',2),(20,'Involved','The stakeholder representatives are actively involved in the work and fulfilling their responsibilities.',2),(21,'Recognized','The stakeholders have been identified.',1),(22,'Represented','The mechanisms for involving the stakeholders are agreed and the stakeholder representatives have been appointed.',1),(23,'Satisfied for deployment','The minimal expectations of the stakeholder representatives have been achieved.',4),(24,'Satisfied_in_use','The system meets or exceeds the minimal stakeholder expectations.',5),(25,'Adjoumed','The team is no longer accountable for carrying out its

```

mission.',5),(26,'Collaborating','The team
members are working together as one
unit.',2),(27,'Formed','The team has been
populated with enough committed people to
start the mission.',2),(28,'Performing','The
team is working effectively and
efficiently.',2),(29,'Seeded','The team's
mission is clear and the know-how needed to
grow the team is in place.',1),(30,'Foundation
established','The key practices, and tools,
that form the foundation of the way of working
are selected and ready for use.',1),(31,'In
place','All team members are using the way-of-
working to accomplish their tasks.',2),(32,'In
use','Some members of the team are using and
adapting, the way-of-
working.',2),(33,'Principles
established','The principles, and
constraints, that shape the way-of-working are
established.',1),(34,'Retired','The way-of-
working is no longer in use by the
team.',4),(35,'Working well','The way of-
working is working well for the
team.',3),(36,'Closed','All remaining
housekeeping tasks have been completed and the
work has been officially
closed.',5),(37,'Concluded','The work to
produce the results has been
concluded.',4),(38,'Initiated','Work has been
requested',1),(39,'Prepared','All pre-
conditions for starting the work have been
met.',1),(40,'Started','The work is
proceeding.',2),(41,'Under control','The work
is going well, risks are under control and
productivity levels are sufficient to achieve
a satisfactory result.',2);
/*!40000 ALTER TABLE `state_alpha` ENABLE KEYS
*/;
UNLOCK TABLES;
-- Table structure for table `tables`
DROP TABLE IF EXISTS `tables`;

```

```

/*!40101 SET @saved_cs_client      =
@@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `tables` (
  `id_table` int(11) NOT NULL AUTO_INCREMENT,
  `id_repository` int(11) NOT NULL,
  `name_table` varchar(45) CHARACTER SET utf8
NOT NULL,
  PRIMARY KEY (`id_table`),
  KEY `id_repository_idx` (`id_repository`),
  CONSTRAINT `FK_id_repository` FOREIGN KEY
(`id_repository`) REFERENCES `repositories`
(`id_repository`) ON DELETE NO ACTION ON
UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
COLLATE=utf8_bin;
/*!40101 SET character_set_client  =
@saved_cs_client */;
-- Dumping data for table `tables`
LOCK TABLES `tables` WRITE;
/*!40000 ALTER TABLE `tables` DISABLE KEYS */;
/*!40000 ALTER TABLE `tables` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
*/;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS
*/;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIEN
T */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESU
LTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTIO
N */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
-- Dump completed on 2019-04-29 23:47:48

```

En la Figura 4.19 se muestra el resultado de la ejecución del script SQL obtenido del diseño lógico mostrado en la sección 4.3, se observa que la ejecución transcurrió sin errores.

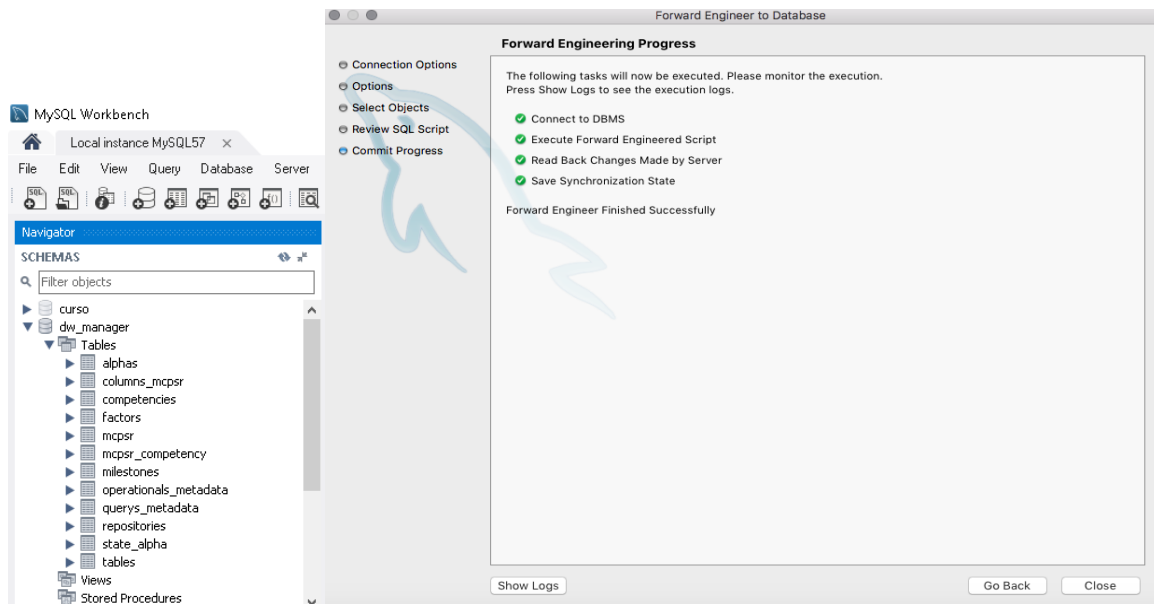


Figura 4.19 Ejecución sin errores del script y estructura final de base de datos dw_manager
Fuente: creación propia.

En este capítulo se seleccionaron los repositorios de software que cumplieron con las reglas establecidas para ser usados en esta investigación; así como el diseño del almacén de datos y su creación. En el siguiente capítulo se especifican los requerimientos y se diseña la herramienta Almacén de Datos Esenciales del Desarrollo de Software (ADE-DS).

Capítulo 5

Requerimientos y Diseño del sistema

En el presente capítulo, se describe el proceso para la Especificación de requerimientos (Sección 5.1) y el proceso para el diseño del sistema ADE-DS (Sección 5.2). Para la especificación de requerimientos se utilizan dos lenguajes de URN (*User Requirements Notation*), basados en la recomendación (ITU-T Z.151, 2012). Para el diseño del sistema se utiliza diagramas de caso de uso, secuencia y clases.

A continuación, se presenta la estructura del capítulo:

- 5.1 Especificación de requerimientos
 - 5.1.1 Notación de requerimientos de usuario
 - 5.1.2 Perspectiva del producto
 - 5.1.3 Funciones del producto
 - 5.1.4 Características del usuario
 - 5.1.5 Especificación URN
 - 5.1.5.1 Diagrama GRL
 - 5.1.5.2 Diagramas UCM
 - 5.1.6 Requerimientos obtenidos
 - 5.1.6.1 Requerimientos funcionales
 - 5.1.6.2 Requerimientos no funcionales
- 5.2 Diseño del sistema
 - 5.2.1 Diagramas caso de uso
 - 5.2.2 Diagramas de secuencia
 - 5.2.3 Diagrama de clases

5.1 Especificación de requerimientos

La realización de este documento de especificación se basa en los lineamientos del estándar IEEE-STD-830-1998, en el que se presenta el contenido de la Especificación de Requerimientos de Software (SRS).

5.1.1 Notación de requerimientos de usuario

La recomendación (ITU-T Z.151, 2012) define al lenguaje GRL (*Goal-oriented Requirements Language*) como un lenguaje para soportar modelado orientado a metas y razonamiento sobre requerimientos, especialmente requerimientos no funcionales y atributos de calidad. GRL proporciona construcciones visuales como se muestra en la Figura 5.1 para expresar diversos tipos de conceptos que aparecen durante el proceso de especificación de requerimientos.

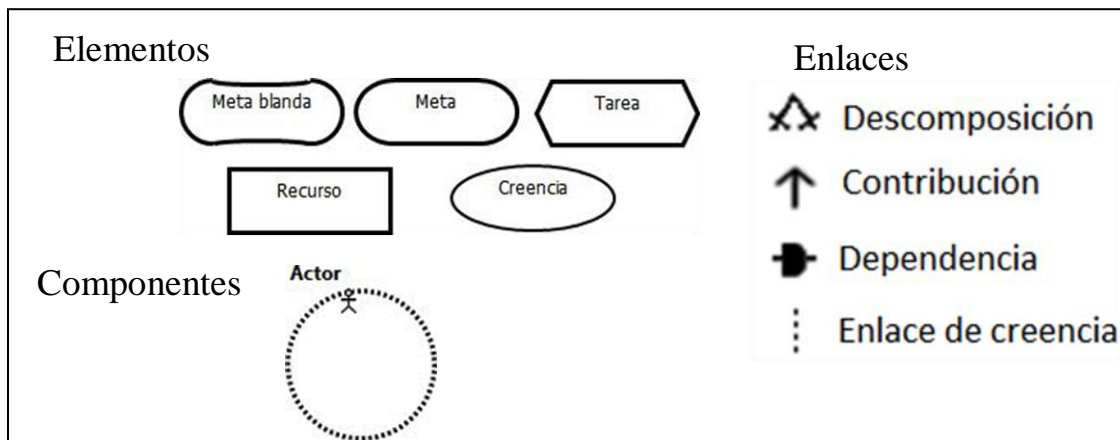


Figura 5.1 Construcciones visuales de GRL herramienta JUCMNav v7.0.0.
Fuente: Obtenida y traducida de (ITU-T Z.151,2012)

En la recomendación (ITU-T Z.151,2012) se menciona que UCM (*Use Case Maps*) es un lenguaje de especificación que debido a su naturaleza visual simple e intuitiva permite utilizarse tanto por modeladores, así como los no especialistas, pero al mismo tiempo ofrece suficiente rigurosidad para los desarrolladores. Que usa los símbolos que se muestran en la Figura 5.2. Así, las especificaciones UCM emplean rutas de escenarios para ilustrar las relaciones casuales entre las responsabilidades, proporcionando una visión integrada de comportamiento y estructura. Los mapas resultantes, pueden representar la superposición de trayectorias de escenarios, en una estructura de componentes abstractos.

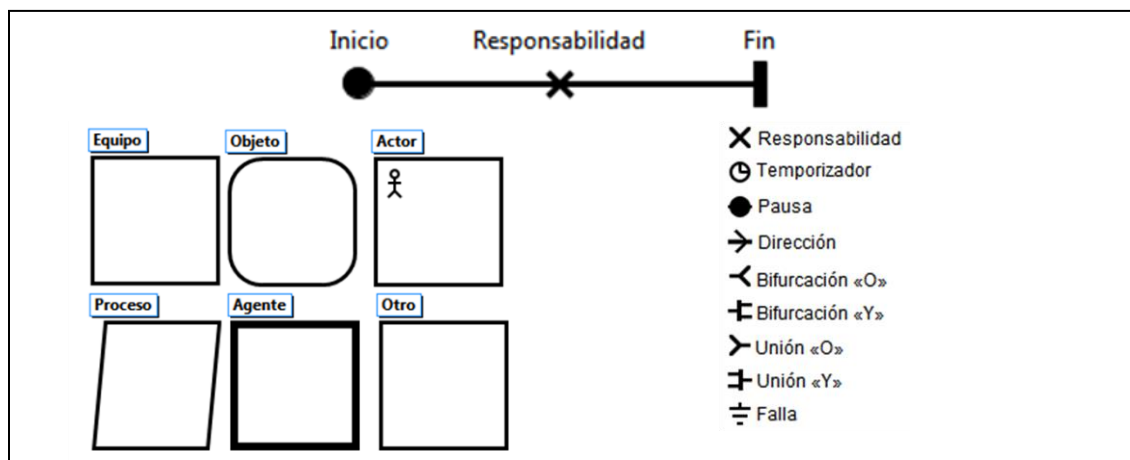


Figura 5.2 Construcciones visuales de UCM herramienta JUCMNav v7.0.0.
Fuente: Obtenida y traducida de (ITU-T Z.151,2012)

Según (Amyot, 2003), los componentes UCM podrían también ser el enlace entre los escenarios de URN (*User Requirements Notations*) y otros lenguajes. Las relaciones de alto nivel que existen entre GRL, UCM, otros lenguajes ITU-T, y UML se resumen en la Figura 5.3 . URN representa la pieza faltante del rompecabezas de modelado que conecta los requisitos informales y los casos de uso de modelos estructurales y de comportamiento.

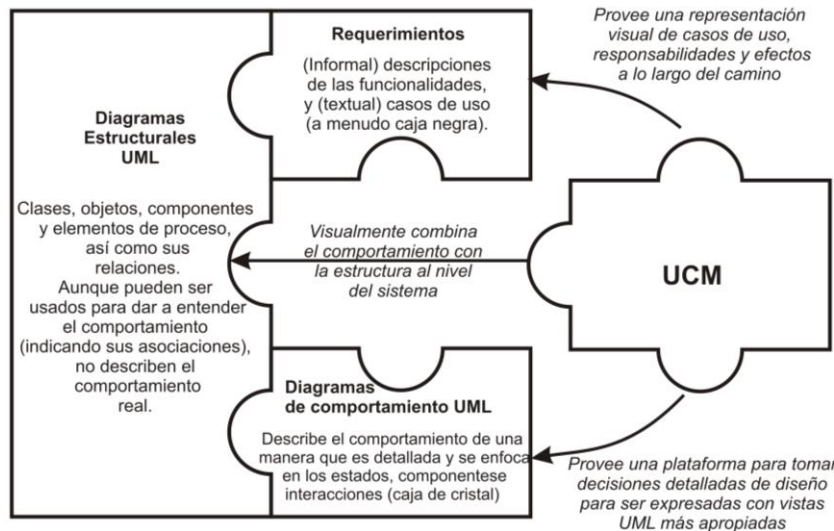


Figura 5.3 GRL y UCM como una pieza faltante del rompecabezas UML.

Fuente: Obtenida de (Amyot, 2003)

5.1.2 Perspectiva del producto

En esta sección se describen las funciones del producto y los requerimientos funcionales y no funcionales, por medio de diagramas GRL y UCM de acuerdo a la recomendación (ITU-T Z.151,2012) para la especificación de requerimientos URN.

5.1.3 Funciones del producto

El sistema realizará lo siguiente:

- Funciones o servicios de un administrador de almacén de datos.
- Carga para integrar los datos de los repositorios FLOSSmole hacia el almacén.

5.1.4 Características del usuario

La funcionalidad de los servicios de un administrador de almacén de datos, no se verá afectada así que el administrador del almacen no necesita conocimientos extra, solo los necesarios para utilizar el sistema administrador de almacén de datos (como los repositorios FLOSSmole a utilizar y la ontología MCPS-R).

5.1.5 Especificación URN

En esta sección se presentan los diagramas GRL y UCM, que describen los requerimientos para el sistema administrador de almacén de datos. Los cuales son el diagrama GRL del sistema administrador de almacén de datos y los diagramas UCM de cada uno de los servicios del administrador.

5.1.5.1 Diagrama GRL

En la Figura 5.4, se visualiza el diagrama GRL del sistema administrador de almacén de datos, este diagrama contiene 4 metas principales que deberá cumplir nuestro sistema: carga, gestión de metadatos, respaldo del almacén de datos e interfaz de consultas. Para lograr cada una de estas metas se requieren diferentes tareas que describen el soporte funcional requerido.

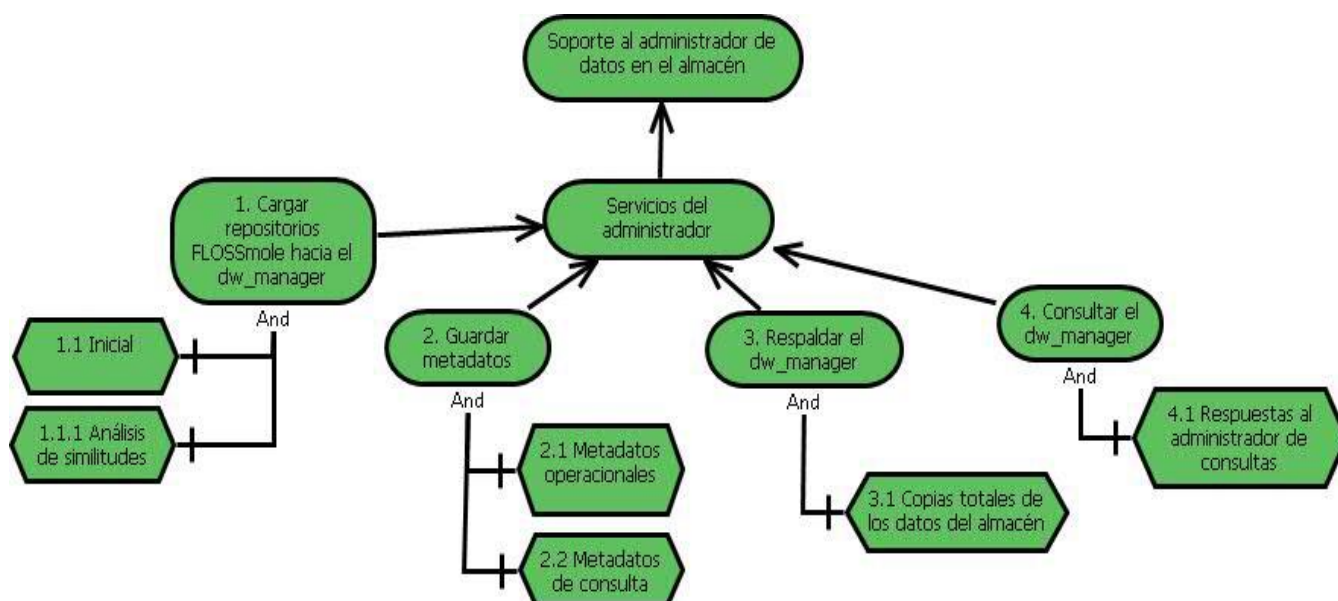


Figura 5.4 Diagrama GRL-Sistema administrador de almacén de datos.

Fuente: creación propia

5.1.5.2 Diagramas UCM

En la Figura 5.5, se observa el proceso para realizar la carga de un repositorio hacia el almacén. En la cual el usuario selecciona en el sistema la opción de carga para la cual elige un repositorio FLOSSmole, el sistema extraerá los datos y propiedades del archivo CSV y creará la estructura del repositorio en el gestor de base de datos. Al finalizar se mostrará un análisis de similitudes del repositorio cargado. Si la carga es exitosa, esta será utilizada por los demás servicios del administrador.

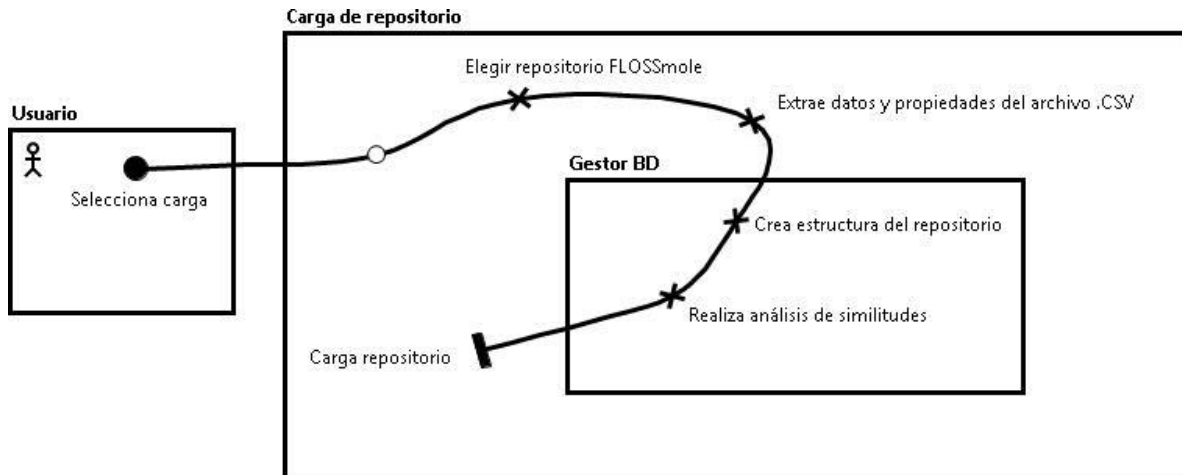


Figura 5.5 Diagrama UCM-Cargar repositorios FLOSSmole hacia el dw_manager.
Fuente: creación propia

En la Figura 5.6, se observa el proceso para realizar la gestión de metadatos de las operaciones dentro del almacén y consultas. En el cual el usuario puede seleccionar los metadatos a consultar (operacionales o consulta), el sistema guardará automáticamente los metadatos realizados con el sistema y consultados en el almacén de datos.

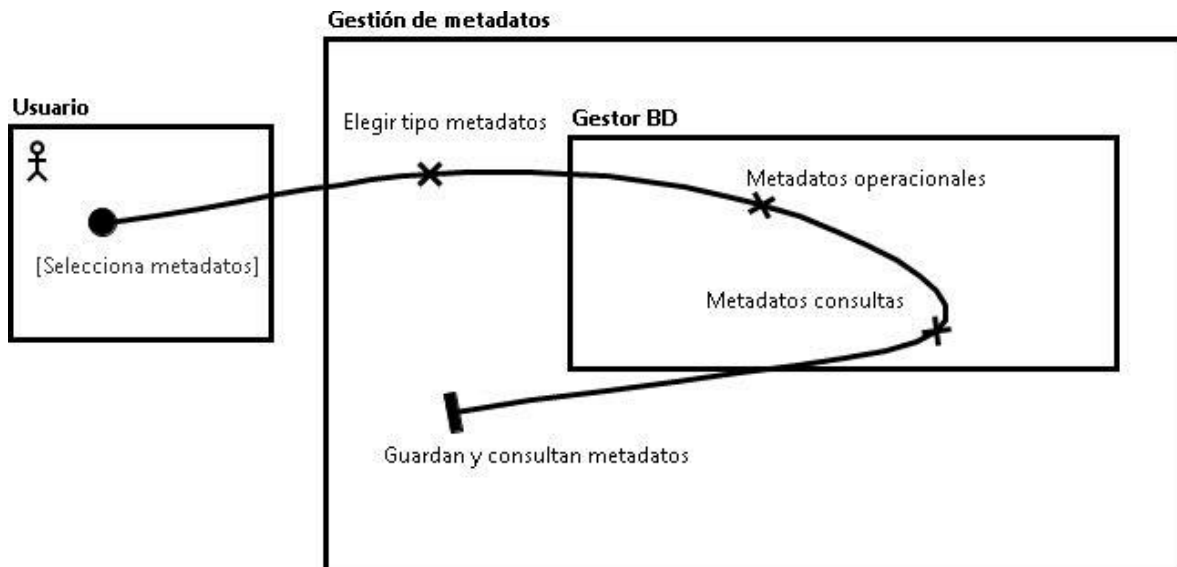


Figura 5.6 Diagrama UCM-Guardar metadatos
Fuente: creación propia

En la Figura 5.7, se observa el proceso para realizar el respaldo del almacén de datos. En la cual el usuario selecciona la ubicación de donde guardara el respaldo y se genera una copia del almacen de datos.

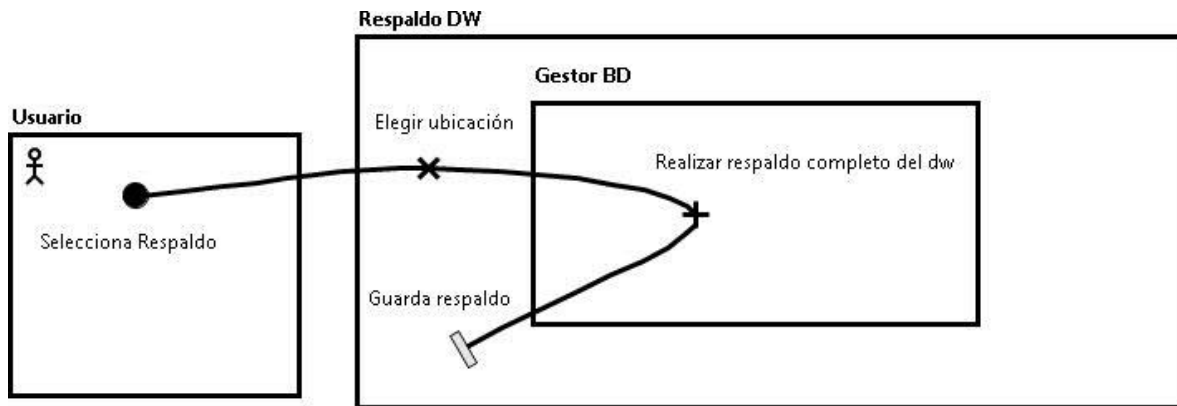


Figura 5.7 Diagrama UCM-Respaldo el dw_manager.
Fuente: creación propia

En la Figura 5.8, se observa el proceso para realizar las consultas dentro del almacén de datos para dar respuestas al usuario. En la cual el usuario selecciona la opción de consultas, mediante la selección de repositorios FLOSSmole cargados en el almacén podrá elegir y la tabla a consultar. El sistema realiza la búsqueda en el almacén de datos y envía una respuesta.

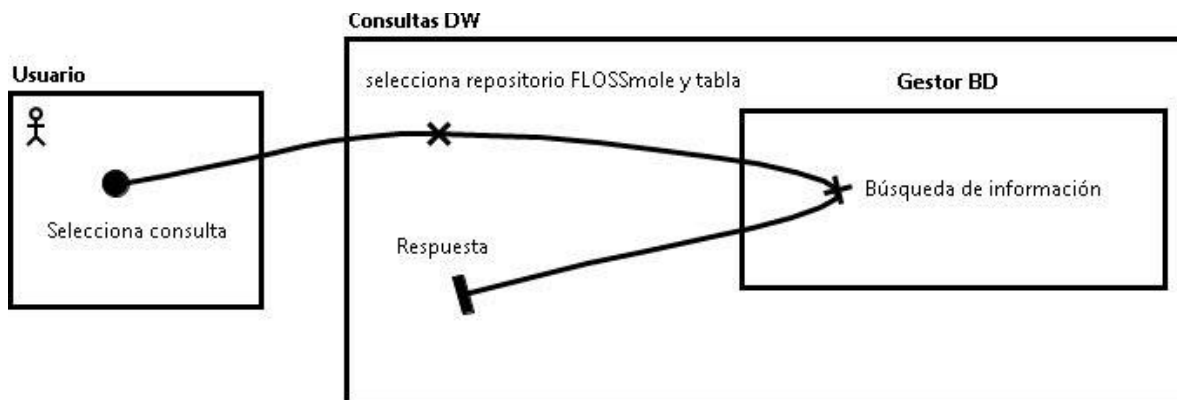


Figura 5.8 Diagrama UCM-Consultar el dw_manager.
Fuente: creación propia

5.1.6 Requerimientos obtenidos

A continuación, se presentan los requerimientos funcionales y no funcionales obtenidos a partir del diagrama GRL, los requerimientos funcionales son esenciales para el soporte funcional que proporcionará el sistema administrador de almacén de datos.

5.1.6.1 Requerimientos funcionales

1. Cargar repositorios FLOSSmole hacia el dw_manager
 - 1.1 Inicial
 - 1.1.1 Análisis de similitudes
2. Guardar metadatos
 - 2.1 Metadatos operacionales
 - 2.2 Metadatos de consulta
3. Respalidar el dw_manager
 - 3.1 Copias totales de los datos del almacén
4. Consultar el dw_manager
 - 4.1 Respuesta al administrador de consultas

5.1.6.2 Requerimientos no funcionales

Usabilidad

El sistema contará con una interfaz gráfica simple, que permitirá reducir su tiempo de aprendizaje por el usuario; además, se proporcionarán mensajes de error cuando se presente algún inconveniente durante su ejecución.

Mantenibilidad

El mantenimiento del ambiente se efectuará de la forma más sencilla posible, el sistema podrá ser modificado con facilidad para corregir fallos o mejorar su funcionalidad.

5.2 Diseño del sistema

En base a los requerimientos obtenidos del diagrama GRL se procedió a realizar los diagramas de caso de uso de todas las funciones del sistema DW-Manager, diagramas de secuencias para describir el comportamiento dinámico del sistema y el diagrama de clases mostrando la estructura de las clases, métodos y objetos que conforman el sistema.

5.2.1 Diagramas caso de uso

En la Figura 5.9 se puede observar que el usuario administrador puede usar el sistema administrador del almacén ADE-DS.

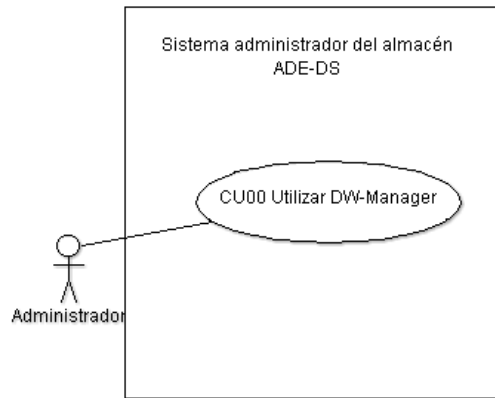


Figura 5.9 Diagrama de caso de uso CU00 Utilizar ADE-DS.
Fuente: creación propia

En la Figura 5.10 se muestra la interacción del usuario administrador con el sistema administrador del almacén ADE-DS.

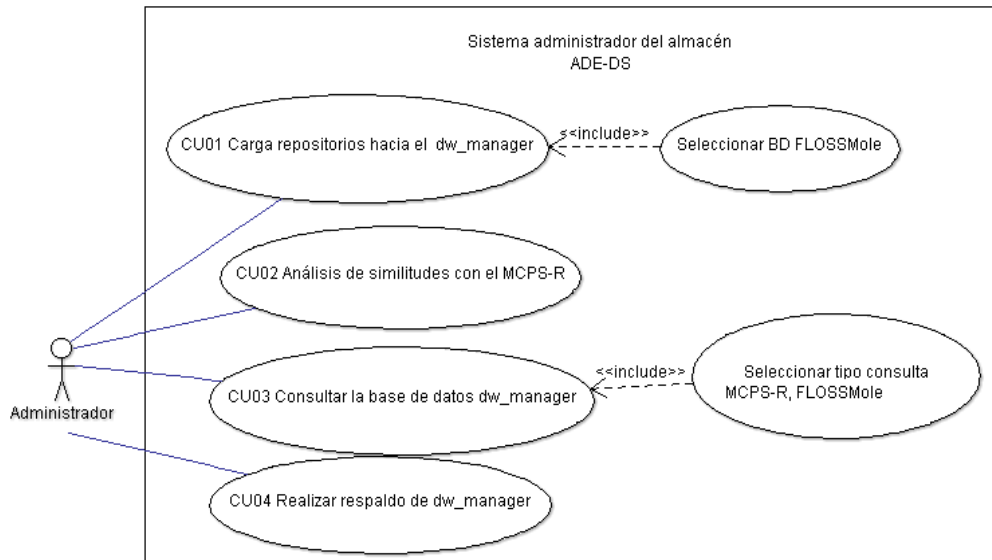


Figura 5.10 Diagrama de casos de uso del administrador.
Fuente: creación propia

En la Figura 5.11 se observan las actividades del sistema para gestionar los metadatos del ADE-DS.

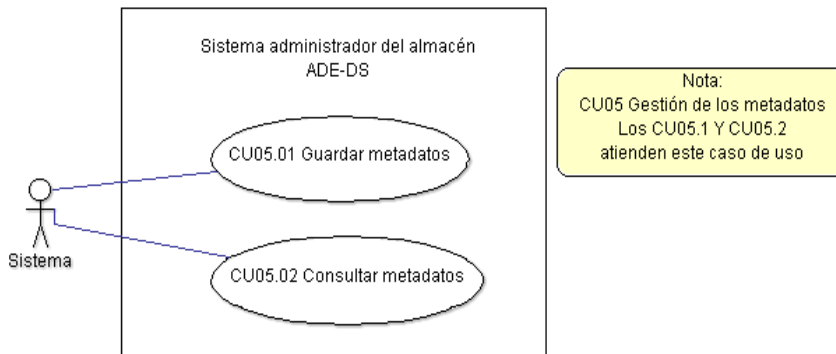


Figura 5.11 Diagrama de casos de uso del sistema ADE-DS.

Fuente: creación propia

A continuación, se describen los casos de uso textuales, que se asocian a los diagramas de caso de uso anteriores, presentados mediante las Tablas 5.1, 5.2, 5.3, 5.4, 5.5 y 5.6.

Tabla 5.1. CU01 Cargar repositorios FLOSSmole hacia el dw_manager.

Fuente: creación propia.

CU01 Carga repositorios FLOSSmole hacia el dw_manager

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU01		
Nombre	Carga repositorios FLOSSmole hacia el dw_manager		
Objetivo	Cargar los archivos CSV de repositorios FLOSSMole hacia el almacén		
Descripción	El sistema de ADE-DS al realizar este caso de uso permitirá realizar la carga de archivos CSV hacia el almacén dw_manager.		
Actores	Administrador		
Condiciones necesarias	El ADE-DS debe ser utilizado por el administrador.		
Escenario principal	<ol style="list-style-type: none"> 1. El administrador ejecuta el ADE-DS para poder utilizarlo. 2. El administrador selecciona la base de datos del repositorio a cargar. 3. El administrador da clic en examinar, para elegir los archivos CSV. 4. El administrador da clic en el botón carga CSV. 5. El sistema muestra análisis de similitudes. 		
Escenario alternativo	3a. Si en el punto 3 del escenario principal, el administrador elige los archivos CSV no correspondientes de la base del repositorio y da clic en carga CSV, mostrará un mensaje CSV incorrectos. A continuación, este caso de uso continúa con el paso 3.		
Escenarios de fracaso	<ol style="list-style-type: none"> 1. Error interno (conexión a BD dw_manager). 2. Paros de sistema inesperado. 		
Condiciones de éxito	Se realiza la carga de los archivos CSV en la base de datos dw_manager y se muestra análisis de similitudes.		

Cuestiones a resolver	N/A
------------------------------	-----

Tabla 5.2. CU02 Análisis de similitudes con el MPCPS-R.

Fuente: creación propia.

CU02 Análisis de similitudes con el MCPS-R

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU02		
Nombre	Análisis de similitudes con el MCPS-R		
Objetivo	Mostrar las similitudes que tienen los archivos CSV cargados en el almacén con el MCPS-R.		
Descripción	El sistema de ADE-DS al realizar este caso de uso permitirá mostrar las similitudes que tienen los archivos CSV con el MCPS-R.		
Actores	Administrador		
Condiciones necesarias	El administrador debe estar en la ventana de carga y haber realizado la carga de archivos CSV en el almacén.		
Escenario principal	<ol style="list-style-type: none"> 1. El administrador da clic en el botón cargar. 2. El administrador visualiza las similitudes que tienen los archivos CSV con el MCPS-R. 		
Escenario alternativo	2a. El administrador no visualiza las similitudes a continuación, este caso de uso continúa con el paso 1.		
Escenarios de fracaso	<ol style="list-style-type: none"> 1. Paros de sistema inesperado. 2. Error interno (conexión a BD dw_manager). 		
Condiciones de éxito	Se muestran las similitudes con el MCPS-R.		
Cuestiones a resolver	N/A		

Tabla 5.3. CU03 Consultar el dw_manager.

Fuente: creación propia.

CU03 Consultar la base de datos dw_manager

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU03		
Nombre	Consultar la base de datos dw_manager		
Objetivo	Poder realizar consultas a la base de datos dw_manager.		
Descripción	El sistema de ADE-DS al realizar este caso de uso permitirá consultar la información de la base de datos dw_manager.		
Actores	Administrador		
Condiciones necesarias	El administrador debe estar en la ventana de consultas.		

Escenario principal	<ol style="list-style-type: none"> 1. El administrador sincroniza a la base de datos dw_manager. 2. El administrador selecciona de un combobox los repositorios encontrados en la base de datos dw_manager. 3. El sistema muestra las tablas contenidas de ese repositorio. 4. El administrador elige la tabla a consultar. 5. El administrador visualiza los resultados de su consulta.
Escenario alternativo	2a. Si el administrador no selecciona del combobox el repositorio y no aparece debe realizar el paso 1.
Escenarios de fracaso	<ol style="list-style-type: none"> 1. Paros de sistema inesperado. 2. Error interno (Conexión BD dw_manager).
Condiciones de éxito	Se mostrará el resultado de la consulta realizada a la base de datos dw_manager.
Cuestiones a resolver	N/A

Tabla 5.4. CU04 Respaldo el dw_manager.

Fuente: creación propia.

CU04 Respaldo el dw_manager

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU04		
Nombre	Realizar respaldo de dw_manager		
Objetivo	Realizar respaldo total de la base de datos dw_manager.		
Descripción	El sistema de ADE-DS al realizar este caso de uso permitirá generar respaldo de la base de datos dw_manager.		
Actores	Administrador		
Condiciones necesarias	El administrador debe estar en la ventana de respaldo.		
Escenario principal	<ol style="list-style-type: none"> 1. El administrador da clic en el botón examinar para seleccionar la ubicación de donde guardará el respaldo de la base de datos dw_manager 2. El administrador da clic en el botón guardar respaldo. 3. El sistema muestra mensaje respaldo exitoso. 		
Escenario alternativo	2a. Si el administrador da clic en guardar respaldo y no selecciono la ubicación, mostrará mensaje error. Este caso de uso continúa con el paso 1.		
Escenarios de fracaso	<ol style="list-style-type: none"> 1. Paros de sistema inesperado. 2. Error interno (Conexión BD dw_manager). 		
Condiciones de éxito	Se realizará el respaldo de la base de datos dw_manager.		
Cuestiones a resolver	N/A		

Tabla 5.5. CU05 Guardar metadatos.

Fuente: creación propia.

CU05.01 Guardar metadatos

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU05.01		
Nombre	Guardar metadatos		
Objetivo	Guardar los metadatos operacionales y de consulta del ADE-DS.		
Descripción	El sistema al realizar este caso de uso permitirá guardar los metadatos operacionales y de consulta del ADE-DS.		
Actores	Sistema		
Condiciones necesarias	El administrador debe estar utilizando el ADE-DS.		
Escenario principal	<ol style="list-style-type: none"> 1. El sistema guardara en la tabla metadatos operacionales las funciones realizadas con el ADE-DS. 2. El sistema guardara en la tabla metadatos de consulta las consultas realizadas en la base de datos dw_manager. 		
Escenario alternativo			
Escenarios de fracaso	<ol style="list-style-type: none"> 1. Paros de sistema inesperado. 		
Condiciones de éxito	Se guardan los metadatos del ADE-DS.		
Cuestiones a resolver	N/A		

Tabla 5.6. CU05 Guardar metadatos.

Fuente: creación propia.

CU05.02 Consultar metadatos

Información de catalogación			
Proyecto	ADE-DS		
Autor	Rebeca Medina Jerónimo		
Versión	1.0	Estado de desarrollo	Borrador/No revisado
Definición del caso de uso			
Código	Caso de uso CU05.02		
Nombre	Consultar metadatos		
Objetivo	Consultar los metadatos guardados del ADE-DS		
Descripción	El sistema al realizar este caso de uso permitirá consultar los metadatos del ADE-DS.		
Actores	Sistema		
Condiciones necesarias	El administrador debe estar en la ventana metadatos.		
Escenario principal	<ol style="list-style-type: none"> 1. El administrador elige los metadatos que quiere consultar (operacionales o de consulta). 2. El administrador da clic en consultar. 3. El sistema muestra en una tabla los resultados. 		

Escenario alternativo	
Escenarios de fracaso	1. Paros de sistema inesperado.
Condiciones de éxito	Se consultan los metadatos del ADE-DS.
Cuestiones a resolver	N/A

5.2.2 Diagramas de secuencia

En la Figura 5.12 se muestra el diagrama de secuencia para cargar repositorios FLOSSmole hacia el dw_manager.

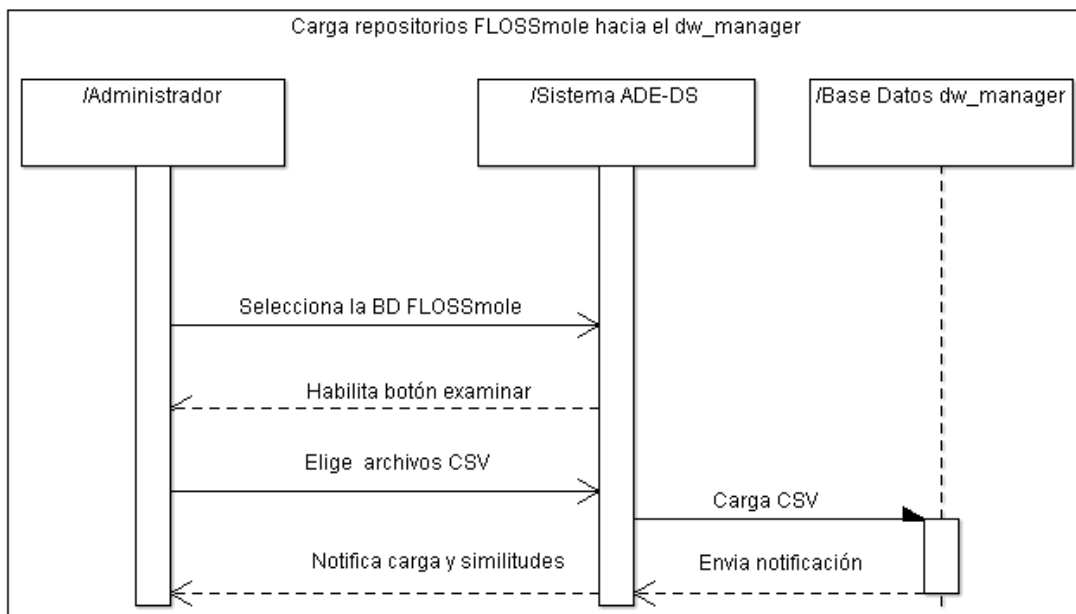


Figura 5.12 Diagrama de secuencia Carga repositorios FLOSSmole hacia el dw_manager.

Fuente: creación propia

En la Figura 5.13 se muestra el diagrama de secuencia para mostrar el análisis de similitudes con el MCPS-R.

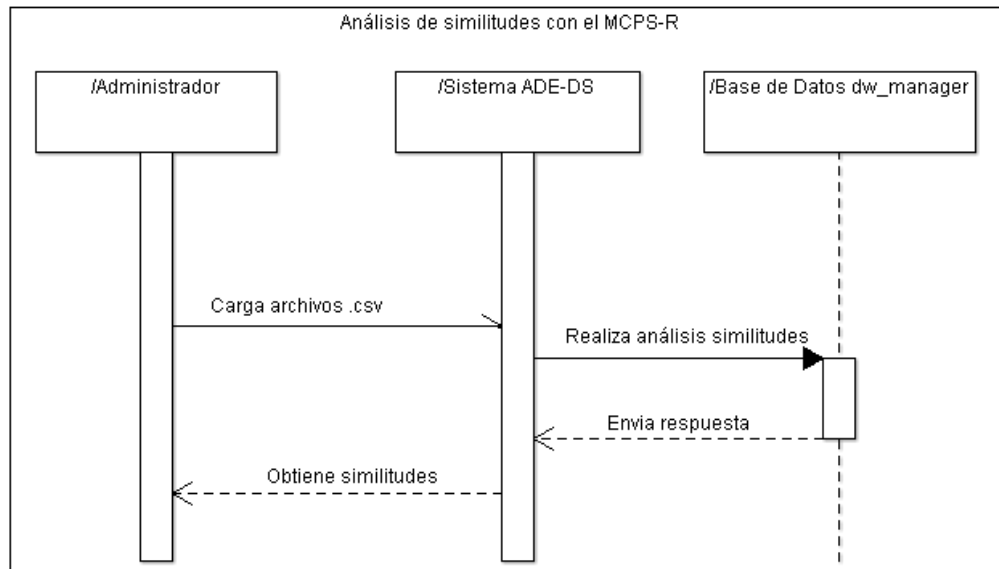


Figura 5.13 Diagrama de secuencia Análisis de similitudes con el MCPS-R.
Fuente: creación propia

En la Figura 5.14 se muestra el diagrama de secuencia para consultar la base de datos dw_manager.

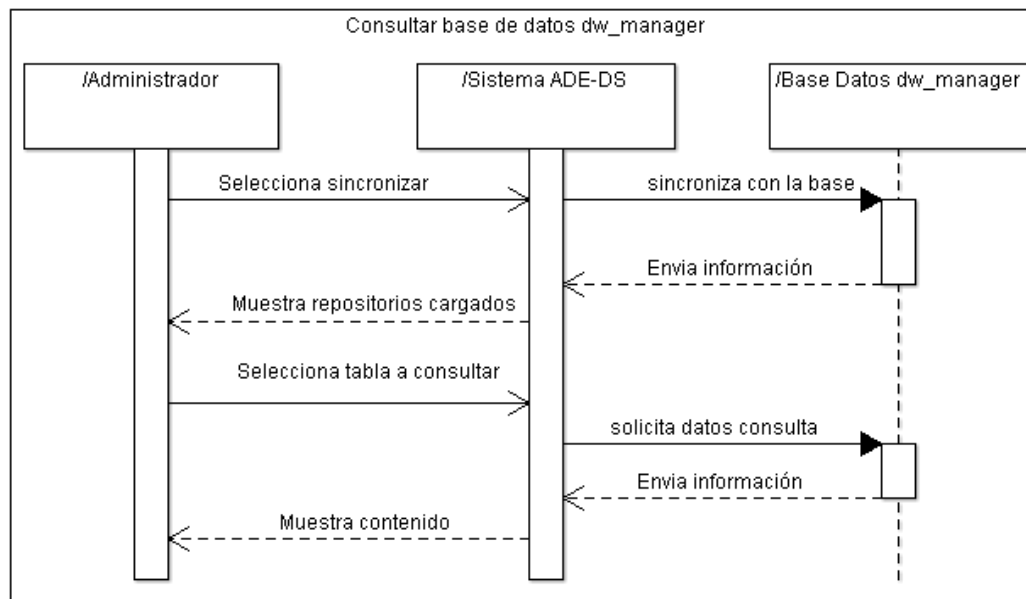


Figura 5.14 Diagrama de secuencia Consultar el dw_manager.
Fuente: creación propia

En la Figura 5.15 se muestra el diagrama de secuencia para respaldar la base de datos dw_manager.

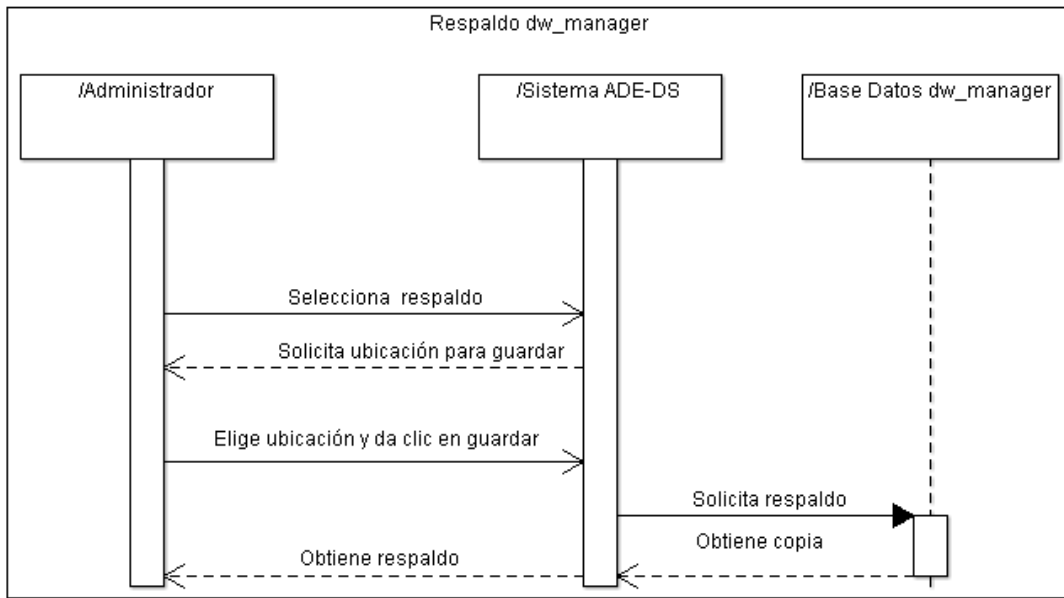


Figura 5.15 Diagrama de secuencia Respaldo el dw_manager.
Fuente: creación propia

En la Figura 5.16 se muestra el diagrama de secuencia para guardar metadatos del sistema ADE-DS.

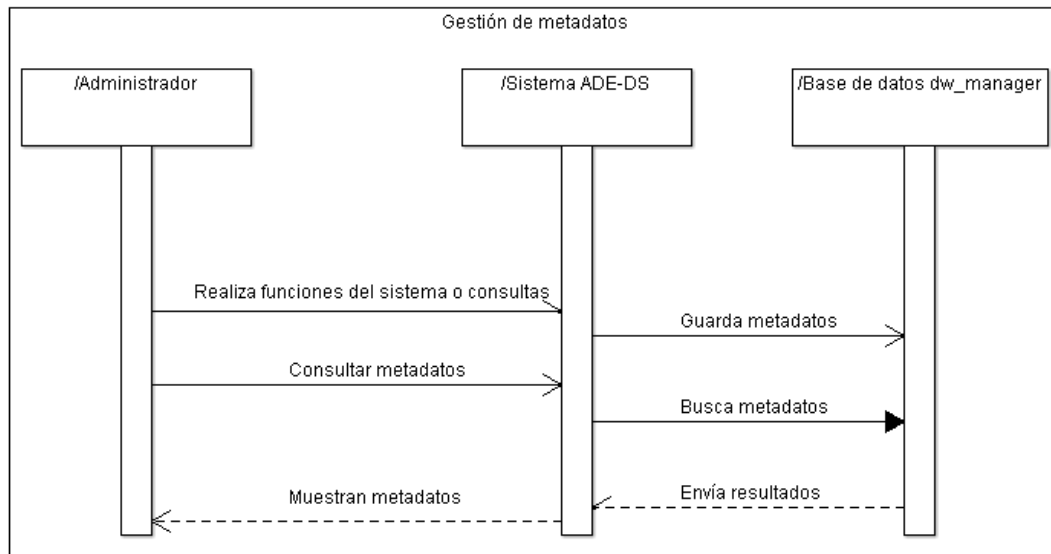


Figura 5.16 Diagrama de secuencia Guardar metadatos.
Fuente: creación propia

5.2.3 Diagrama de clases

En el diagrama de clases de la Figura 5.17 se observa la estructura de las clases, métodos y objetos que conforman el sistema ADE-DS.

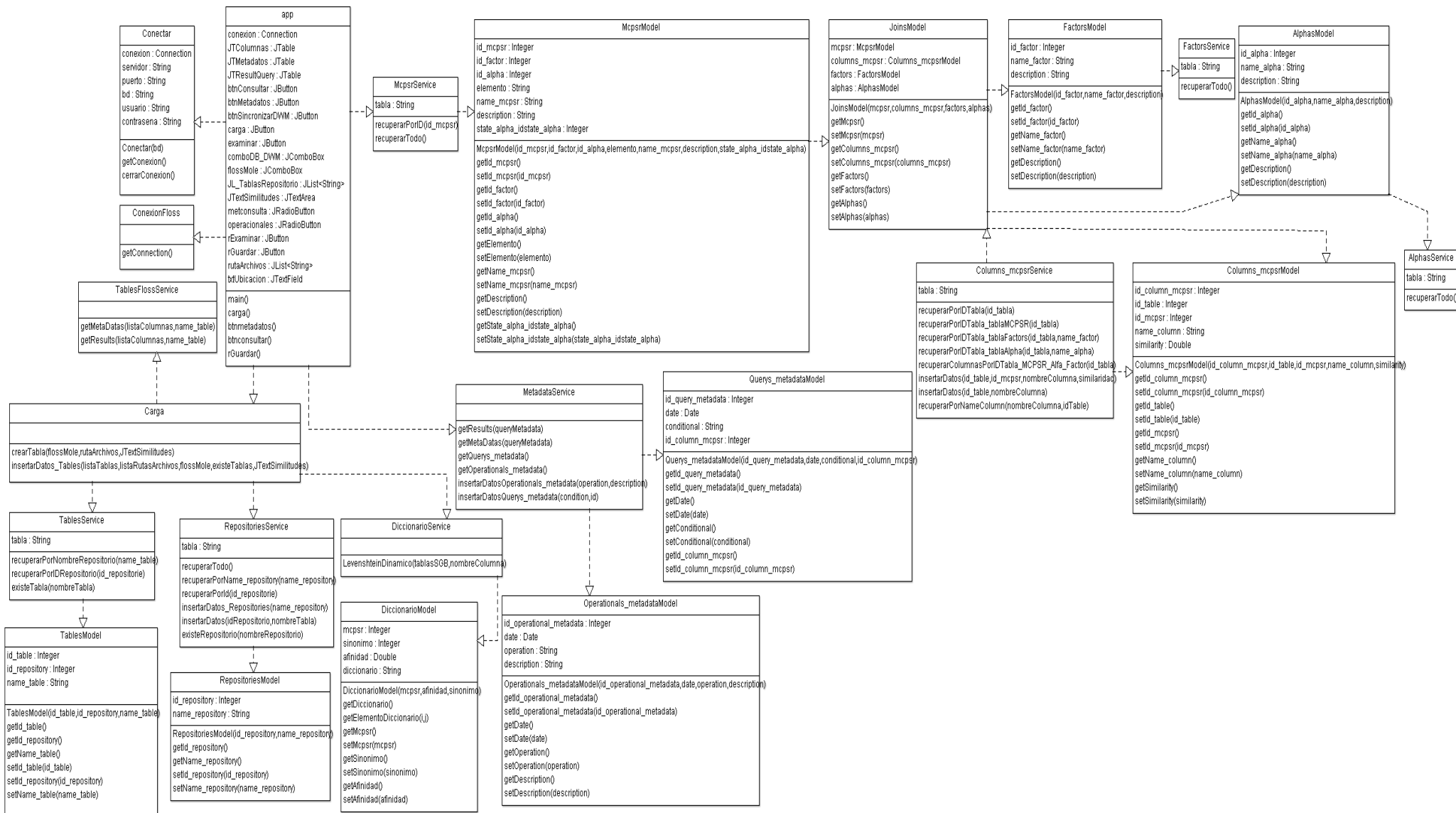


Figura 5.17 Diagrama de clases.
Fuente: creación propia

app: Clase principal de la herramienta, llama a las clases: Conectar, ConexionFloss, Carga, MetadataService, McpsrService.

Conectar: Clase encargada de establecer una conexión con la base de datos local dw_manager.

ConexionFloss: Clase encargada de establecer una conexión con la base de datos de los repositorios FLOSSmole.

Carga: Clase encargada de realizar la carga de un CSV a la base de datos local del dw_manager y devuelve las similitudes de archivo con el MCPS-R. Llama a las clases: TablesFLossService, TablesService, RepositoriesService, DiccionarioService.

MetadataService: Clase encargada de obtener los metadatos operacionales y de consulta dentro del sistema para guardarlos en la base de datos dw_manager. Llama a las clases: Operationals_metadataModel, Querys_metadataModel.

McpsrService: Clase encargada de obtener los datos de las consultas generales realizadas a la base de datos local dw_manager. Llama a la clase McpsrModel.

En este capítulo se desarrollaron diagramas GRL y UCM para especificar los requerimientos de la herramienta ADE-DS, en base a estos requerimientos se desarrollaron diagramas UML para diseñar el funcionamiento de la herramienta. En el siguiente capítulo se describe como son implementadas las funciones para los servicios del administrador de almacén de datos.

Capítulo 6

Implementación

En este capítulo, se presenta la construcción del sistema ADE-DS, que soporta la administración del almacén de datos `dw_manager`. El ADE-DS es el resultado de esta investigación y en este capítulo también se describe como se utilizan sus funcionalidades, mediante su interfaz de usuario.

A continuación, se presenta la estructura del capítulo:

6.1 Cargar repositorios FLOSSmole hacia el `dw_manager` y Análisis de similitudes

6.2 Guardar metadatos

6.3 Consultar el `dw_manager`

6.4 Respaldar el `dw_manager`

6.5 Interfaz de usuario

6.1 Cargar repositorios FLOSSmole hacia el `dw_manager` y Análisis de similitudes

La carga de los archivos `.csv`, que contienen datos provenientes de los repositorios FLOSSmole, hacia el almacén de datos `dw_manager`, se realiza mediante la lectura del archivo `.csv`. Un archivo de valores separados por comas (CSV) es solo un archivo de texto normal, almacena los datos columna por columna y lo divide por un separador (por ejemplo, normalmente es una coma “,”).

A través de la librería OpenCSV para Java, se realiza la lectura del archivo `.CSV` para crear la estructura de la tabla del repositorio elegido en el almacén de datos. Al mismo tiempo se realiza un análisis de similitudes con el MCPS-R, en la Figura 6.1 se observa de manera visual el proceso desarrollado en esta investigación para llevar a cabo la carga de archivos `.csv`.

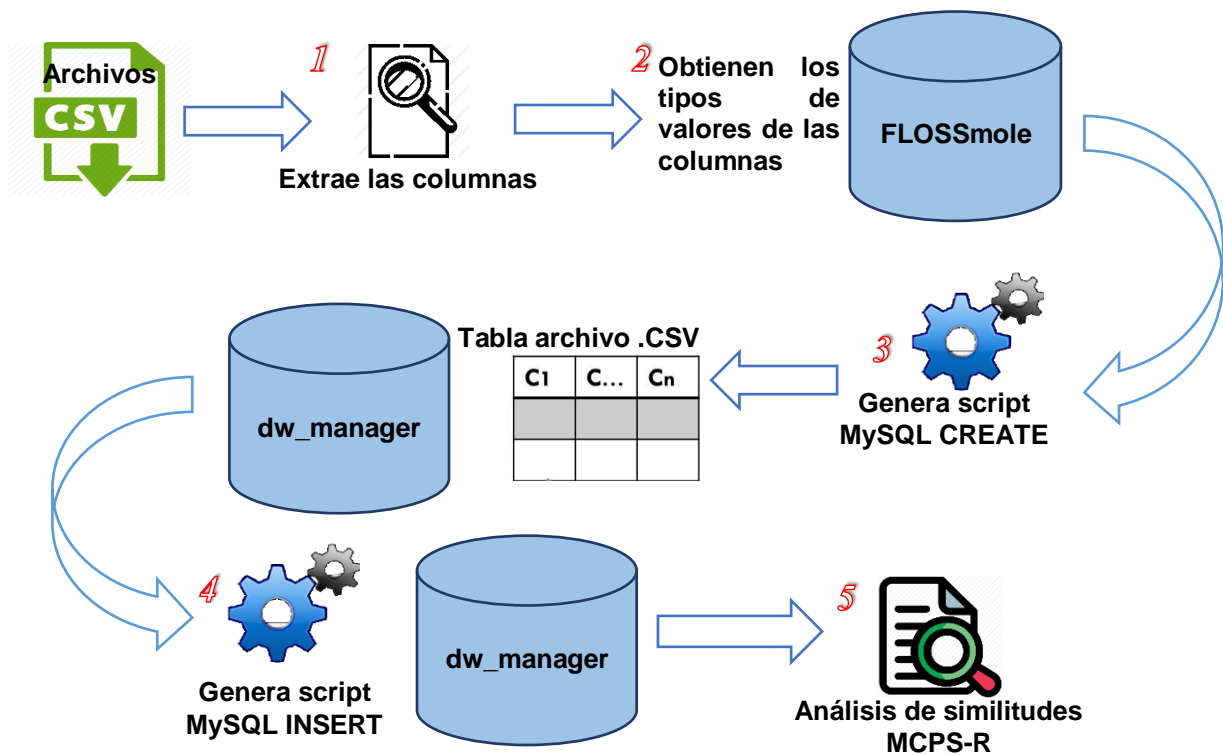


Figura 6.1 Proceso para cargar repositorios FLOSSmole hacia el dw_manager
Fuente: creación propia

A continuación, se describen los 5 pasos representados en la Figura 6.1:

1. Con el uso de la librería OpenCSV se abre el archivo .csv del repositorio y se extraen las columnas.
2. Se analizan las columnas y se obtienen los tipos de valores de cada una, a través de una conexión a la base de datos FLOSSmole.
3. Se realizan los scripts de create para crear la estructura de la tabla en el almacén de datos, el cual es ejecutado.
4. Una vez creada la estructura o tabla en el almacén de datos se generan los scripts de insert para insertar el contenido de los registros de dicho archivo .csv y se ejecuta.
5. Al finalizar la carga del archivo o archivos .csv se muestra un análisis de similitudes con el MCPS-R realizado.

El análisis de similitudes con el MCPS-R, se realiza mediante el uso del algoritmo de Levenshtein con la finalidad de obtener el grado de similitud. Para realizar dicho análisis se creó un diccionario de datos MCPS-R mostrado en el Anexo E y un ejemplo detallado de como utilizar el algoritmo.

6.2 Guardar metadatos

La inserción de los metadatos operacionales realizados con la herramienta ADE-DS y la inserción de los metadatos de las consultas generales realizadas, fue posible mediante el uso de dos scripts de inserción SQL generados automáticamente mostrados en las Figuras 6.2 y 6.3.

```
"INSERT INTO operationals_metadata (date, operational,
description) VALUES (now (), "" + operation + "", "" + description + "");"
```

Figura 6.2 Script de inserción SQL metadatos operacionales
Fuente: creación propia

```
"INSERT INTO querys_metadata (date, conditional, id_column_mcpsr)
VALUES (now (), "" + condition + "", "" + id + "");"
```

Figura 6.3 Script de inserción SQL metadatos de consultas
Fuente: creación propia

De igual manera para las consultas de los metadatos operacionales y de consulta con la herramienta ADE-DS, se realizó dos scripts de consulta SQL para acceder a los datos guardados en la base de datos local dw_manager como se muestra en las Figuras 6.4 y 6.5.

```
"SELECT * FROM operationals_metadata;"
```

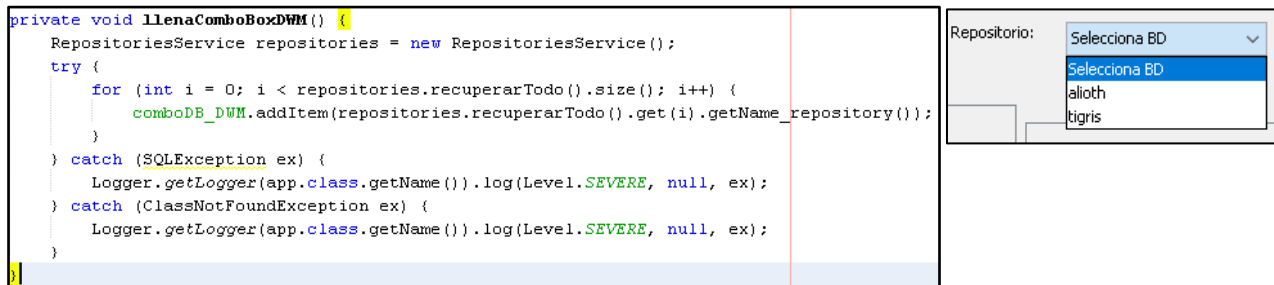
Figura 6.4 Script de consulta SQL metadatos operacionales
Fuente: creación propia

```
"SELECT T1.date, T7.name_repository, T6.name_table, T2.name_column, T3.elemento,
T3.name_mcpsr, T4.name_factor, T5.name_alpha\n" +
"FROM dw_manager.querys_metadata as T1\n" +
"LEFT OUTER JOIN dw_manager.columns_mcpsr as T2\n" +
"ON T1.id_column_mcpsr = T2.id_column_mcpsr\n" +
"LEFT OUTER JOIN dw_manager.mcpsr as T3\n" +
"ON T3.id_mcpsr = T2.id_mcpsr\n" +
"LEFT OUTER JOIN dw_manager.factors as T4\n" +
"ON T4.id_factor = T3.id_factor\n" +
"LEFT OUTER JOIN dw_manager.alphas as T5\n" +
"ON T3.id_alpha = T5.id_alpha\n" +
"LEFT OUTER JOIN dw_manager.tables as T6\n" +
"ON T2.id_table = T6.id_table\n" +
"LEFT OUTER JOIN dw_manager.repositories as T7\n" +
"ON T7.id_repository = T6.id_repository;"
```

Figura 6.5 Script de consulta SQL metadatos de consulta
Fuente: creación propia

6.3 Consultar el dw_manager

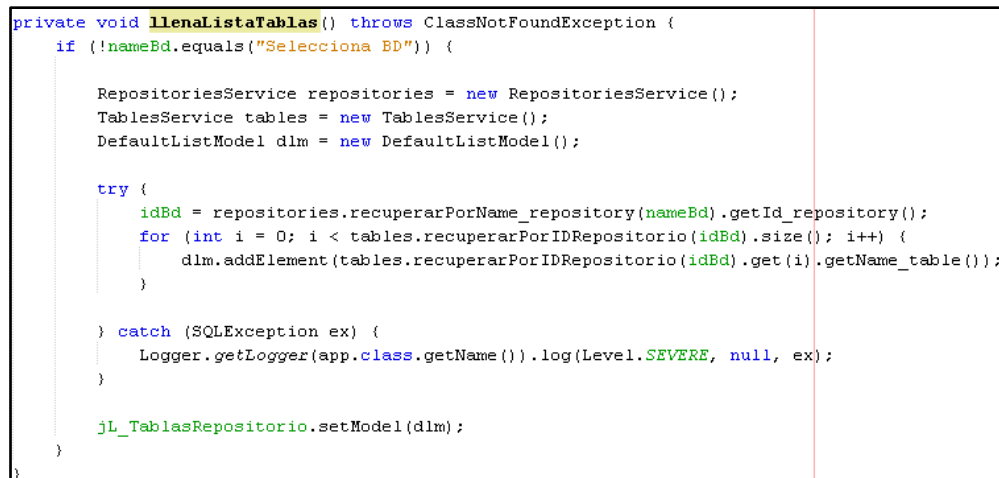
Las consultas generales, se realiza mediante la sincronización a la base de datos local dw_manager lo cual nos permite llenar un ComboBox con las bases de datos FLOSSmole que se encuentren agregadas en la Figura 6.6 se visualiza parte del código que nos permite ejecutar dicha acción y el resultado.



```
private void llenaComboBoxDWM() {
    RepositoriesService repositories = new RepositoriesService();
    try {
        for (int i = 0; i < repositories.recuperarTodo().size(); i++) {
            comboDB_DWM.addItem(repositories.recuperarTodo().get(i).getName_repository());
        }
    } catch (SQLException ex) {
        Logger.getLogger(app.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(app.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Figura 6.6 ComboBox repositorios cargados en dw_manager
Fuente: creación propia

Al seleccionar el repositorio FLOSSmole a consultar se llena una lista con las tablas cargadas en la base de datos local dw_manager de dicho repositorio como se muestra en la Figura 6.7 parte del código utilizado para dicha acción.



```
private void llenarListaTablas() throws ClassNotFoundException {
    if (!nameBd.equals("Selecciona BD")) {
        RepositoriesService repositories = new RepositoriesService();
        TablesService tables = new TablesService();
        DefaultListModel dlm = new DefaultListModel();

        try {
            idBd = repositories.recuperarPorName_repository(nameBd).getId_repository();
            for (int i = 0; i < tables.recuperarPorIDRepositorio(idBd).size(); i++) {
                dlm.addElement(tables.recuperarPorIDRepositorio(idBd).get(i).getName_table());
            }
        } catch (SQLException ex) {
            Logger.getLogger(app.class.getName()).log(Level.SEVERE, null, ex);
        }

        jL_TablasRepositorio.setModel(dlm);
    }
}
```

Figura 6.7 Código para llenar la lista de tablas de repositorios FLOSSmole
Fuente: creación propia

Al seleccionar cada una de las tablas se visualiza de forma general el nombre de las columnas de dicha tabla y a que atributo del MCPS-R es similar como se muestra en la Figura 6.8.

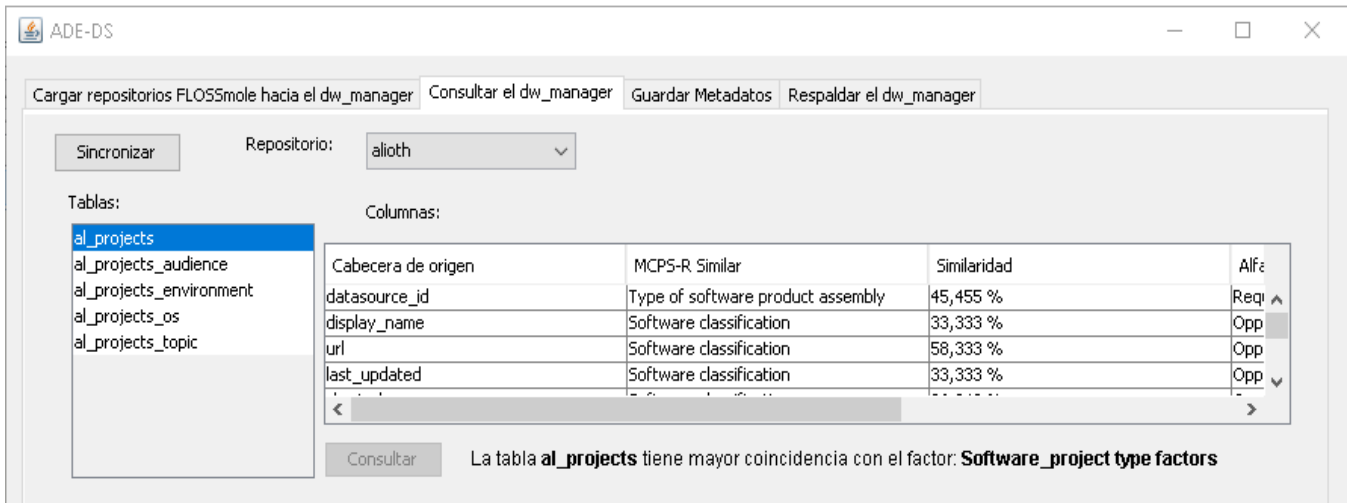


Figura 6.8 Tablas y columnas del repositorio seleccionado
Fuente: creación propia

Para visualizar la información de las tablas de los repositorios FLOSSmole es necesario seleccionar una o todas las columnas que se quieran consultar esto nos permitirá obtener los datos mediante la columna del repositorio, como se muestra en la Figura 6.9.

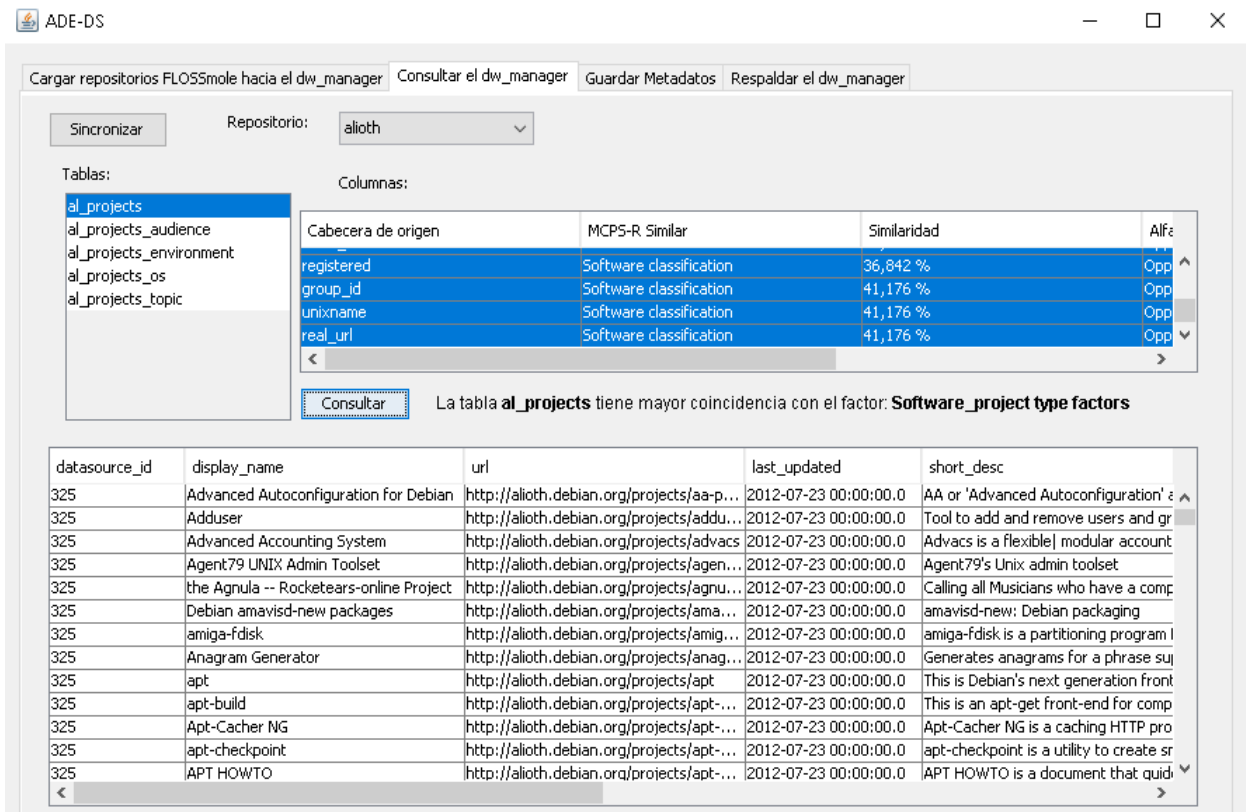


Figura 6.9 Ejemplo de consulta
Fuente: creación propia

6.4 Respaldar el dw_manager

El respaldo se realizó mediante el uso del comando mysqldump del sistema gestor de base de datos MySQL que nos permite realizar copias de seguridad (backup) de nuestra base de datos dw_manager como se muestra en la Figura 6.10.

```
Runtime runtime = Runtime.getRuntime();
p = runtime.exec("C:/Program Files/MySQL/MySQL Workbench 6.3 CE/mysqldump.exe -uroot -proot "
+ "--add-drop-database -B dw_manager -r" + replace);
```

Figura 6.10 Código para ejecutar el mysqldump

Fuente: creación propia

6.5 Interfaz de usuario

En la Figura 6.11, se observa la interfaz que se muestra por defecto al ejecutar la herramienta ADE-DS. Esta interfaz cuenta con las 4 pestañas para realizar las operaciones de cargar repositorios FLOSSmole hacia el dw_manager, consultar el dw_manager, guardar metadatos y respaldar el dw_manager.

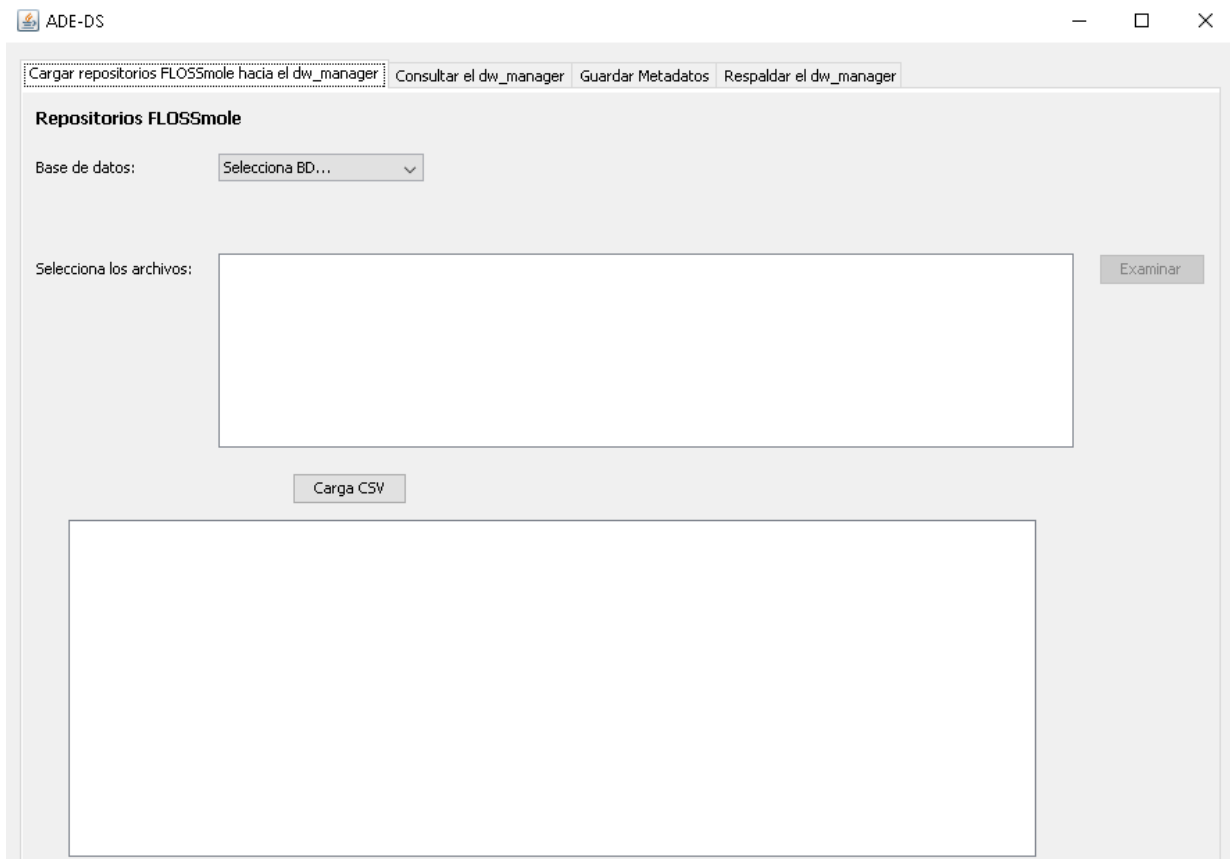


Figura 6.11 Interfaz ADE-DS

Fuente: creación propia

Lo primero que se realiza en la pestaña de cargar repositorios FLOSSmole hacia el dw_manager es seleccionar la base de datos FLOSSmole que se desea cargar, una vez realizado esto se activa el botón de examinar como se muestra en la Figura 6.12 el cual nos permite elegir los archivos CSV a cargar en nuestra base de datos dw_manager.

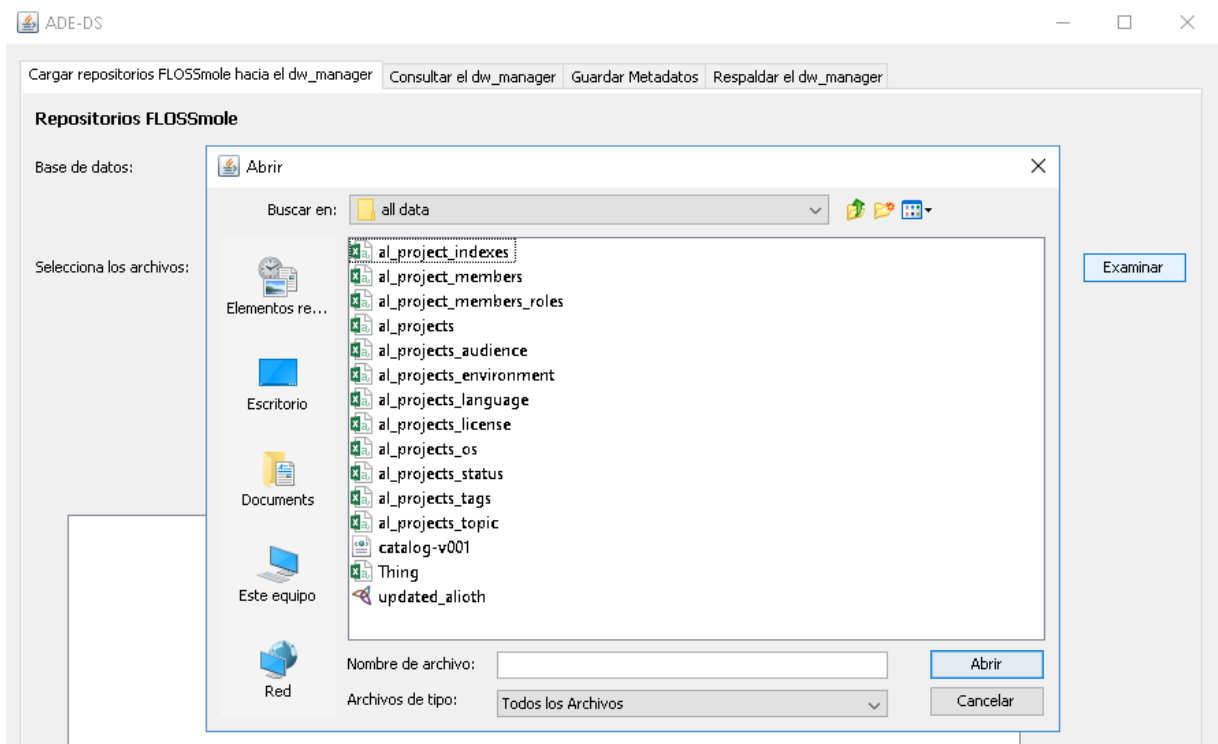


Figura 6.12 Cuadro de diálogo para abrir archivos CSV
Fuente: creación propia

Para realizar la carga de los archivos CSV es necesario dar clic en el botón de Carga CSV el cual al finalizar el proceso nos mostrará un análisis de similitudes encontrado con el MCPS-R como se muestra en la Figura 6.13.

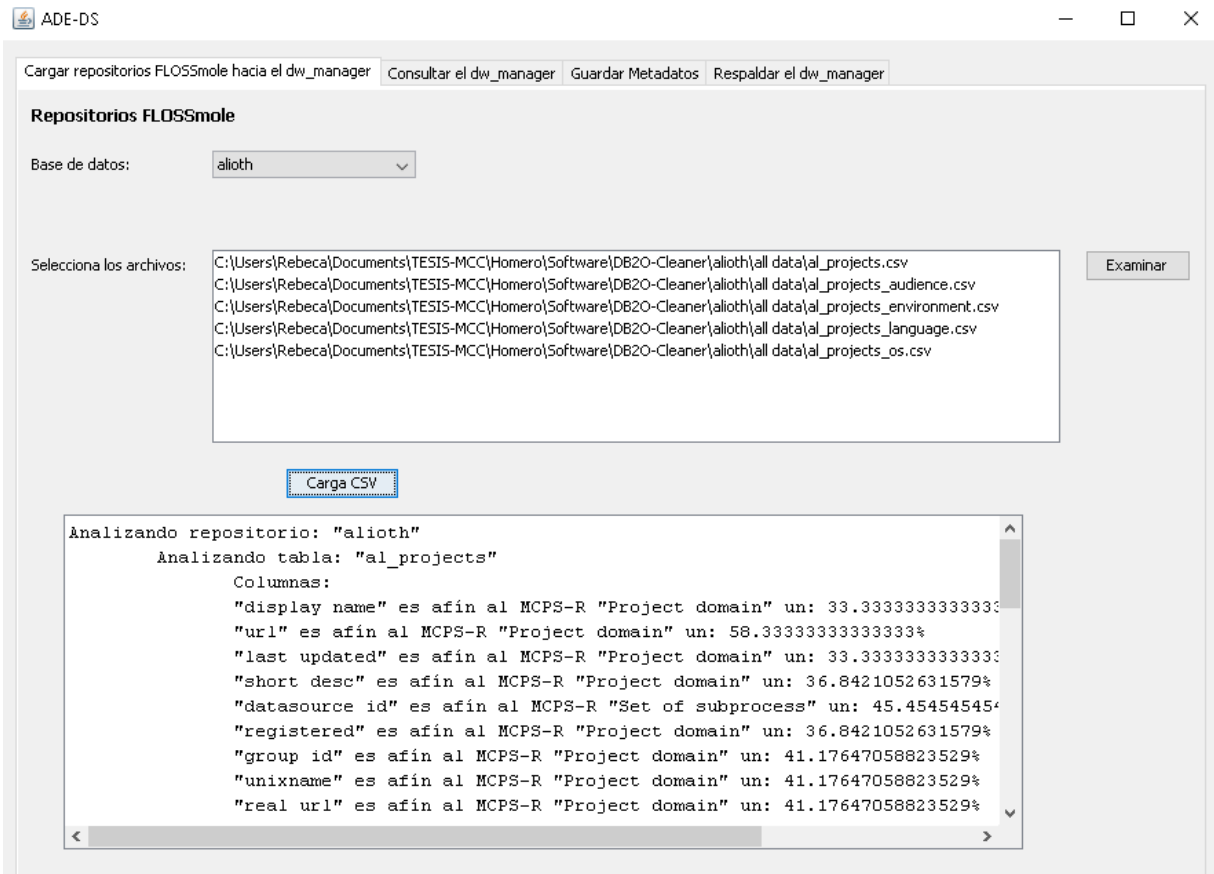


Figura 6.13 Cargar repositorios FLOSSmole hacia el dw_manager y análisis de similitudes
 Fuente: creación propia

Para realizar una consulta lo primero que debemos realizar es dar clic en el botón Sincronizar, esto nos permitirá hacer la conexión a la base de datos dw_manager y se activará el combobox de repositorio para mostrar las bases de datos FLOSSmole cargadas en nuestra base de datos dw_manager como se muestra en la Figura 6.14.

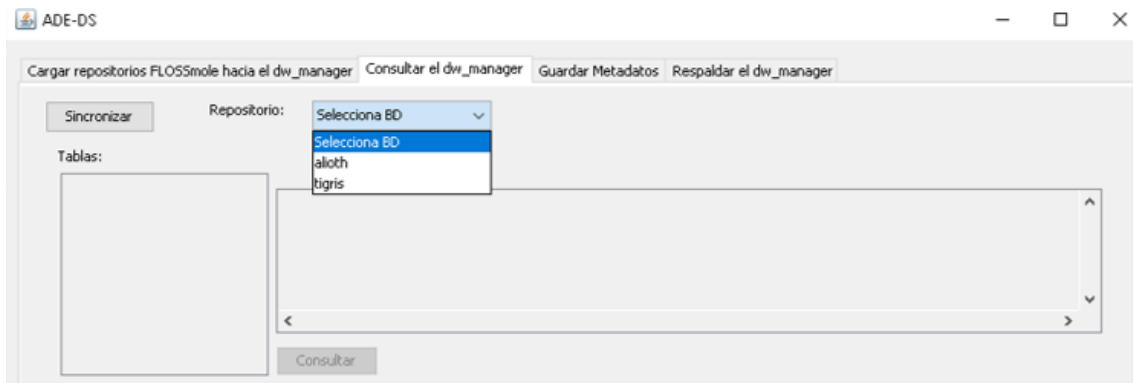


Figura 6.14 Bases de datos FLOSSmole cargadas en dw_manager
 Fuente: creación propia

Al seleccionar la base de datos se mostrarán las tablas cargadas de ese repositorio, al seleccionar la tabla del lado derecho se muestra en Columnas su estructura y a que MCPS-R pertenecen como se muestra en la Figura 6.15.

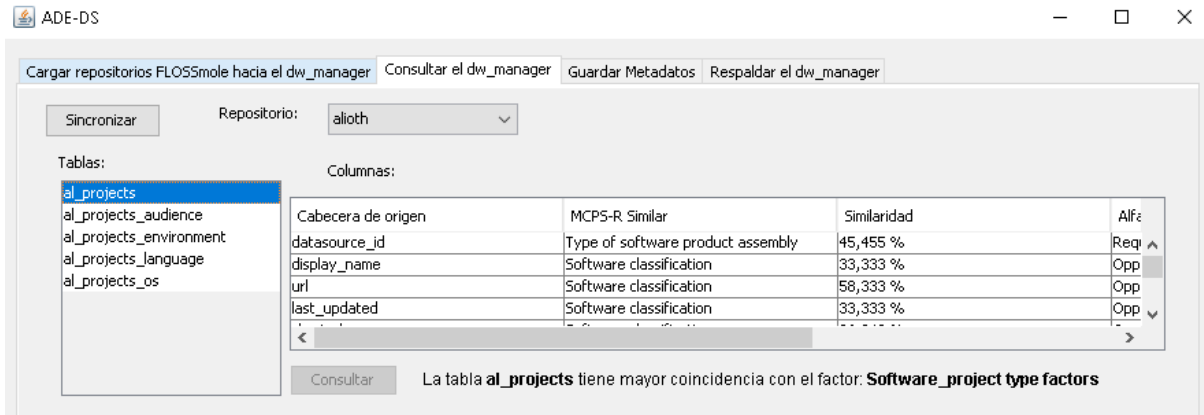


Figura 6.15 Tablas y columnas de repositorio seleccionado

Fuente: creación propia

Para consultar los datos de la tabla seleccionada, elegimos todas las columnas que deseamos consultar y damos clic en el botón Consultar. Una vez finalizada la consulta se muestra en la Figura 6.16 los datos de la tabla seleccionada.

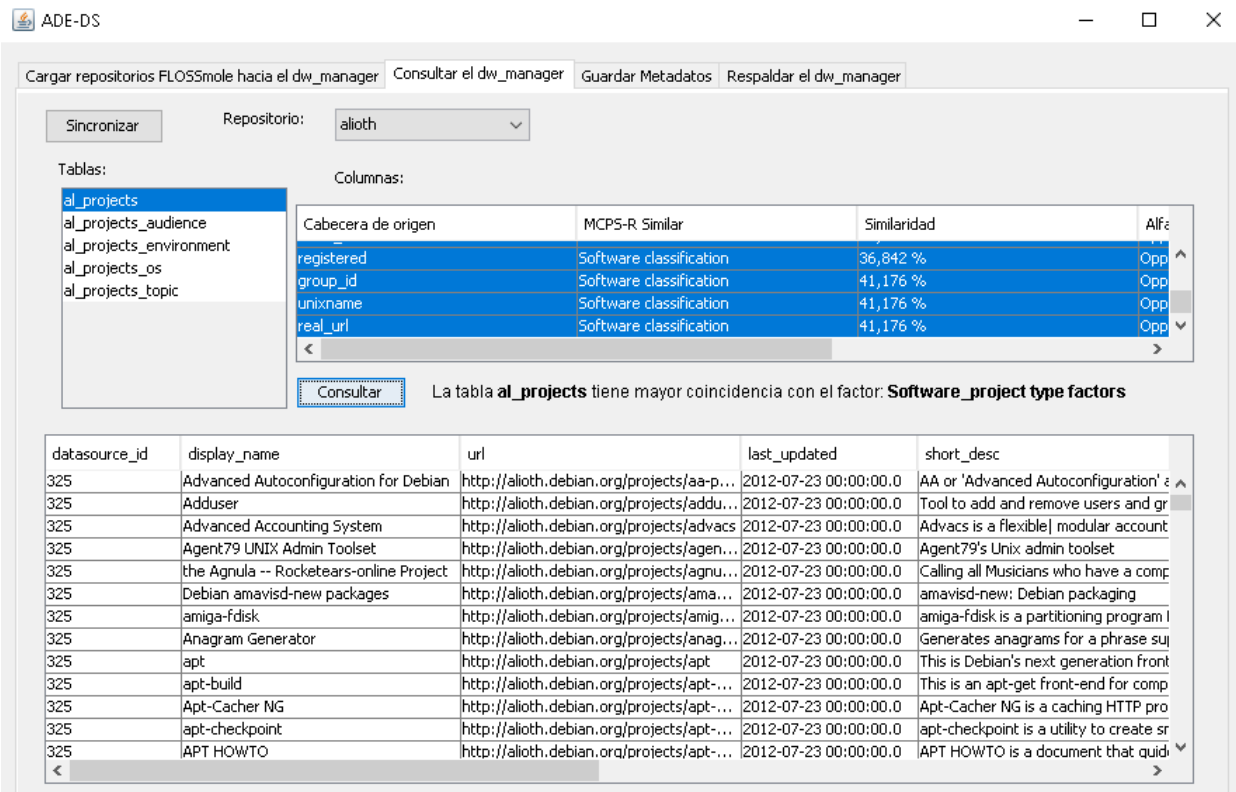


Figura 6.16 Sección consultar el dw_manager

Fuente: creación propia

En la ventana de guardar metadatos se encuentran dos opciones para consultar, operacionales y de consulta. Seleccionamos primero el metadato a consultar y damos clic en consultar al finalizar consultar metadatos se mostrar los datos operacionales o de consulta encontrados como se muestra en la Figura 6.17.

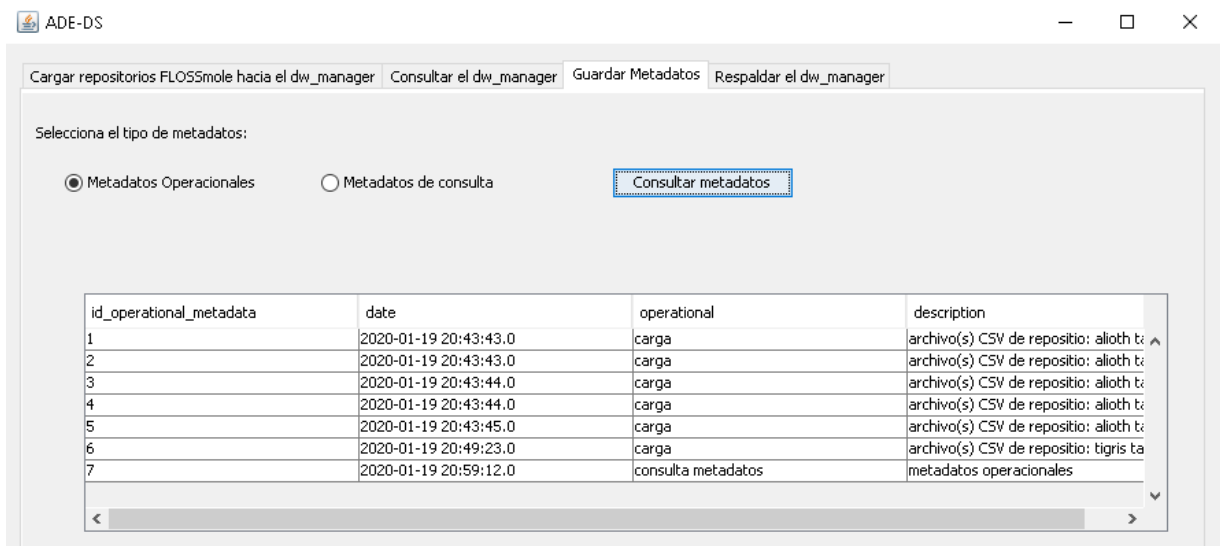


Figura 6.17 Sección de guardar metadatos
Fuente: creación propia

En la ventana de respaldar el dw_manager primero damos clic en el botón examinar para elegir la ubicación de donde guardaremos nuestro respaldo de la base de datos. Teniendo la ubicación para nuestro respaldo damos clic en el botón Guardar respaldo, al finalizar se mostrará el mensaje de respaldo exitoso como se muestra en la Figura 6.18.

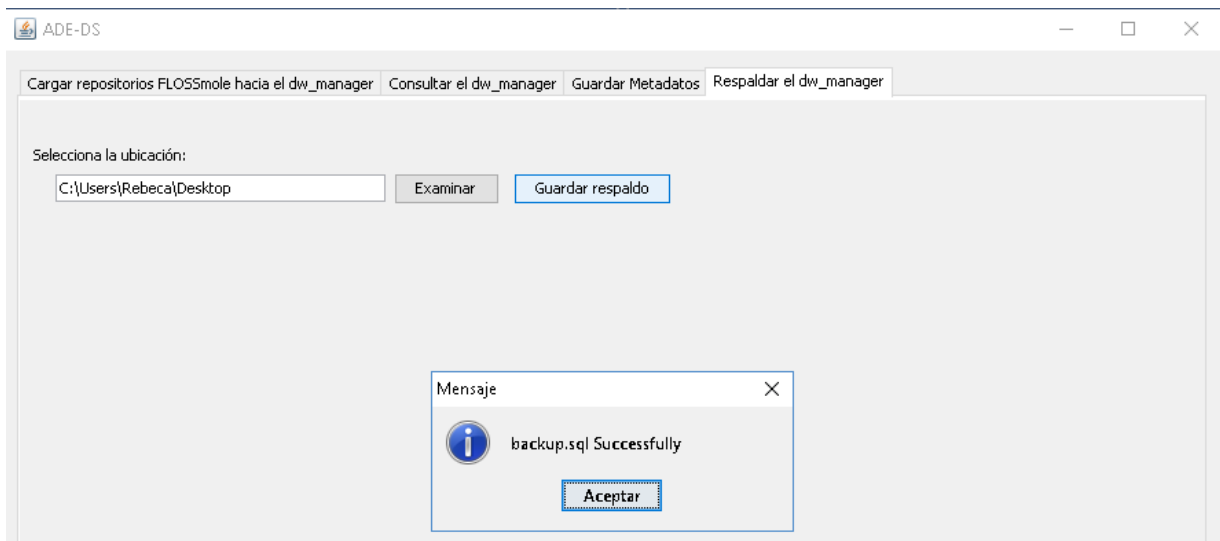


Figura 6.18 Sección de respaldar el dw_manager
Fuente: creación propia

En este capítulo se presentó la construcción del sistema ADE-DS, que soporta la administración del almacén de datos dw_manager, así como la interfaz y descripción del sistema. En el siguiente capítulo se describe como es realizado el plan de pruebas hasta el reporte de resultados obtenido.

Capítulo 7

Pruebas y Resultados

En este capítulo se presentan los documentos generados, desde el plan de pruebas hasta el reporte de resultados, para las pruebas de la herramienta ADE-DS y se describe como fue ejecutado cada caso de prueba. También, se presentan las anomalías arrojadas durante la ejecución de las pruebas y se presentan los resultados obtenidos después de haber corregido dichas anomalías.

A continuación, se presenta la estructura del capítulo:

- 7.1 Plan de pruebas
 - 7.1.1 Elementos de prueba
 - 7.1.2 Requerimientos a probar
 - 7.1.3 Matriz de Trazabilidad de pruebas
 - 7.1.4 Criterio de aceptación de pruebas
 - 7.1.5 Criterio de no aceptación de pruebas
 - 7.1.6 Características del ambiente de pruebas
- 7.2 Diseño de Pruebas
 - 7.2.1 Características a probar
 - 7.2.2 Enfoque de pruebas
- 7.3 Casos de Prueba
 - 7.3.1 Especificaciones de caso de prueba TC-01
- 7.4 Procedimiento de Pruebas
 - 7.4.1 Procedimiento de prueba TPr-01
- 7.5 Registro de Pruebas
 - 7.5.1 Registro de prueba TL-01
- 7.6 Reporte de Pruebas
 - 7.6.1 Resultados generales
 - 7.6.2 Resultados detallados

7.1 Plan de pruebas

Esta sección describe los elementos a probar y proporciona una matriz de trazabilidad de pruebas, la cual relaciona los elementos a probar con los requerimientos especificados en el Capítulo 5 Requerimientos y Diseño del sistema. Además, se presentan las características del ambiente de pruebas y se describen los criterios de aceptación y suspensión de pruebas.

7.1.1 Elementos de prueba

Se prueba la funcionalidad de cada uno de los módulos de software que conforman la herramienta ADE-DS:

Cargar repositorios FLOSSmole hacia el dw_manager
 Consultar el dw_manager
 Guardar metadatos
 Respaldar el dw_manager

7.1.2 Requerimientos a probar

A continuación, se describen los requerimientos de la herramienta ADE-DS que presenta cada uno de los elementos de prueba:

1.- Cargar repositorios FLOSSmole hacia el dw_manager. Este requerimiento permite cargar los archivos CSV obtenidos de los repositorios FLOSSmole hacia la base de datos dw_manager. Así como mostrar las similitudes que tienen los nombres de los campos de los archivos CSV cargados en la base de datos con el nombre de los campos del Marco para la Caracterización de Proyectos de Software Refinado (MCPS-R).

2.- Consultar el dw_manager. Este requerimiento permite al usuario consultar la información de la base de datos dw_manager. Eligiendo un repositorio FLOSSmole y tabla de los que se encuentran cargados en la base de datos, para mostrar sus campos en base al MCPS-R y Essence. Consultando también los datos contenidos en la tabla.

3.- Guardar metadatos. Este requerimiento permite guardar los metadatos operacionales y de consulta realizados con el sistema DW-Manager, en la base de datos, así como la visualización de los mismos.

4.- Respaldar el dw_manager. Este requerimiento permite generar el respaldo total de la base de datos dw_manager eligiendo la ubicación donde se guardará.

7.1.3 Matriz de Trazabilidad de pruebas

En la Tabla 7.1, se presentan los casos de prueba (TC) a ejecutar y el requerimiento y elemento de prueba que corresponden a cada uno de ellos.

Tabla 7.1. Matriz de trazabilidad de pruebas.
 Fuente: creación propia.

Requerimiento	Caso de Prueba
1. Cargar repositorios FLOSSmole hacia el dw_manager	TC-01
2. Consultar el dw_manager	TC-02
3. Guardar metadatos	TC-03
4. Respaldar el dw_manager	TC-04

7.1.4 Criterio de aceptación de pruebas

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

7.1.5 Criterio de no aceptación de pruebas

La suspensión de las pruebas está regida por la presencia de fallas en los componentes que impidan el cumplimiento de sus funcionalidades, al registrarse una falla se suspende la prueba y se reanuda cuando se corrija el error que ocasionó la falla.

7.1.6 Características del ambiente de pruebas

El ambiente de pruebas presenta las siguientes características:

Software

- Sistema Operativo: Windows 10 Home de 64 bits
- Gestor de Base de Datos: MySQL 5.7
- MySQL Connector Java 5.1.11
- Java 1.8

Hardware

- Procesador: Intel® Core™ i7-7500U 2.70 GHz.
- Memoria RAM: 8 GB
- Disco duro: 1 TB

7.2 Diseño de Pruebas

En esta sección se describen las características a probar en cada uno de los casos de prueba y se establece el enfoque para ejecutar las pruebas.

7.2.1 Características a probar

En la Tabla 7.2, se presentan las características a probar, estas características se obtuvieron a partir de los requerimientos de la herramienta ADE-DS. Además, se observa que cada procedimiento de prueba (TPr) corresponde a un caso de prueba (TC) con sus respectivas características.

Tabla 7.2. Características a probar.

Fuente: creación propia.

Requerimiento	Característica	Caso de Prueba	Procedimiento de Prueba
1. Cargar repositorios FLOSSmole	1. Solicitar repositorio FLOSSmole 2. Solicitar archivo(s) .csv de tablas FLOSSmole	TC-01	TPr-01

hacia el dw_manager	3. Mostrar similitudes con el MCPS-R		
2. Consultar el dw_manager	1. Sincronizar con la base de datos dw_manager 2. Solicitar repositorio FLOSSmole que se encuentre en la base dw_manager 3. Solicitar tabla de repositorio FLOSSmole a consultar 4. Mostrar datos de las tablas consultadas	TC-02	TPr-02
3. Guardar metadatos	1. Seleccionar metadato operacional o de consulta 2. Mostrar los metadatos consultados	TC-03	TPr-03
4. Respaldar el dw_manager	1. Solicitar ubicación para guardar respaldo 2. Genera un backup de la base de datos dw_manager	TC-04	TPr-04

7.2.2 Enfoque de pruebas

Las pruebas se ejecutarán usando los datos de los repositorios seleccionados en el Capítulo 4 Diseño del almacén de datos sección 4.1 Repositorios FLOSSmole (Alioth, RubyForge y Tigris), para probar la funcionalidad de cada uno de los elementos de prueba.

7.3 Casos de Prueba

En esta sección se especifican los casos de prueba que se llevarán a cabo para probar las funcionalidades de la herramienta ADE-DS. Para cada caso de prueba se incluyen: un identificador para cada caso de prueba, su objetivo, sus entradas y sus salidas. Necesarios para llevar a cabo su ejecución.

7.3.1 Especificaciones de caso de prueba TC-01

A continuación, se describe el objetivo, la entrada y salida del caso de prueba TC-01. Los demás casos de prueba se localizan en el Anexo F.

ID: TC-01

Objetivo: El objetivo de este caso de prueba es validar el cumplimiento del requerimiento Cargar repositorios FLOSSmole hacia el dw_manager.

Entradas: La entrada son los archivos .csv de los repositorios FLOSSmole. Seleccionando, mediante el botón izquierdo del ratón, el repositorio, ver Figura 7.1 y los archivos .csv a cargar, ver Figura 7.2.

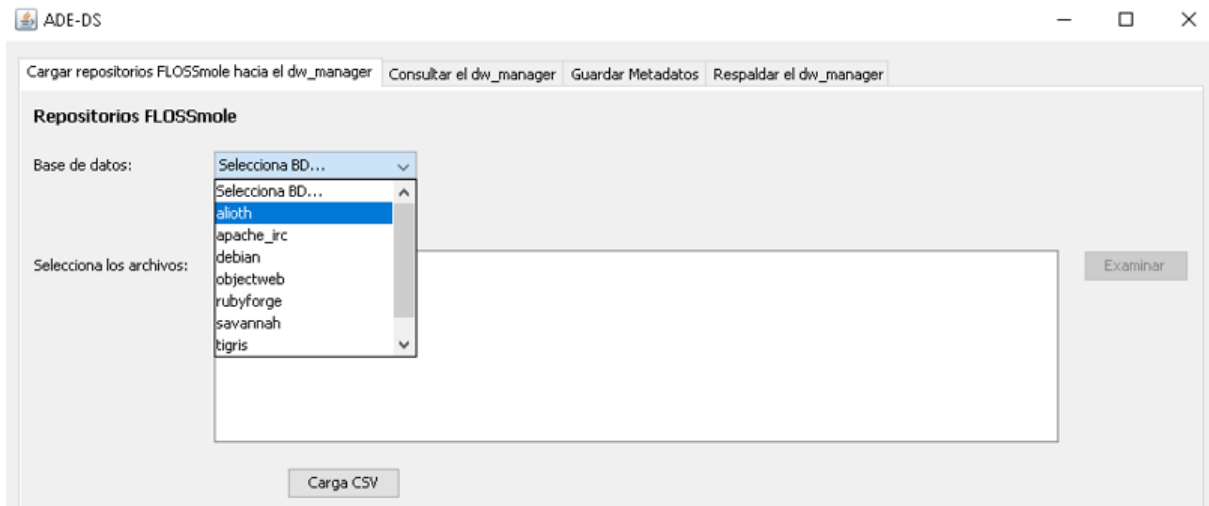


Figura 7.1 Selección de repositorio FLOSSmole a cargar
Fuente: creación propia

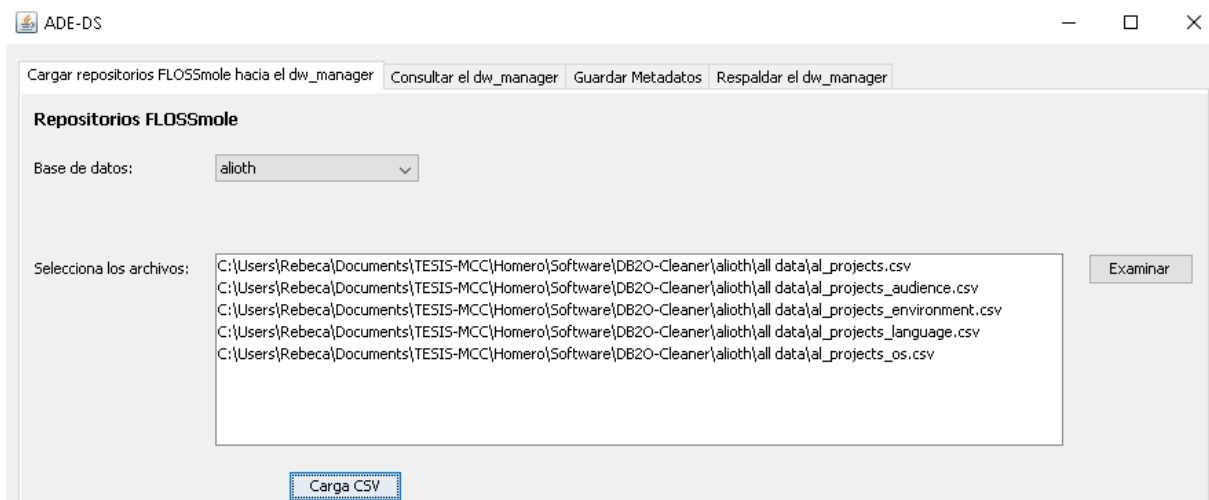


Figura 7.2 Datos de entrada para el primer requerimiento funcional Carga
Fuente: creación propia

Salidas: La salida es la carga de los archivos .csv hacia la base de datos dw_manager y las similitudes con el MCPS-R de los campos de cada archivo .csv, ver Figura 7.3.

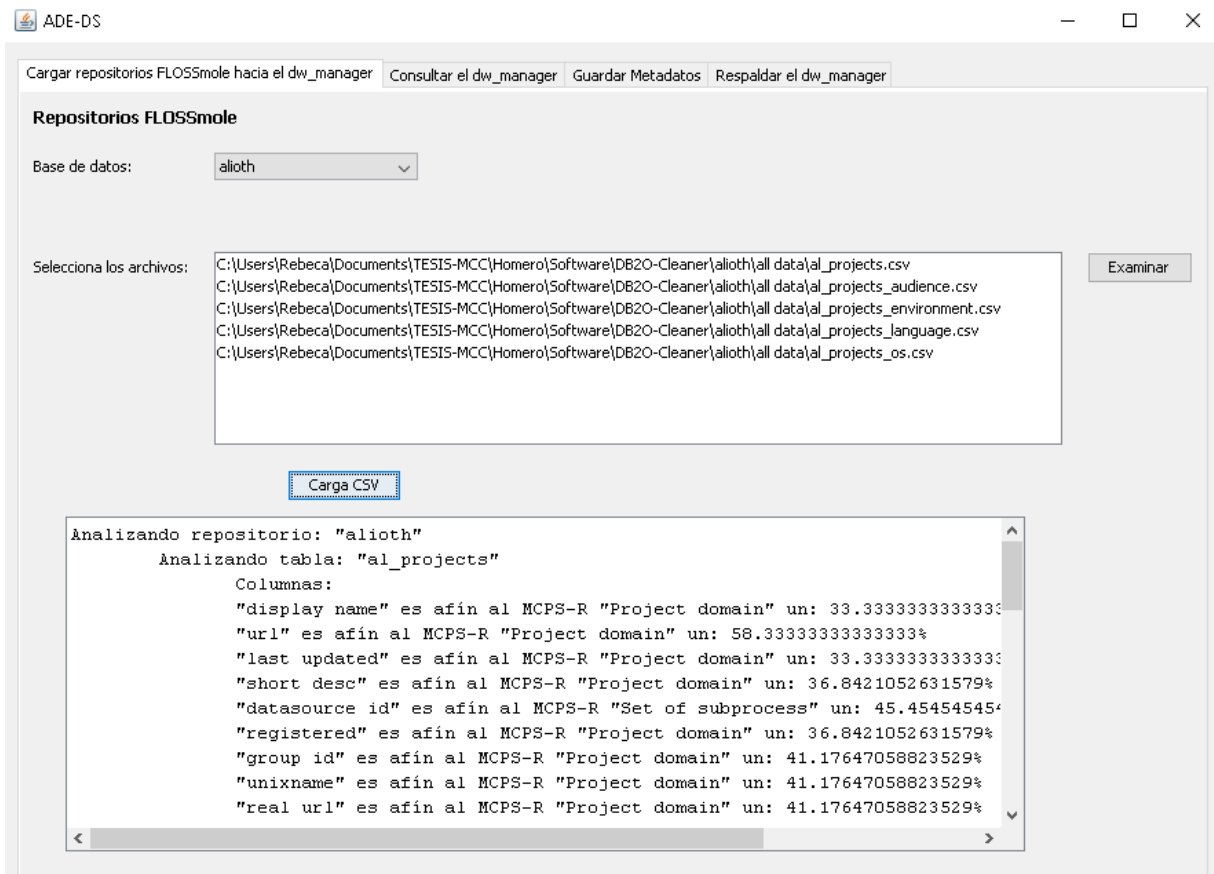


Figura 7.3 Similitudes con el MCPS-R de los archivos .csv cargados
Fuente: creación propia

7.4 Procedimiento de Pruebas

En esta sección se describen los diferentes procedimientos a seguir para llevar a cabo la ejecución de cada uno de los casos de prueba, de manera que se cumpla con las funcionalidades de la herramienta ADE-DS. A continuación, se muestra el procedimiento de prueba TPr-01 correspondiente al caso de prueba TC-01, los demás procedimientos de pruebas se localizan en el Anexo F.

7.4.1 Procedimiento de prueba TPr-01

Entradas, salidas y requerimientos especiales

Documentos:

Plan de prueba

Diseño de prueba

Procedimiento de Prueba

Reporte de Prueba

Servidores:

MySQL

Aplicaciones:

XAMPP

Archivos:

Archivos .csv de repositorios FLOSSmole

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

Reportes: La elaboración de los reportes es de forma manual, reportando el resultado de hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba, en el documento de reporte de pruebas. De esta misma forma los errores, problemas o anomalías se registrarán en el mismo documento.

Preparación: Iniciar todas las aplicaciones y servidores, antes de comenzar con el procedimiento.

Inicio: En la interfaz de la herramienta, se selecciona el repositorio FLOSSmole y se habilita, mediante el botón izquierdo del ratón, el botón de examinar.

Proceso: El usuario debe seleccionar los archivos .csv del repositorio FLOSSmole a cargar en la base de datos dw_manager.

Medición: El momento de realizar la medición, depende del tiempo que el usuario se tome en seleccionar los archivos .csv que requiere.

Suspender: El componente Cargar repositorios FLOSSmole hacia el dw_manager se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

Reiniciar: La prueba puede ser reiniciada inmediatamente después de finalizar su ejecución.

Cierre: La prueba termina con la carga de los archivos .csv seleccionados y el despliegue de las similitudes de los campos del MCPS-R con los campos de cada archivo .csv.

Contingencias: Al presentarse anomalías la prueba tiene que suspenderse y el error o fallo se corrige. Para poder reiniciar la prueba se debe haber corregido la anomalía.

7.5 Registro de Pruebas

Esta sección proporciona un registro cronológico de los detalles relevantes sobre la ejecución de las pruebas ejecutadas, para probar la funcionalidad de los elementos de prueba de la herramienta ADE-DS.

7.5.1 Registro de prueba TL-01

El registro de prueba TL-01 cubre el procedimiento de prueba TPr-01 correspondiente al caso de prueba TC-01. Los archivos usados en la ejecución de la prueba corresponden a los repositorios FLOSSmole, en este caso Alioth.

Actividades y eventos

A continuación, se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-01.

Paso	Actor	Descripción	Verificado
1	Usuario	Ejecuta la herramienta dw_manager	Si
2	Usuario	Selecciona el repositorio FLOSSmole a cargar	Si
3	Usuario	Hacer clic en examinar y seleccionar los archivos .csv a cargar	Si
4	Usuario	Hacer clic en cargar CSV	Si
5	Herramienta	Carga los archivos .csv en la base de datos dw_manager	Si
6	Herramienta	Genera y muestra las similitudes de las columnas de los archivos .csv con el MCPS-R	Si

Anomalías

A continuación, se describen las anomalías encontradas en las pruebas. La columna Paso contiene el número del paso de la lista anterior en el que se originó la anomalía. Se presenta una breve descripción, así como la solución.

Paso	Descripción	Solución
4	Se realizó el proceso de carga para algunos archivos .csv de los repositorios FLOSSmole y no permitió la carga exitosamente.	Se verificó el contenido de los archivos .csv que no se cargaron y se encontró algunos campos duplicados como last_update. Se procedió a eliminar el campo duplicado manualmente y guardar el cambio en el archivo .csv

En la Figura 7.4 se muestra un ejemplo de la anomalía encontrada en uno de los archivos .csv del repositorio object web y como manualmente se elimina ese campo duplicado para guardar el archivo .csv ver Figura 7.5.

	A	B	C	D	E	F	G	H	I
1	proj_unixnaris_admin		position	datasource	date_collected	date_collected	dev_loginname		
2	activexml	1	Project Mana	27	29/7/2006	29/7/2006 21:48	abitebou		
3	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	antonella		
4	activexml	1	Project Mana	27	29/7/2006	29/7/2006 21:48	benjello		
5	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	bogdanmarinoiu		
6	activexml	1	Developer	27	29/7/2006	29/7/2006 21:48	cautis		
7	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	edaronde		
8	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	etaropa		
9	activexml	1	Developer	27	29/7/2006	29/7/2006 21:48	gabriel		
10	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	hpouyllau2		
11	activexml	1	Project Mana	27	29/7/2006	29/7/2006 21:48	mioana		
12	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	pettelno		
13	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	simonkm		
14	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	vodislav		
15	activexml	0	Developer	27	29/7/2006	29/7/2006 21:48	vrdojak		
16	apollon	1	Undefined	27	29/7/2006	29/7/2006 21:48	contrera		
17	apollon	0	Undefined	27	29/7/2006	29/7/2006 21:48	jmesnil		
18	apollon	0	Undefined	27	29/7/2006	29/7/2006 21:48	rouvoy		
19	asm	1	Project Mana	27	29/7/2006	29/7/2006 21:48	andrei		
20	asm	0	Undefined	27	29/7/2006	29/7/2006 21:48	chassand		
21	asm	1	Project Mana	27	29/7/2006	29/7/2006 21:48	ebruneton		
22	asm	1	Project Mana	27	29/7/2006	29/7/2006 21:48	ekuleshov		
23	asm	0	Undefined	27	29/7/2006	29/7/2006 21:48	mdproctor		

Figura 7.4 Anomalia encontrada en uno de los archivos .csv de repositorio ObjectWeb
Fuente: creación propia

	A	B	C	D	E	F	G	H	I
1	proj_unixnaris_admin	1 Project Manager	datasource	29/7/2006 21:48	abitebou				
2	activexml	0 Developer		29/7/2006 21:48	antonella				
3	activexml	1 Project Manager		29/7/2006 21:48	benjello				
4	activexml	0 Developer		29/7/2006 21:48	bogdanmarinoiu				
5	activexml	1 Developer		29/7/2006 21:48	cautis				
6	activexml	0 Developer		29/7/2006 21:48	edaronde				
7	activexml	0 Developer		29/7/2006 21:48	etaropa				
8	activexml	1 Developer		29/7/2006 21:48	gabriel				
9	activexml	0 Developer		29/7/2006 21:48	hpouyllau2				
10	activexml	1 Project Manager		29/7/2006 21:48	mioana				
11	activexml	0 Developer		29/7/2006 21:48	pettelno				
12	activexml	0 Developer		29/7/2006 21:48	simonkm				
13	activexml	0 Developer		29/7/2006 21:48	yodislav				
14	activexml	0 Developer		29/7/2006 21:48	vrdoljak				
15	apollon	1 Undefined		29/7/2006 21:48	contrera				
16	apollon	0 Undefined		29/7/2006 21:48	jmesnil				
17	apollon	0 Undefined		29/7/2006 21:48	rouvoy				
18	asm	1 Project Manager		29/7/2006 21:48	andrei				
19	asm	0 Undefined		29/7/2006 21:48	chassand				
20	asm	1 Project Manager		29/7/2006 21:48	ebruneton				
21	asm	1 Project Manager		29/7/2006 21:48	ekuleshov				
22	asm	0 Undefined		29/7/2006 21:48	mdproctor				
23									

Figura 7.5 Eliminación del campo duplicado en el archivo .csv de repositorio ObjectWeb
Fuente: creación propia

Ejemplo de archivos utilizados para la prueba

A continuación, se muestran algunas tablas del repositorio Alioth cargadas ver Figura 7.6 con el fin de probar el requerimiento funcional Cargar repositorios FLOSSmole hacia el dw_manager, ver Figura 7.7.



Figura 7.6 Archivos .csv de repositorio Alioth
Fuente: creación propia

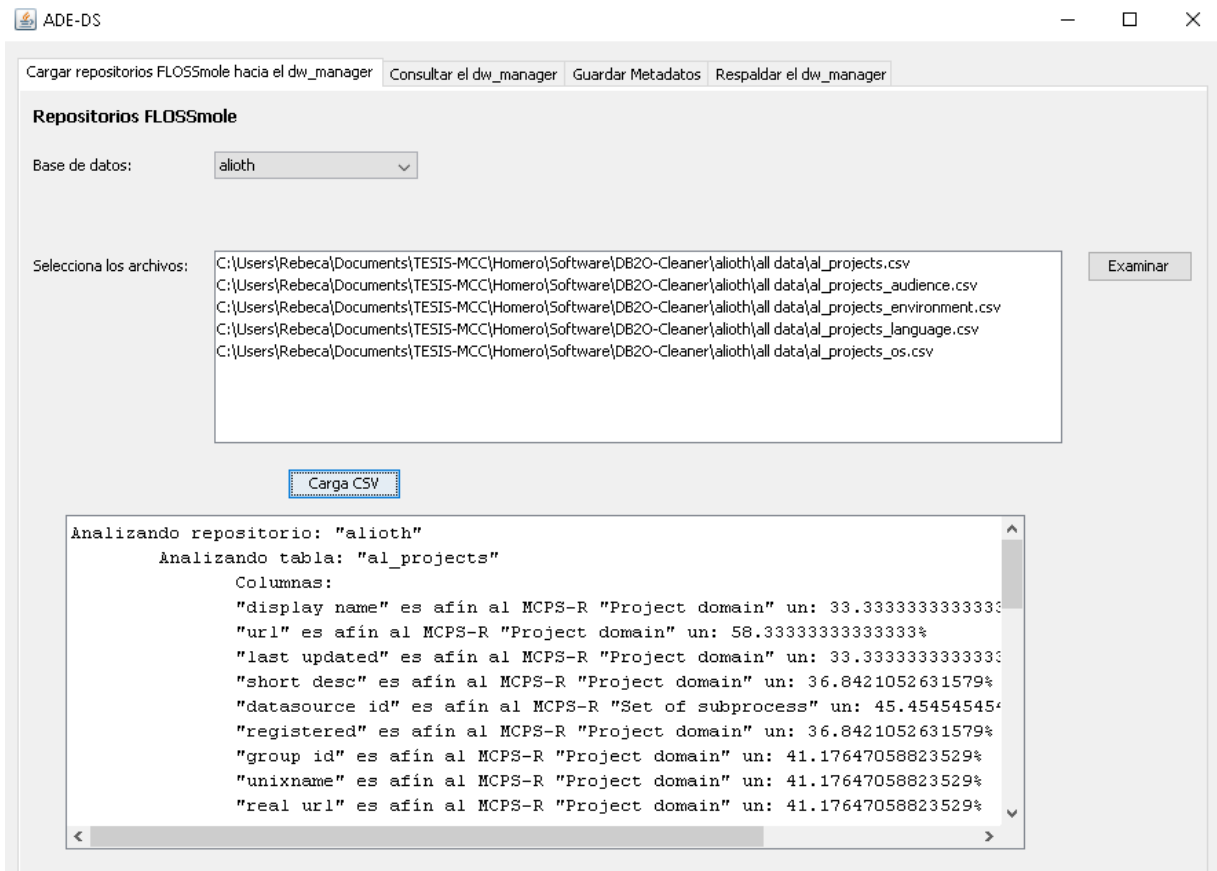


Figura 7.7 Similitudes con el MCPS-R de los archivos .csv del repositorio Alioth
Fuente: creación propia

7.6 Reporte de Pruebas

En esta sección se proporciona una vista general y una vista detallada de los resultados obtenidos al finalizar la ejecución de pruebas.

7.6.1 Resultados generales

A continuación, se presentan los resultados obtenidos de las pruebas, una vez corregidas las anomalías identificadas en cada caso de prueba.

Requerimiento	Caso de Prueba	Resultado de la Prueba
1. Cargar repositorios FLOSSmole hacia el dw_manager	TC-01	Exitoso
2. Consultar el dw_manager	TC-02	Exitoso
3. Guardar metadatos	TC-03	Exitoso
4. Respaldar el dw_manager	TC-04	Exitoso

7.6.2 Resultados detallados

Las pruebas presentaron un resultado exitoso, lo que conlleva al cumplimiento de los requerimientos funcionales en su totalidad. La carga automática de los archivos .csv por el módulo Cargar repositorios FLOSSmole hacia el dw_manager nos permitió generar un análisis de las similitudes de las columnas de los archivos .csv con el MCPS-R. Se hace posible al utilizar el algoritmo de distancia de Levenshtein y al implementar un diccionario de datos que contiene las palabras a comparar del MCPS-R, generando un porcentaje de similitud.

El análisis de similitudes con el MCPS-R puede requerir de una gran cantidad de tiempo para su ejecución, este tiempo depende de varios factores: el tamaño del archivo .csv, las características del hardware y la conexión de red. Este análisis generado en el módulo de Cargar repositorios FLOSSmole hacia el dw_manager se guarda en la base de datos dw_manager. El cual nos permite en el módulo de consultas mostrar la información general de las columnas ordenadas en base al MCPS-R, alfa y factor correspondiente. El módulo de Guardar metadatos, nos permitió visualizar las actividades operacionales utilizando la herramienta ADE-DS y las consultas realizadas hacia el almacén de datos. El módulo de Respaldo, se hace posible utilizando el comando mysqldump con el cual generamos un backup de la base de datos dw_manager. Permitiendo obtener un script para ejecutar en nuestro manejador de base de datos MySQL.

En este capítulo se incluyó el plan de pruebas y los resultados obtenidos. En el siguiente capítulo se describe las conclusiones obtenidas y trabajos futuros.

Capítulo 8

Conclusiones y Trabajos

futuros

En este capítulo se presentan las conclusiones a las que se llegaron y las aportaciones obtenidas como resultado de esta investigación, además, se sugieren algunos trabajos futuros para dar continuidad a la investigación.

A continuación, se presenta la estructura del capítulo:

8.1 Conclusiones

8.2 Aportaciones

8.3 Trabajos Futuros

8.1 Conclusiones

Los diseños conceptual, lógico y físico del almacén de datos `dw_manager`, permite la integración de fuentes heterogéneas de datos de proyectos, su estructuración y consulta, como parte del sistema ADE-DS, de soporte a decisiones, para el desarrollo de software.

El sistema ADE-DS tiene como objetivo: facilitar la carga e integración de datos de proyectos de desarrollo de software provenientes de los repositorios FLOSSmole, en formato valores separados por coma (CSV). La integración de los datos se realiza mediante el uso del algoritmo Levenshtein, para comparar los nombres de las características del MCPS-R con los nombres de las columnas de cada tabla de los repositorios FLOSSmole.

El sistema permite seleccionar las consultas administrativas utilizando las clasificaciones de los atributos del desarrollo de software, mediante las alfas de Essence, los “Factores de desarrollo de software” o por la estructura original proveniente del repositorio FLOSSmole específico.

El sistema ADE-DS alcanzó correctamente los objetivos proyectados, de acuerdo a los resultados de las pruebas realizadas, con distintos repositorios FLOSSmole.

8.2 Aportaciones

Las aportaciones obtenidas como resultado de esta investigación son:

- Creación de estructura de base de datos, del almacén de datos `dw_manager`, para los datos de proyectos de software.
- Herramienta ADE-DS, que permite:
 - La carga e integración de datos de proyectos de desarrollo de software, en formato .CSV, provenientes de repositorios de FLOSSmole
 - Consultar el almacén `dw_manager`
 - Guardar metadatos y

- Respaldar el dw_manager

8.3 Trabajos Futuros

Agregar interacción entre el administrador o usuario del sistema, durante la selección de repositorios del FLOSSmole y sus campos, para que pueda especificar qué campos fuente cargar y los campos destino (en el almacén) donde se cargarán.

Mejorar el módulo de consultar el dw_manager ya que en este trabajo se realizaron solamente consultas generales de las tablas de los repositorios FLOSSmole, útiles para el administrador del almacén principalmente. Se requiere, entonces, ampliar la funcionalidad de las consultas para realizar consultas específicas reconfigurables, con la ayuda de índices y vistas en la base de datos.

Referencias

- Cancino Fuentes, A. (2020). Priorización de características del desarrollo de software de acuerdo a la experiencia de los equipos de desarrollo de software orientado a la mejora de consultas a un almacén de datos históricos. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México.
- Delgado Solis, C. (2008). Caracterización de proyectos de software para configurar su desarrollo y habilitar la comparación entre casos almacenados en la memoria organizacional. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México.
- Sanchez Santamaria, M. (2010) Evaluación de técnicas de comparación de diferentes grupos de características de proyectos de software. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México.
- Perez Luna, E. (2016) Componentes de acoplamiento para la infraestructura de soporte a minería de datos de desarrollo de software. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México.
- Aguilar Viveros, H. (2016) Preparación de repositorios de software usando ontologías. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México.
- Elamin, e., & feki, j. (2014). Toward an ontology based approach for data warehousing state of the art and proposal. *International Journal In Foundations Of Computer Science & Technology*, 170–179.
- Abdalaziz Ahmedl, R., & Mohamed Ahmed, T. (2014). Generating Data Warehouse Schema. *International Journal In Foundations Of Computer Science & Technology*, 4(1), 1-16. doi: 10.5121/ijfcst.2014.4101
- Hanlin, Q., Xianzhen, J., & Xianrong, Z. (2012). Research on Extract, Transform and Load (ETL) in Land and Resources Star Schema Data Warehouse. 2012 Fifth International Symposium on Computational Intelligence and Design. doi: 10.1109/iscid.2012.38
- Homayouni, H., (2018). Testing Extract-Transform-Load Process in Data Warehouse Systems. 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). doi: 10.1109/ISSREW.2018.000-6.
- Warners, H. and Randriatoamanana, R., (2016). Datawarehouse: A data warehouse artist who have ability to understand data warehouse schema pictures. 2016 IEEE Region 10 Conference (TENCON). doi: 10.1109/TENCON.2016.7848419.

- Rodzi, N., Othman, M. and Yusuf, L., (2015). Significance of data integration and ETL in business intelligence framework for higher education. 2015 International Conference on Science in Information Technology (ICSITech). doi: 10.1109/ICSITech.2015.7407800.
- Sabtu, A., Azmi, N., Sjarif, N., Ismail, S., Yusop, O., Sarkan, H., & Chuprat, S. (2017). The challenges of Extract, Transform and Loading (ETL) system implementation for near real-time environment. 2017 International Conference On Research And Innovation In Information Systems (ICRIIS). doi: 10.1109/icriis.2017.8002467
- Kimball, R. & Caserta, J. (2004) *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*, Wiley.
- Connolly, t., & begg, c. (2005). *Database systems a practical approach to design, implementation, and management* (4th ed., p. 1157).
- El Idrissi Esserhrouchni, O., Frikh, B. & Ouhbi, B. (2014). Building Ontologies: a State of the Art, and an Application to Finance Domain. 2014 Fifth International Conference on Next Generation Networks and Services (NGNS), Casablanca.
- Calero, C., Piattini, M., & Ruiz, F. (2006). *Ontologies for Software Engineering and Software Technology*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.
- González, M. G., Solís, C. D., Santamaria, M. S., Daniel, O., Bonilla, F., Salgado, S., & Salazar, A. M. (2017). Adecuación del Desarrollo de Software Resumen (No. 2). Cuernavaca, Morelos, México.
- Rashiti, A., & Damoni, A. (2017). Adaption of Levenshtein Algorithm for Albanian Language. 2017 International Conference On Computational Science And Computational Intelligence (CSCI). doi: 10.1109/csci.2017.23
- OMG. (2015). *Kernel and Language for Software Engineering Methods (Essence) Version 1.1*. Needham, MA.
- Centelles, M. (2005). Taxonomías para la categorización y la organización de la información en sitios web. Consultado 15 May 2018, de <http://www.upf.edu/hipertextnet/numero-3/taxonomias.html>
- Abed Gregio, A., Barbato, L., Duarte, L., Montes, A., Hoepers C., Stedding-Jessen, K. (2007). *Taxonomías de Vulnerabilidades: Situación Actual*. Consultado 15 May 2018, de <http://www.lbd.dcc.ufmg.br/colecoes/sbseg/2005/009.pdf>.
- ITU-T Z.151. (2012). *User Requirements Notation (URN)*. Language definition. Telecommunication Standardization Sector of ITU.
- Amyot, D. (2003). Introduction to the User Requirements Notation: learning by example. *Computer Networks*, 42(3), 285-301. doi: 10.1016/s1389-1286(03)00244-5

- Fernández, M. L., Gómez, A. P., & Juristo, N. (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI. Stanford University, California, 33–40. Consultado de <http://oa.upm.es/5484/Methontology.pdf>.
- Fernández, M. L., Gómez, A. P., & Sierra, J. P. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems & their applications, 37–46. Consultado de [http://12f.inesc-id.pt/~joana/prc/artigos/06b Building a Chemical Ontology Using Methontology and the Ontology Design Environment - Fernandez Lopez.pdf](http://12f.inesc-id.pt/~joana/prc/artigos/06b%20Building%20a%20Chemical%20Ontology%20Using%20Methontology%20and%20the%20Ontology%20Design%20Environment%20-%20Fernandez%20Lopez.pdf)
- IEEE. (1999). IEEE Standard for Developing Software Life Cycle Processes. New York: IEEE Computer Society. doi: 10.1109/IEEESTD.1990.101064.
- Poljak, R., Posic, P., & Jaksic, D. (2017). Comparative analysis of the selected relational database management systems. 2017 40Th International Convention On Information And Communication Technology, Electronics And Microelectronics (MIPRO). doi: 10.23919/mipro.2017.7973658
- Hassan, A. E. (2008). The Road Ahead for Mining Software Repositories. Consultado de [https://thelackthereof.org/projects/school/uiuc/cs512 - Data Mining Principles and Algorithms - Spring 2010/library/The road ahead for mining software repositories.pdf](https://thelackthereof.org/projects/school/uiuc/cs512%20-%20Data%20Mining%20Principles%20and%20Algorithms%20-%20Spring%202010/library/The%20road%20ahead%20for%20mining%20software%20repositories.pdf)
- G. Robles, U. Re y, and J. Carlos (2010). Replicating MSR : A study of the potential replicability of papers published in the Int. Work. Conf. Min. Softw. Repos. pp. 171– 180.
- Gonzalez-Barahona, J. M., Izquierdo-Cortazar, D., & Squire, M. (2010). Repositories with public data about software development. International Journal of Open Source Software and Processes (IJOSSP). Consultado de <http://www.igi-global.com/article/international-journal-open-source-software/44968>
- de F. Farias, M., Novais, R., Júnior, M., da Silva Carvalho, L., Mendonça, M., & Spínola, R. (2016). A systematic mapping study on mining software repositories. Proceedings Of The 31St Annual ACM Symposium On Applied Computing - SAC '16. doi: 10.1145/2851613.2851786
- Curto Díaz, J., & Conesa Caralt, J. (2012). Introducción al business intelligence. Barcelona: Editorial UOC.
- Jacobson, I., Pan-Wei, N., McMahan, P. E., Spence, I., Lidman, S., & Zapata-Jaramillo, C. M. (2013). La Esencia de la Ingeniería de Software: El Núcleo de Semat. Revista Latinoamericana de Ingeniería de Software, 1(3), 71–78. doi:10.18294/relais.2013.71-78

- Ribeiro, M., Duarte, V., Salgado, E., & Castro, C. (2017). Prioritization of Critical Success Factors In The Process of Software Development. *IEEE Latin America Transactions*, 15(1), 137-144. doi: 10.1109/tla.2017.7827917
- Rockart, J. F., & Sloan, A. P. (1978). A New Approach To Defining The Chief Executive's Information Needs. Consultado de <https://pdfs.semanticscholar.org/1b3a/7bf37b2728f137960bd21762e0dc98c6bfb1.pdf>
- Tawanda B., & Mnkandla, E. (2017). Modelling the critical success factors of agile software development projects in South Africa. *SA Journal of Information Management*, 19(1), 8. Consultado de <https://sajim.co.za/index.php/sajim/article/view/838/1168>
- Taromirad, M. & Ramsin, R. (2008). Cefam: Comprehensive evaluation framework for agile methodologies, paper presented at the Software Engineering Workshop, 2008. SEW'08. 32nd Annual IEEE, Kassandra, Greece, October 15–16, 2008.
- Yaghoobi, T. (2017). Prioritizing key success factors of software projects using fuzzy AHP. *Journal Of Software: Evolution And Process*, 30(1), e1891. doi: 10.1002/smr.1891

Anexo A Atributos por factor del MCPS-R

A continuación, se muestran los cinco factores que constituyen el Marco de Características de Proyectos de Software (MCPS-R). En la Tabla A.1 se muestran los atributos correspondientes al factor tipo de proyecto de software; en la Tabla A.2 se muestran los atributos correspondientes al factor medibles; en la Tabla A.3 se muestran los atributos correspondientes al factor sociológico; en la Tabla A.4 se muestran los atributos correspondientes al factor tecnológico; y en la Tabla A.5 se muestran los atributos correspondientes al factor configurables.

Tabla A.1. Factor tipo de proyecto de software
Fuente: (González et al., 2017).

Nombre del Atributo	Identificador del atributo A {X1...Xn}	Criterio para la asignación del valor
Clasificación del software	X ₁	Ubicar el proyecto en alguno de los tipos generales: software de sistema, software para comercio, sistemas de información, software militar, software de tiempo real, aplicación Web u otro.
Contexto o dominio del proyecto	X ₂	Indicar el entorno profesional donde se utilizará el sistema: administrativo, bancario, contabilidad, educación, ingeniería, finanzas, medicina, ciencia, ventas u otro.
Naturaleza del proyecto	X ₃	Identificar la naturaleza del proyecto acuerdo a sus antecedentes: tiene antecedentes, desarrollo de nueva aplicación, mejora de aplicación, mantenimiento de aplicación o reingeniería.
Riesgos del proyecto	X ₄	Identificar el nivel de riesgo de tecnología para el proyecto: muy bajo, bajo, moderado, alto o muy alto.
	X ₅	Identificar el impacto de los riesgos de tecnología, en caso de suceder: ninguno, insignificante, tolerable, serio o catastrófico.
	X ₆	Identificar el nivel de riesgo de personas para el proyecto: muy bajo, bajo, moderado, alto o muy alto.
	X ₇	Identificar el impacto de los riesgos de personas, en caso de suceder: ninguno, insignificante, tolerable, serio o catastrófico.
	X ₈	Identificar el nivel de riesgo organizacional para el proyecto: muy bajo, bajo, moderado, alto o muy alto.
	X ₉	Identificar el impacto de los riesgos de organizacionales, en caso de suceder ninguno, insignificante, tolerable, serio o catastrófico.
	X ₁₀	Identificar el nivel de riesgo de herramienta para el proyecto: muy bajo, bajo, moderado, alto o muy alto.

	X ₁₁	Identificar el impacto de los riesgos de herramientas, en caso de suceder: ninguno, insignificante, tolerable, serio o catastrófico.
	X ₁₂	Identificar el nivel de riesgo de requerimiento para el proyecto: muy bajo, bajo, moderado, alto o muy alto.
	X ₁₃	Identificar el impacto de los riesgos de requerimientos, en caso de suceder: ninguno, insignificante, tolerable, serio o catastrófico.
	X ₁₄	Identificar el nivel de riesgo de estimación para el proyecto: muy bajo, bajo, moderado, alto o muy alto.
	X ₁₅	Identificar el impacto de los riesgos de estimación, en caso de suceder: ninguno, insignificante, tolerable, serio o catastrófico.

Tabla A.2. Factor atributos medibles

Fuente: (González et al., 2017).

Nombre del Atributo	Identificador del atributo A {X ₁ ...X _n }	Criterio para la asignación del valor
Tamaño del proyecto	X ₁₆	Cantidad real del total de líneas de código (LOC-R) (sin contar comentarios y líneas en blanco) del sistema.
	X ₁₇	Cantidad estimada del total de líneas de código (LOC-E) (sin contar comentarios y líneas en blanco) del sistema.
	X _(18, 20, 22, 24, 26, 28, 30)	Cantidad real de líneas de código del producto final de tipo: base, borradas, reutilizadas, añadidas, modificadas, nuevas y cambiadas, y nuevas para reúso.
	X _(19, 21, 23, 25, 27, 29, 31)	Cantidad estimada de líneas de código del producto final de tipo: base, borradas, reutilizadas, añadidas, modificadas, nuevas y cambiadas, y nuevas para reúso.
	X ₃₂	Cantidad de puntos de función reales del producto final.
	X ₃₃	Cantidad de puntos de función estimadas del producto final.
Defectos en el proyecto	X ₃₄	Total de defectos insertados en el producto final.
	X ₃₅	Total de defectos removidos en el producto antes de la entrega.
Costo del proyecto	X ₃₆	Costo estimado del proyecto (expresado en pesos).
	X ₃₇	Costo real del proyecto (expresado en pesos).
Complejidad del proyecto	X ₃₈	Complejidad de programación: bajo, medio o alto.
	X ₃₉	Complejidad en los datos: bajo, medio o alto.
	X ₄₀	Complejidad de organización: una persona, un grupo de un departamento, varios grupos de un departamento, varios grupos de varios departamentos, varios grupos de varios sitios no co-localizados o desarrollo de subcontratos por terceros.

Tiempo de desarrollo del proyecto	X ₄₁	Tiempo de duración estimado para el desarrollo del proyecto. Expresado en minutos.
	X ₄₂	Tiempo de duración real en el desarrollo del proyecto. Expresado en minutos.
	X ₄₃	Tiempo de desarrollo por producto de trabajo. Expresado en minutos.
Esfuerzo en el proyecto	X ₄₄	Esfuerzo estimado del desarrollo del proyecto. Expresado en Minutos/Hombre.
	X ₄₅	Esfuerzo real del desarrollo del proyecto. Expresado en Minutos/Hombre.

Tabla A.3. Factor atributos sociológicos
Fuente: (González et al., 2017).

Nombre del Atributo	Identificador del atributo A {X₁...X_n}	Criterio para la asignación del valor
Número de desarrolladores	X ₄₆	Número de participantes de tiempo completo.
Nivel máximo de educación	X ₄₇	Identificar el nivel de educación de cada desarrollador: técnico, ingeniería o licenciatura, maestría, doctorado u otro.
Experiencia del equipo en el desarrollo del proyecto	X ₄₈	Experiencia en años en el desarrollo de software.
	X ₄₉	Forma de adquirir la experiencia por desarrollador. Ubicarla en alguna de las alternativas: en institución educativa, en la industria o de manera independiente.
Experiencia del equipo en el lenguaje de programación	X ₅₀	Experiencia en años en implementaciones en el lenguaje de programación a medir.
	X ₅₁	Forma de adquirir la experiencia por desarrollador. Ubicarla en alguna de las alternativas: en institución educativa, en la industria o de manera independiente.
Experiencia del equipo en la metodología de desarrollo	X ₅₂	Experiencia en años en la aplicación de la metodología a medir.
	X ₅₃	Forma de adquirir la experiencia por desarrollador. Ubicarla en alguna de las alternativas: en institución educativa, en la industria o de manera independiente.
Experiencia del equipo en el dominio del proyecto	X ₅₄	Experiencia en años en el desarrollo de software.
	X ₅₅	Forma de adquirir la experiencia por desarrollador. Ubicarla en alguna de las alternativas: en institución educativa, en la industria o de manera independiente.
Experiencia del equipo en el uso de las herramientas de soporte	X ₅₆	Ubicar la experiencia en el uso de las herramientas de soporte: ninguna, baja, media, alta o muy alta.

Tabla A.4. Factor atributos tecnológicos
Fuente: (González et al., 2017).

Nombre del Atributo	Identificador del atributo A {X1...Xn}	Criterio para la asignación del valor
Dominio de aplicación	X ₅₇	Nombre del dominio tecnológico en donde se aplica: web, medico, administrativo o de ingeniería.
Paradigma del lenguaje principal	X ₅₈	Paradigma del lenguaje principal: procedural, orientado a objetos, funcional, lógica, dirigida por eventos, imperativa, orientada a arreglo, orientada a pila, orientada a aspectos, flujo de datos, visual o ensamble.
Lenguajes de programación	X ₅₉	Nombres de lenguajes de programación usados: Java, C++, C, C#, Cliper, Cobol, JavaScript, Visual Basic, Pascal, Delphi u otro.
Herramientas de planeación	X ₆₀	Nombres de las herramientas de planificación: Rational Rose, Microsoft Office Visio, Together, Microsoft Office Project u otro.
Herramientas de modelado	X ₆₁	Nombres de las herramientas de modelado: Rational RosE, Enterprise Architect, Altova, Eclipse Modelig framework u otro.
Herramientas de programación	X ₆₂	Nombres de las herramientas de programación: IDE-General, IDE-Programación, NetBeans, Eclipse u otro.
Manejador de base de datos	X ₆₃	Nombres de los manejadores: Acces, Informix, MySQL, Oracle, PostgresQL, SQL Server, Sybase u otro.
Sistema operativo	X ₆₄	Nombres de los sistemas operativos en el que correrá la aplicación: Apple-Macintosh, Linux, Microsoft Windows, Unix u otro.
Tipo de máquina	X ₆₅	Nombres de los tipos de máquina en los que se utilizará: mainframe, mini computadora, pc, dispositivo móvil u otro.

Tabla A.5. Factor atributos configurables
Fuente: (González et al., 2017).

Nombre del Atributo	Identificador del atributo A {X1...Xn}	Criterio para la asignación del valor
Conjunto de productos de trabajo a obtener	X ₆₆	Documentación de Planificación.
		Documentación del Modelo de negocio.
		Documentación de Identificación de componentes.
		Documentación de Prototipo de sistema.
		Documentación de Requerimientos de sistema.

		Documentación de Revisión de requerimientos y prototipo del sistema.
		Documentación de Ingeniería inversa.
		Documentación de Ingeniería directa.
		Documentación de Prototipo de software.
		Documentación de Requerimientos de software.
		Documentación de Especificación de pruebas de aceptación.
		Documentación de Revisión de requerimientos de software y especificación de pruebas de aceptación.
		Documentación de Definición de plataforma y restricciones a los requerimientos.
		Documentación de Arquitectura de software (HLD).
		Documentación de Especificación de pruebas de integración.
		Documentación de Revisión de arquitectura de software.
		Documentación de Estructura de división del trabajo.
		Documentación de Diseño detallado de unidades de trabajo.
		Documentación de Especificación de pruebas unitarias.
		Documentación de Revisión de diseño detallado y especificación de pruebas unitarias.
		Documentación de Código de unidad.
		Documentación de Revisión de código de unidad.
		Documentación de Prueba de unidad.
		Documentación de Prueba de integración.
		Documentación de Prueba de aceptación.
		Documentación de Reporte post mórtem del proceso de desarrollo.
Estructura general del producto de software	X ₆₇	Nombre los tipos de ensamble-parte incluidos en el patrón de estructura: sistema, subsistema, producto, componente, módulo u objeto
Conjunto de subprocesos	X ₆₈	Nombre de los subprocesos incluidos en el patrón de proceso: lanzamiento & estrategia, planificación, especificación de requerimientos, análisis & diseño, implementación, prueba o evaluación post mórtem.
Estructura del equipo de trabajo	X ₆₉	Nombre de roles incluidos en el patrón de roles: líder del equipo, administrador de interface con cliente, administrador de diseño, administrador de implementación, administrador de pruebas, administrador de planeación, administrador de proceso, administrador de calidad, administrador de soporte.

		(Existe un solo líder del equipo, y éste puede desempeñar otros roles. Todos los miembros del equipo son desarrolladores y además desempeñan otros roles)
Forma de realizar el desarrollo	X ₇₀	Elección de trabajo colaborativo o individual: individual o colaborativo
Forma de realizar el seguimiento y control del proceso	X ₇₁	Periodicidad: diaria, semanal, quincenal, mensual u otro
	X ₇₂	Horario: preestablecido o variable
	X ₇₃	Duración
Forma de realizar inspecciones	X ₇₄	Tipo de inspección: técnica formal o técnica informal
	X ₇₅	Duración
	X ₇₆	Número promedio de asuntos o defectos identificados.
Modelo de desarrollo	X ₇₇	Principal modelo de proceso de desarrollo: Cascada, Evolutivo, Incremental, Espiral, Win win, ágil u Otro

Anexo B Estructuras de repositorios FLOSSmole a utilizar

A continuación, se especifica la estructura de los repositorios FLOSSmole seleccionados para realizar las pruebas. En la Tabla B.1, se muestran cuantas tablas, atributos y versiones de datos contiene el repositorio Alioth; en la Tabla B.2, se muestran cuantas tablas, atributos y versiones de datos contiene el repositorio RubyForge; y en la Tabla B.3, se muestran cuantas tablas, atributos y versiones de datos contiene el repositorio Tigris.

Tabla B.6. Estructura del Repositorio Alioth.

Fuente: extraído de cuenta phpMyAdmin <http://flossdata.syr.edu/db2/index.php> .

Tabla	Nombre de atributo	Versión de datos
al_projects	<ul style="list-style-type: none"> • datasource_id • display_name • group_id • last_updated • real_url • registered • short_desc • unixname • url 	<ul style="list-style-type: none"> • 287 • 295 • 315 • 325 • 326 • 341 • 349 • 380
al_projects_audience	<ul style="list-style-type: none"> • audience • cat • datasource_id • last_updated • unixname 	<ul style="list-style-type: none"> • 386 • 392 • 1575 • 8084
al_projects_environment	<ul style="list-style-type: none"> • cat • datasource_id • environment • last_updated • subcat • subenvironment • unixname 	<ul style="list-style-type: none"> • 43168
al_projects_language	<ul style="list-style-type: none"> • cat • datasource_id • language • last_updated • unixname 	
al_projects_license	<ul style="list-style-type: none"> • cat • datasource_id • last_updated • license • subcat • sublicense • unixname 	
al_projects_os	<ul style="list-style-type: none"> • cat • datasource_id 	

	<ul style="list-style-type: none"> • last_updated • os • subcat • subcat2 • subos • subos2 • unixname 	
al_projects_status	<ul style="list-style-type: none"> • cat • datasource_id • last_updated • status • unixname 	
al_projects_tags	<ul style="list-style-type: none"> • datasource_id • last_updated • tag • unixname 	
al_projects_topic	<ul style="list-style-type: none"> • cat • datasource_id • last_updated • subcat • subcat2 • subtopic • subtopic2 • topic • unixname 	
al_projects_indexes	<ul style="list-style-type: none"> • datasource_id • indexhtml • last_updated • memberhtml • unixname 	
al_projects_members	<ul style="list-style-type: none"> • datasource_id • fullname • last_updated • unixname • userid • username 	
al_projects_members_roles	<ul style="list-style-type: none"> • datasource_id • last_updated • role • unixname • username 	

Tabla B.7. Estructura del Repositorio RubyForge.

Fuente: extraído de cuenta phpMyAdmin <http://flossdata.syr.edu/db2/index.php> .

Tabla	Nombre de atributo	Versión de datos
rf_developers	<ul style="list-style-type: none"> • dev_loginname • realname • email • date_collected • datasource_id 	<ul style="list-style-type: none"> • 24 • 30 • 31 • 35 • 39
rf_developer_projects	<ul style="list-style-type: none"> • dev_loginname • proj_unixname • is_admin • position • date_collected • datasource_id 	<ul style="list-style-type: none"> • 43 • 48 • 59 • 64 • 70 • 76
rf_projects	<ul style="list-style-type: none"> • proj_unixname • url • real_url • date_registered • proj_long_name • proj_id_num • dev_count • activity_percentile • date_collected • datasource_id 	<ul style="list-style-type: none"> • 82 • 88 • 95 • 100 • 105 • 108 • 112 • 121 • 125 • 134
rf_project_description	<ul style="list-style-type: none"> • proj_unixname • description • datasource_id • date_collected 	<ul style="list-style-type: none"> • 140 • 145 • 157 • 166
rf_project_environment	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description • date_collected 	<ul style="list-style-type: none"> • 169 • 174 • 179 • 184 • 189
rf_project_indexes	<ul style="list-style-type: none"> • proj_unixname • indexhtml • date_collected • datasource_id 	<ul style="list-style-type: none"> • 201 • 219 • 229 • 238
rf_project_intended_audience	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description • date_collected 	<ul style="list-style-type: none"> • 248 • 256 • 266 • 279 • 289
rf_project_licenses	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description 	<ul style="list-style-type: none"> • 298 • 307 • 317 • 332
rf_project_natural_language	<ul style="list-style-type: none"> • proj_unixname 	<ul style="list-style-type: none"> • 336

	<ul style="list-style-type: none"> • datasource_id • code • description • date_collected 	<ul style="list-style-type: none"> • 344 • 375 • 382 • 388
rf_project_operating_system	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description • date_collected 	<ul style="list-style-type: none"> • 1573 • 8080 • 12987
rf_project_programming_language	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description • date_collected 	
rf_project_status	<ul style="list-style-type: none"> • proj_unixname • code • description • code_on_page • date_collected • datasource_id 	
rf_project_topic	<ul style="list-style-type: none"> • proj_unixname • datasource_id • code • description • date_collected 	

Tabla B.8. Estructura del Repositorio Tigris

Fuente: extraído de cuenta phpMyAdmin <http://flossdata.syr.edu/db2/index.php>

Tabla	Nombre de atributo	Versión de datos
tig_messages	<ul style="list-style-type: none"> • creator • datasource_id • description • forumid • guid • link • postDate • postDateStr • project • pubDate • title 	<ul style="list-style-type: none"> • 284 • 293 • 301 • 310 • 322 • 334 • 340 • 348 • 379 • 385 • 391
tig_people	<ul style="list-style-type: none"> • datasource_id • full_name • last_updated • username 	<ul style="list-style-type: none"> • 1577 • 8083 • 43167

tig_projects	<ul style="list-style-type: none"> • body • datasource_id • description • last_updated • license • unixname 	
tig_project_categories	<ul style="list-style-type: none"> • category • datasource_id • last_updated • project 	
tig_project_developer_indexes	<ul style="list-style-type: none"> • datasource_id • html • last_updated • project 	
tig_project_developer_roles	<ul style="list-style-type: none"> • datasource_id • last_updated • project • role • username 	
tig_project_discussions	<ul style="list-style-type: none"> • datasource_id • description • discussion • forumid • last_comment • last_updated • project 	
tig_project_discussion_indexes	<ul style="list-style-type: none"> • datasource_id • html • last_updated • project 	
tig_project_indexes	<ul style="list-style-type: none"> • datasource_id • html • last_updated • unixname 	
tig_project_list_indexes	<ul style="list-style-type: none"> • datasource_id • html • last_updated 	

Anexo C Metodología Methontology

La metodología Methontology (Fernández, Gómez, & Juristo, 1997; Fernández, Gómez, & Sierra, 1999) se desarrolló por el grupo de ingeniería ontológica de la Universidad Politécnica de Madrid. Esta metodología permite construir ontologías en el nivel de conocimientos, y tiene sus raíces en las actividades identificadas por el proceso de desarrollo de software propuesto por (IEEE, 1999) y en otras metodologías de ingeniería de conocimientos (Gómez, Juristo, Montes, & Pazos, 1997).

La Methontology proporciona guías sobre cómo lleva a cabo el desarrollo de la ontología a través de las actividades de especificación, conceptualización, formalización, implementación y mantenimiento. A continuación, se describe brevemente en que consiste cada una de estas actividades:

- La actividad de especificación permite determinar por qué se construye la ontología, cuál será su uso y quienes serán sus usuarios finales.
- La actividad de conceptualización se encarga de organizar y convertir una percepción informal del dominio en una especificación semi-formal, para lo cual utiliza un conjunto de representaciones intermedias, basadas en notaciones tabulares y gráficas, que pueden comprenderse por los expertos de dominio y los desarrolladores de ontologías.
- La actividad de formalización se encarga de la transformación de dicho modelo conceptual en un modelo formal o semi-computable.
- La actividad de implementación construye modelos computables en un lenguaje de ontologías (Ontolingua, RDF Schema, OWL, etc.) La mayor parte de las herramientas de ontologías permiten llevar a cabo esta actividad de manera automática. Por ejemplo, Protégé es un editor para construir ontologías y un marco general para representar el conocimiento.
- La actividad de mantenimiento se encarga de la actualización y/o corrección de la ontología, en caso necesario.

Principales componentes de modelado de ontologías

Methontology propone conceptualizar una ontología usando un conjunto de representaciones intermedias tabulares y gráficas, las cuales permiten modelar los principales componentes. Estos componentes son:

- **Conceptos:** Son objetos o entidades, considerados desde un punto de vista amplio. Ejemplo, en el dominio legal son conceptos: persona física, tribunal, menor de edad, etc. Los conceptos de una ontología están normalmente organizados en taxonomías en las cuales se pueden aplicar mecanismos de herencia.

- **Relaciones:** Representan un tipo de asociación entre conceptos del dominio, por ejemplo, como se muestra en la Figura C.1.

Concepto	Relación Binaria	Concepto
Juicio	Se celebra en	Juzgado

Figura C.1 Ejemplo de relaciones en un dominio

- **Instancias:** Representan individuos en la ontología, por ejemplo, una instancia del concepto tribunal, que se muestra en la Figura C.2.

Concepto	Instancias
Tribunal	<ul style="list-style-type: none"> - Audiencia Provincial de Albacete - Tribunal Constitucional

Figura C.2 Ejemplo de instancias

- **Constantes:** Valores numéricos que no cambian en un largo periodo de tiempo, por ejemplo: En México la mayoría de edad es 18 años.
- **Atributos:** Describen propiedades. Se pueden distinguir dos tipos de atributos.
 - a) **Atributos de instancia:** Describen propiedades de las instancias de los conceptos, en las cuales toman su(s) valor(es). Estos atributos se definen en un concepto y se heredan a sus subconceptos e instancias. Un ejemplo se muestra en la Figura C.3.

Concepto	Atributos de instancia
Persona natural	Número de cédula

Figura C.3 Ejemplo de atributos de instancia

- b) **Atributos de clase:** Describen conceptos y toman su(s) valor(es) del concepto en el cual se definen. Estos atributos no se heredan ni a los subconceptos ni a las instancias. Un ejemplo se muestra en la Figura C.4.

Concepto	Atributos de clase
Persona	<ul style="list-style-type: none"> • Tipo de persona: Persona natural / Persona jurídica

Figura C.4 Ejemplo de atributos de clase

- **Axiomas formales:** Expresiones lógicas que son siempre verdaderas y son utilizadas normalmente para especificar restricciones en la ontología. Por ejemplo: Una persona no puede ser el demandado y el demandante en el mismo juicio.
- **Reglas:** Se usan generalmente para inferir conocimiento en la ontología, tales como valores de atributos, instancias de relaciones, etc. Por ejemplo: Un juicio donde el acusado es un menor de edad que tiene 14 años se celebra en un tribunal de menores.

Para realizar el modelado conceptual de una ontología es necesario llevar a cabo ciertas tareas de conceptualización como las mostradas en la Figura C.5.

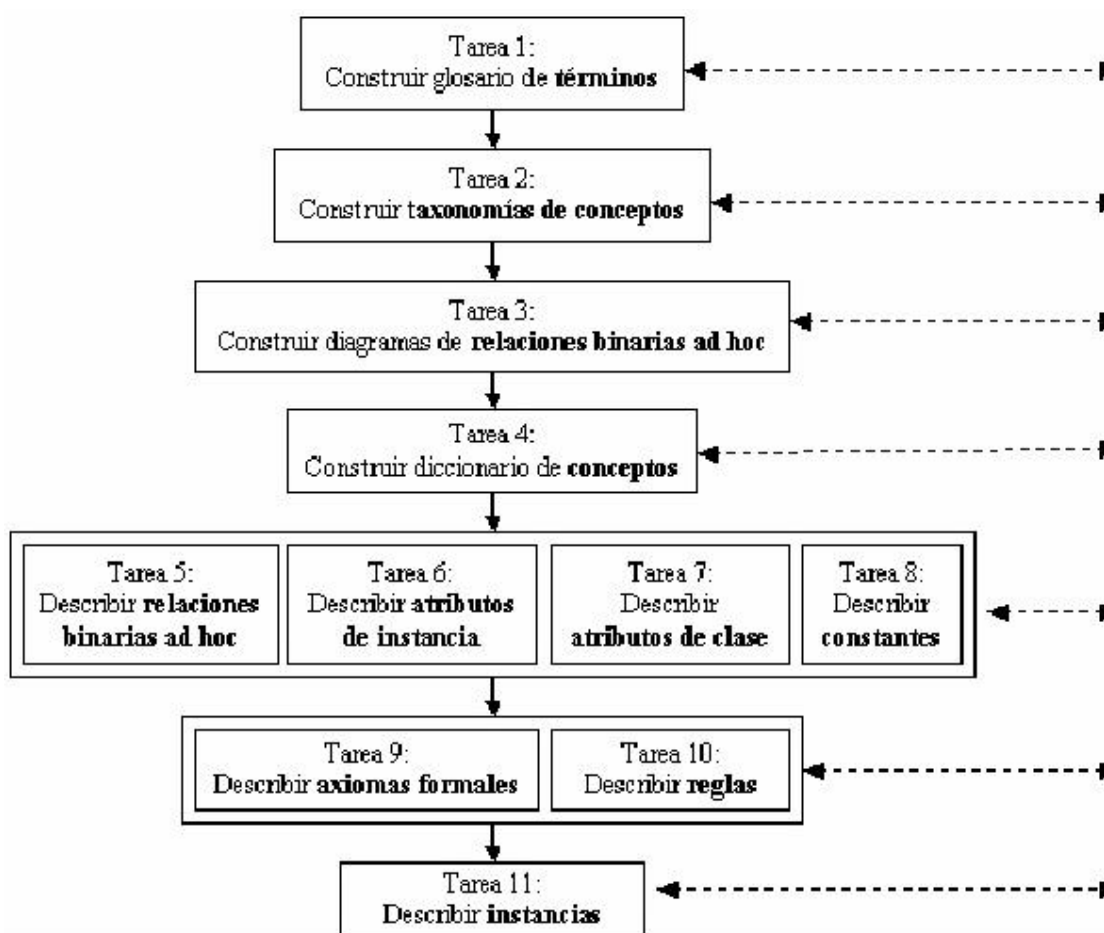


Figura C.5 Modelado conceptual de una ontología

Las tareas de conceptualización en Methontology, mostradas en la Figura C.5, se describen a continuación:

Tarea 1: Construir el glosario de terminos. El glosario de terminos debe incluir todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, sus sinonimos y acronimos. Es importante mencionar, que al empezar pueden existir varios terminos que se refieren al mismo concepto los cuales deben ser identificados y colocados como sinonimos.

Tarea 2: Construir la taxonomia de conceptos. Cuando el glosario de terminos tenga una cantidad importante de elementos, se construye una taxonomia que defina la jerarquia entre conceptos.

Tarea 3: Construir un diagrama de relaciones binarias. El objetivo de este diagrama es establecer las relaciones entre los conceptos de una o mas taxonomias de conceptos. Se debe evaluar que el diagrama creado no contenga errores.

Tarea 4: Construir el diccionario de conceptos. El diccionario de conceptos contiene los conceptos del dominio, sus relaciones, instancias, atributos de clases y atributos de instancias. Las relaciones, atributos de instancias, y atributos de clases son locales al concepto, lo que significa que sus nombres pueden repetirse en diferentes conceptos.

Tarea 5: Describir las relaciones binarias en detalle. Se crea la tabla de relaciones binarias en la que se describe detalladamente todas las relaciones binarias incluidas en el diccionario de conceptos. Para cada relacion binaria se debe especificar: nombre, conceptos fuente y destino, cardinalidad y relacion inversa.

Tarea 6: Describir los atributos de instancia en detalle. Se crea la tabla de atributos de instancias en la que se describe detalladamente todos los atributos de instancias incluidos en el diccionario de conceptos. Los atributos de instancias son aquellos atributos que describen las instancias de un concepto, y sus valores pueden ser diferentes para cada instancia del concepto. Para cada atributo de instancia, se debe especificar: nombre, concepto al que pertenece, tipo de valor, rango de valores (en el caso de valores numericos) y cardinalidad.

Tarea 7: Describir los atributos de clases en detalle. Se crea la tabla de atributos de clases en la que se describe detalladamente todos los atributos de clases incluidos en el diccionario de conceptos. Para cada atributo de clase, se debe especificar: nombre, concepto donde es definido, tipo de valor, valor y cardinalidad.

Tarea 8: Describir las constantes en detalle. Se crea la tabla de constantes en la que se describe detalladamente cada una de las constantes definidas en el glosario de terminos. Para cada constante, se debe especificar: nombre, tipo de valor, valor y unidad de medida (para constantes

numericas).

Tarea 9: Describir los axiomas formales. Se deben identificar los axiomas formales necesarios en la ontologia y describirlos con precision en una tabla. Para cada definicion de axioma formal de debe especificar: nombre, descripcion, expresion logica que formalmente lo describe (preferiblemente utilizando logica de primer grado), los conceptos, atributos y relaciones binarias a las cuales el axioma hace referencia y las variables utilizadas.

Tarea 10: Definir las reglas. Se deben identificar cuales reglas son necesarias en la Ontologia y describirlas en una tabla de reglas. Para cada regla, se debe especificar: nombre, descripcion, expresion que formalmente la describe, los conceptos, los atributos y las relaciones a los que hace referencia y las variables usadas en la expresion. Para la especificacion de las reglas se sugiere la forma: Si <condiciones> entonces <consecuencias o acciones>.

Tarea 11: Describir las instancias. Una vez que el modelo conceptual de la Ontologia ha sido creado, se deben definir las instancias relevantes que aparecen en el diccionario de conceptos en una tabla de instancias. Para cada instancia se debe especificar: nombre, concepto al que pertenece y valores de los atributos.

Anexo D Diseño de base de datos

A continuación, se muestra la comparación realizada sobre las ventajas y desventajas de las bases de datos relacionales y bases de datos basadas en ontologías como se muestra en la Tabla D.1.

Tabla D.1. Comparación de BD relacionales VS OBDB

Fuente: creación propia.

Artículo	BD relacionales	OBDB	Ventajas	Desventajas
(Lependu & Dou, 2011)		✓	<ul style="list-style-type: none"> • Susceptible de procesamiento y razonamiento automatizados. • Útiles manejar consultas basadas en ontologías. 	<ul style="list-style-type: none"> • No tiene integridad de datos incorporada. • Los datos se desnormalizan y requieren actualizaciones masivas.
(Sir, Bradac and Fiedler, 2015)	✓		<ul style="list-style-type: none"> • Utilizan el sistema CWA y UNA de representación del conocimiento • Se concentran en el almacenamiento claro de datos. • Diseño de metodo es desde 0. • Son compatibles con ACID, lo que significa que satisfacen los requisitos de atomicidad, consistencia, aislamiento y durabilidad. • Garantiza evitar la duplicidad de registros y la integridad referencial. • Sintaxis es el modelo entidad-relación 	<ul style="list-style-type: none"> • Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica. • Los RDBMS no funcionan bien, o en absoluto, con datos no estructurados o semiestructurados, debido a restricciones de esquema y tipo.
(Sir, Bradac and Fiedler, 2015)		✓	<ul style="list-style-type: none"> • Utilizan el sistema OWA de representación del conocimiento. • Se enfocan en agregar significado y comprensión. • Diseño de metodos es en base a ontologias existentes. • Sintaxis lenguajes OWL,RDF. 	<ul style="list-style-type: none"> • No utiliza el sistema CWA y UNA para representar el conocimiento. • Se enfoca en la integración de datos semánticos para la comprensión de la estructura de la información.

En base a la tabla anterior se nota que las ontologías proporcionan un medio para especificar formalmente descripciones y relaciones complejas sobre la información de una manera que es expresiva utilizando un vocabulario común. Por lo tanto, se utilizaron ontologías solamente para definir el modelo conceptual, y se generó la ontología MCPS-R. Así mismo, se decidió utilizar

el diseño de base de datos relacional, y se eligió utilizar como sistema gestor de base de datos MySQL.

Para tomar la decisión de utilizar el gestor de datos MySQL, se tomó en cuenta el análisis comparativo de los sistemas de gestión de bases de datos relacionales (Poljak, Poscie, & Jakšie, 2017), donde se consideraron los criterios de datos básicos, tipos de datos y rendimiento de velocidad. La primera comparación es en base a los datos básicos para cada RDBMS como se muestra en la Tabla D.2.

Tabla D.2. Comparación de datos básicos de los RDBMS
Fuente: traducido de (Poljak, Poscie, & Jakšie, 2017).

	RDBMS		
	MySQL	Oracle 11g	PostgreSQL
Modelo de base de datos	DBMS relacional	DBMS relacional	DBMS relacional
Desarrollador	Oracle	Oracle	PostgreSQL Global Development Group
Licencia	Libre	Comercial	Libre
Sistema operativo soportado	Windows, Linux, OSX, Solaris, FreeBSD	Windows, Linux, OSX, Solaris, HPUX, AIX	Windows, Linux, OSX, Solaris, HPUX, Unix, FreeBSD, NetBSD, OpenBSD
Lenguajes de programación soportados	C, C#, C++, D, Haskell, Perl, PHP, Java, Python, Ruby, Objectiv-C	C, C#, C++, Cobol, Fortran, Haskell, Java, JavaScript, Lisp, Perl, PHP, Python, Ruby, Visual Basic, Objectiv-C	.Net, C, C++, Java, Perl, Python, Tcl
Transacciones	ACID	ACID	ACID

De la Tabla D.2 podemos ver que MySQL y Oracle 11g tienen el mismo desarrollador, Oracle 11g es el único RDBMS comercial, PostgreSQL es compatible con más sistemas operativos, incluso 9. Oracle es compatible con la mayoría de los lenguajes de programación. Todos los RDBMS tienen ACID. La segunda comparación es la diferencia entre tipos de datos que usa un RDBMS en particular que se muestra en la Tabla D.3.

Tabla D.3. Comparación de tipos de datos de los RDBMS
Fuente: traducido de (Poljak, Poscie, & Jakšie, 2017).

Tipo de dato	RDBMS		
	MySQL	Oracle 11g	PostgreSQL
ARRAY	-	+	+
CHAR	+	+	+
CHARACTER	+	-	+
DATE	+	+	+
CHARACTER VARYING	-	-	+
DECIMAL DEC	+	-	+
DOUBLE	+	-	+
ENUM	+	-	+
FLOAT	+	+	-
INT	+	-	-
LONG	-	+	-
NUMBER	-	+	-
NUMERIC	+	-	+
RAW	-	+	-
REAL	+	-	+
TEXT	+	-	+
TIME	+	-	+
TIMESTAMP	+	+	+
UUID	-	-	+
VARCHAR	+	+	+
VARCHAR2	-	+	-
YEAR	+	-	-

La tercera comparación es el rendimiento de velocidad ejecutando ciertos comandos en los RDBMS como se muestra en la Tabla D.4.

Tabla D.4. Comparación de rendimiento de velocidad RDBMS
Fuente: traducido de (Poljak, Poscie, & Jakšie, 2017)

Comandos y consultas	Tiempo de ejecución en ms		
	MySQL	Oracle 11g	PostgreSQL
CREATE TABLE TIRES(ID_TIRES VARCHAR(6) NOT NULL, TIRES_NAME VARCHAR(30) NOT NULL, ID_PRO VARCHAR(2) NOT NULL, ABOVE_CLASS VARCHAR(6), PURPOSE VARCHAR(15) NOT NULL, PRICE NUMBER(6,2) NOT NULL, STOCK NUMBER NOT NULL, SOLD NUMBER NOT NULL);	0.22	0.03	0.49
INSERT INTO TIRES VALUES ('GY3455', 'DURAGRIP', 'GY', NULL, 'LJETNE', 528.66, 12, 1567);	0.05	0.001	0.006
SELECT NAZIV_TIRES, PURPOSE, PRICE FROM TIRES WHERE ID_PRO = 'MN' AND PURPOSE IN (SELECT PURPOSE FROM TIRES WHERE ID_PRO IN (SELECT	0.001	0.01	0.02

ID_PRO FROM MANUFACTURER WHERE PRO_NAME='PIRELLI');			
SELECT PRO_NAME, HEADQUARTERS FROM MANUFACTURER WHERE ID_PRO IN (SELECT ID_PRO FROM TIRES WHERE PURPOSE = 'LJETNE' GROUP BY ID_PRO);	0.001	0.001	0.009
SELECT MANUFACTURER.ID_PRO, MANUFACTURER.PRO_NAM E, TIRES.TIRES_NAMEFROM MANUFACTURER INNER JOIN TIRES ON MANUFACTURER.ID_PRO=TI RES.ID_PRO;	0.001	0.001	0.006
UPDATE TIRES SET NAZIV_TIRES='DURAGRIP1', STOCK=13 WHERE NAZIV_TIRES='DURAGRIP';	0.06	0.001	0.004
DROP TABLE TIRES;	0.16	0.44	0.08

De acuerdo con la tabla anterior se puede ver que la velocidad de ejecución la tiene Oracle, que es mejor usándolo en sistemas grandes y complejos. En segundo lugar, está MySQL destinado a usuarios libres, educación y negocios. Además, MySQL es la base de datos del sistema de fuente abierta más ampliamente utilizada. PostgreSQL también es muy popular para la gestión de bases de datos y se utiliza en situaciones similares a la utilización de MySQL. Solo que PostgreSQL se utiliza en menor número de aplicaciones MySQL.

Anexo E Diccionario de datos MCPSR

A continuación, se presenta en la Figura D.1 como está compuesto el diccionario de datos MCPSR. Cada renglón del diccionario se formar por: 1) la palabra más relevante del factor al que corresponde el atributo del MCPSR (ej. project, factor), 2) el nombre del alfa a la que corresponde (ej. oportunity) y 3) las palabras claves del nombre del atributo del MCPSR (ej. software, classification). Se agregaron, al final del renglón, algunas abreviaciones de dichas palabras (ej. proj, fact, opportun, classif).

```
public String [][] diccionario = {
{"project","factor","oportunity","software","classification","proj","fact","opportun","classif"},
{"factor","stakeholder","project","domain","fact","stakehold","proj"},
{"factor","stakeholder","nature","project","fact","stakehold","natur","proj"},
{"project","factor","oportunity","technological","risk","level","proj","fact","opportun","technolog"},
{"project","factor","oportunity","possible","impact","technological","risk","proj","fact","opportun","possibl","technolog"},
{"project","factor","oportunity","people","risk","level","proj","fact","opportun","peopl"},
{"project","factor","oportunity","possible","impact","people","risk","proj","fact","opportun","possibl","peopl"},
{"project","factor","oportunity","organizational","risk","level","proj","fact","opportun","organiz"},
{"project","factor","oportunity","possible","impact","organizational","risk","proj","fact","opportun","possibl","organiz"},
{"project","factor","oportunity","tool","risk","level","proj","fact","opportun"},
{"project","factor","oportunity","possible","impact","tool","risk","proj","fact","opportun","possibl"},
{"project","factor","oportunity","requeriment","risk","level","proj","fact","opportun","requeri"},
{"project","factor","oportunity","possible","impact","requeriment","risk","proj","fact","opportun","possibl","requeri"},
{"project","factor","oportunity","estimation","risk","level","proj","fact","opportun","estim"},
{"project","factor","oportunity","possible","impact","estimation","risk","proj","fact","opportun","possibl","estim"},
{"measurable","system","actual","amount","total","code","line","measur","cod"},
{"measurable","work","estimated","amount","total","code","line","measur","estim","cod"},
{"measurable","system","actual","amount","end","product","code","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","actual","amount","code","end","product","line","measur","cod"},
{"measurable","work","estimated","quantity","code","final","product","line","measur","estim","cod"},
{"measurable","system","number","actual","function","point","final","product","measur","cod"},
{"measurable","work","number","estimated","function","point","final","product","measur","estim","cod"},
{"measurable","system","total","defect","product","inserted","final","measur","insert"},
{"measurable","system","total","defect","removed","product","before","delivery","measur","remov"},
{"measurable","work","estimated","cost","measur","estim"},
{"measurable","oportunity","actual","cost","measur","oportun"},
{"measurable","work","scheduling","complexity","measur","schedul","complex"},
{"measurable","work","complexity","data","measur","complex"},
{"measurable","stakeholder","organizational","complexity","measur","stakehold","organiz","complex"},
{"measurable","work","estimated","time","measur","estim"},
{"measurable","work","actual","duration","time","measur","estim"},
{"measurable","work","development","time","work","product","measur","develop"},
{"measurable","work","estimated","project","development","effort","measur","estim","proj","develop"},
{"measurable","work","actual","project","development","effort","measur","proj","develop"},
{"sociological","team","number","developer","sociolog","develop"},
{"sociological","team","maximum","level","education","sociolog","max","edu"},
{"sociological","team","experience","software","development","sociolog","experi","develop"},

```

```

{"sociological","team","way","acquire","experience","developer","sociolog","experi","develop"},
{"sociological","team","experience","programming","language","implementation","sociolog","experi","program","lang","implement"},
{"sociological","team","way","experience","acquire","language","sociolog","experi","lang"},
{"sociological","team","experience","methodology","develop","sociolog","experi","methodolog"},
{"sociological","team","way","acquire","experience","methodology","sociolog","experi","methodolog"},
{"sociological","team","domain","experience","sociolog","experi"},
{"sociological","team","experience","domain","sociolog","experi"},
{"sociological","team","experience","using","support","tool","sociolog","experi","use"},
{"technological","opportunity","project","application","domain","technolog","opportun","proj"},
{"technological","requirement","paradigm","language","technolog","requir","lang"},
{"technological","requirement","programming","language","technolog","requir","program","lang"},
{"technological","work","planning","tool","technolog","plan"},
{"technological","work","modeling","tool","technolog","model"},
{"technological","work","programming","tool","technolog","program"},
{"technological","system","database","manager","technolog","databas","manag"},
{"technological","requirement","operating","system","technolog","requir","oper"},
{"technological","requirement","type","machine","technolog","requir","machin"},
{"configurable","work","planning","configur","plan"},
{"configurable","stakeholder","business","model","configur","stakehold","busi"},
{"configurable","system","identification","component","configur","identif"},
{"configurable","system","prototype","configur"},
{"configurable","system","requirement","configur","requir"},
{"configurable","work","requirement","review","system","prototype","configur","requir"},
{"configurable","requirement","reverse","engineering","configur","engin"},
{"configurable","system","direct","engineering","configur","engin"}, {"configurable","system","software","prototype","configur"},
{"configurable","software","requirement","configur","requir"},
{"configurable","stakeholder","specification","acceptance","test","configur","stakehold","specif","accept"},
{"configurable","work","review","software","requirement","specification","acceptance","test","configur","requir","specif","accept"},
{"configurable","platform","definition","requirement","restriction","configur","requir","definit"},
{"configurable","system","HLD","software","architecture","configur"},
{"configurable","system","specification","integration","test","configur","specif"},
{"configurable","work","software","architecture","review","configur"},
{"configurable","division","work","structure","configur","div"},
{"configurable","requirement","detailed","design","work","unit","configur","requir","detail"},
{"configurable","system","unit","test","specification","configur","specific"},
{"configurable","work","detailed","design","review","specification","unit","test","configur","detail","specific"},
{"configurable","system","unit","code","configur"},
{"configurable","work","drive","code","review","configur"},
{"configurable","system","unit","test","documentation","configur","document"},
{"configurable","system","integration","test","documentation","configur","integr","document"},
{"configurable","stakeholder","tests","documentation","acceptance","configur","stakehold","document","accept"},
{"configurable","work","post","mortem","report","development","process","configur","develop"},
{"configurable","requirement","type","software","product","assembly","configur","requir","datasource"},
{"configurable","work","set","subprocess","status","configur"},
{"configurable","team","structure","work","configur","name"},
{"configurable","team","way","perform","development","configur","develop"},
{"configurable","work","periodicity","date","configur","period"},
{"configurable","work","schedule","configur","sched"},
{"configurable","work","way","track","control","process","configur"},
{"configurable","work","type","inspection","configur","inspect"},
{"configurable","work","duration","configur"},
{"configurable","work","way","conduct","inspection","configur","inspect"},
{"configurable","work","project","development","model","configur","proj","develop"}

```

A continuación, se da un ejemplo de como se obtuvo el análisis de similitudes con el algoritmo de Levenshtein.

1.- Se muestra en este ejemplo solo dos atributos del Marco para la Caracterización de Proyectos de Software, el atributo X1 Software classification y el atributo X69 Structure of the work team.

```
{ "project", "factor", "opportunity", "software", "classification", "proj", "fact", "opportun", "classif" }, //X1
Software classification
{ "configurable", "team", "structure", "work", "configur", "name" }, //X69 Structure of the work team
```

Como se puede ver el atributo X1 Software clasificacion corresponde al factor Software project type factor y a el alfa Opportunity de los cuales se tomaron las palabras mas relevantes y sus abreviaciones. De esa misma forma para el atributo X69 Structure of the work team correspondiente al factor Configurables Factors y a el alfa Team.

2.- Recordando el algoritmo de Levenshtein nos muestra que la formula a seguir es la siguiente:

$$\text{Afinidad} = 1 - (D / L)$$

D: Distancia de Levenshtein.

L: Longitud de la palabra más larga.

3.- Como ejemplo se toma en cuenta al repositorio de Flossmole:

Repositorio: Alioth

Tabla: al_projects

Columna: unixname

4.- Tomando en cuenta que se van a comparar cada palabra del diccionario con cada columna del repositorio para encontrar la similitud. En este caso la columna unixname se le agrega el nombre de la tabla sin su abreviación del repositorio como se muestra a continuación:

projects_unixname longitud: 17

Esto con la finalidad de que algunos nombres de columnas se repiten en las tablas del repositorio y tienen diferente significado. Se prosigue a calcular la distancia con la siguiente formula:

$$\text{Distancia} = \text{tamColumn} - \text{tamSimilares}$$

tamColumn: Es el tamaño de la columna del repositorio a comparar

tamSimilares: Es el tamaño solamente de las palabras que encuentre similares en el diccionario

5.- A continuación, se muestra la aplicación del algoritmo con los atributos X1 y X69 del Marco para la Caracterización de Proyectos de Software.

projects_unixname longitud: 17

X1

Palabra similar: project

D= 17-7 = 10

Afinidad= 1-(10/17)

1-(0.58)

0.42

0.42*100 = **42%**

X69

Palabra similar: name

D= 17-4 = 13

Afinidad= 1-(13/17)

1-(0.76)

0.24

0.24*100 = **24%**

Aplicando la formula de la Distancia de Levenshtein podemos observar que con el atributo X1 la única palabra similar que se encontró fue project con una longitud de 7. Tomando en cuenta el resultado de la distancia se procede a aplicar la formula del algoritmo de Levenshtein y nos da como resultado un porcentaje del 42. De esta misma forma se hace con el atributo X69, que nos genera un porcentaje del 24. Y este procedimiento se realiza con los 90 atributos del MCPSR. En este ejemplo con estos dos atributos podemos observar que la columna projects_unixname tiene mas similitud con el atributo X1 Software Classification.

Anexo F Documentación de pruebas

En el presente anexo contiene documentación complementaria del Capítulo 7 Pruebas y Resultados, específicamente: F.1) casos de prueba, F.2) procedimiento de pruebas y F.3) registro de pruebas.

F.1 Casos de prueba

En esta sección se especifican los casos de prueba que se llevarán a cabo para probar las funcionalidades de la herramienta ADE-DS, se incluye: un identificador para cada caso de prueba, sus entradas, sus salidas y su objetivo para llevar a cabo su ejecución.

F.1.1 Especificaciones de caso de prueba TC-02

A continuación, se describe el objetivo, la entrada y salida del caso de prueba TC-02.

ID: TC-02

Objetivo: El objetivo de este caso de prueba es validar el cumplimiento del requerimiento Consultar el dw_manager.

Entradas: La entrada corresponde a la conexión con la base de datos dw_manager para elegir alguno de los repositorios que se encuentran cargados, ver Figura F.1. Mostrando las tablas disponibles del repositorio a consultar, ver Figura F.2

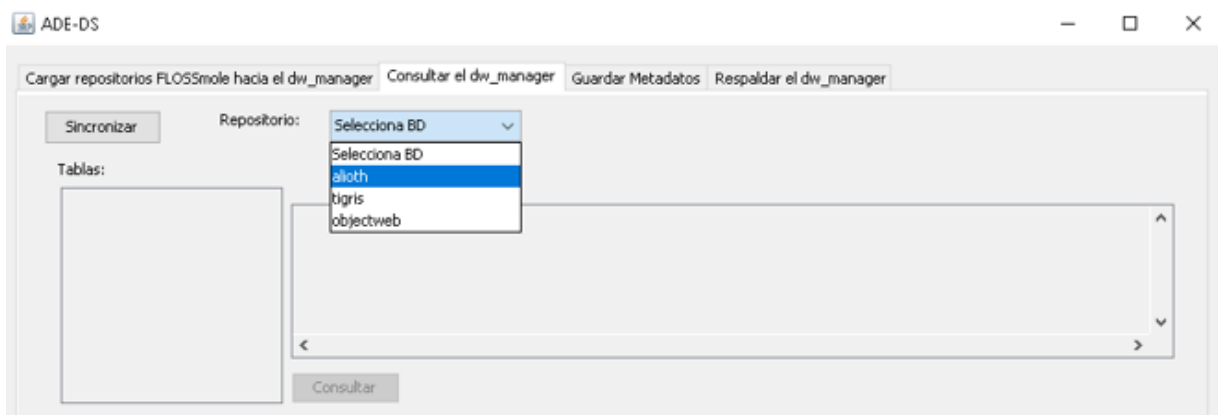


Figura F.1 Sincronización a la base de datos dw_manager

Fuente: creación propia

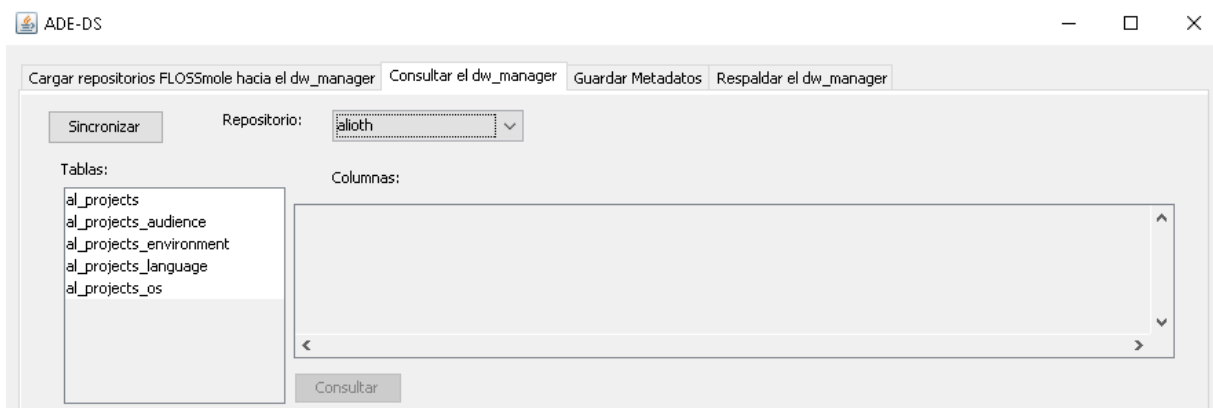


Figura F.2 Tablas disponibles para consultar de los repositorios FLOSSmole cargados
Fuente: creación propia

Salidas: La salida son las columnas de la tabla elegida a consultar mostrando a que MCPS-R es similar, alfa y factor, donde también podemos visualizar con que factor tiene mayor coincidencia ver Figura F.3. Al seleccionar las columnas a consultar se muestra la información general de esta tabla, ver Figura F.4.

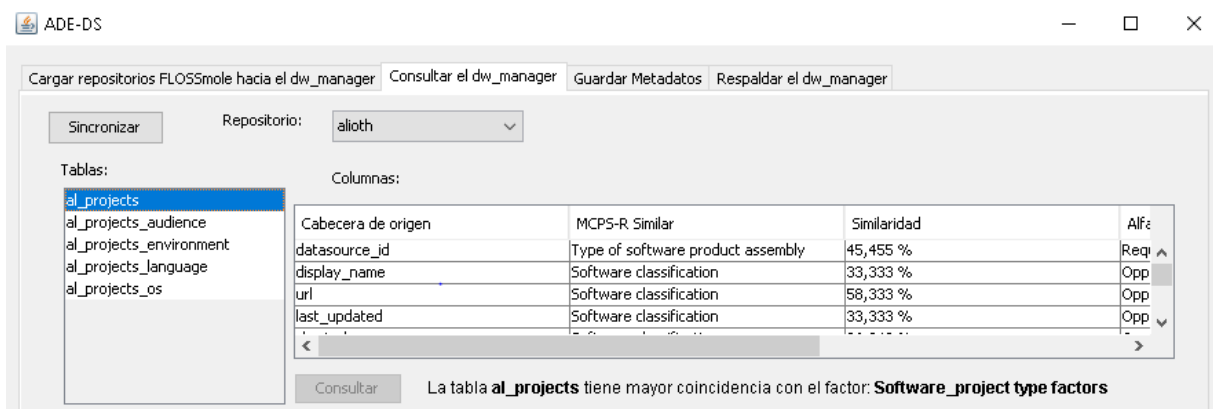


Figura F.3 Columnas de la tabla seleccionada a consultar
Fuente: creación propia

Cargar repositorios FLOSSmole hacia el dw_manager Consultar el dw_manager Guardar Metadatos Respaldar el dw_manager

Sincronizar Repositorio: alioth

Tablas: al_projects, al_projects_audience, al_projects_environment, al_projects_language, al_projects_os

Columnas: Cabecera de origen, MCPS-R Similar, Similaridad, Alf

Cabecera de origen	MCPS-R Similar	Similaridad	Alf
registered	Software classification	36,842 %	Opp
group_id	Software classification	41,176 %	Opp
unixname	Software classification	41,176 %	Opp
real_url	Software classification	41,176 %	Opp

Consultar La tabla **al_projects** tiene mayor coincidencia con el factor: **Software_project type factors**

datasource_id	display_name	url	last_updated	short_desc	regist
325	Advanced Autoconfiguration for Debian	http://alioth.debian.org/projects/aa-p...	2012-07-23 0...	AA or 'Advanced Autoconfiguration' all...	2005-0
325	Adduser	http://alioth.debian.org/projects/addu...	2012-07-23 0...	Tool to add and remove users and gro...	2003-0
325	Advanced Accounting System	http://alioth.debian.org/projects/advacs	2012-07-23 0...	Advacs is a flexible modular accountin...	2004-0
325	Agent79 UNIX Admin Toolset	http://alioth.debian.org/projects/agen...	2012-07-23 0...	Agent79's Unix admin toolset	2004-0
325	the Agnula -- Rocketears-online Project	http://alioth.debian.org/projects/agnu...	2012-07-23 0...	Calling all Musicians who have a compu...	2003-1
325	Debian amavisd-new packages	http://alioth.debian.org/projects/ama...	2012-07-23 0...	amavisd-new: Debian packaging	2003-0
325	amiga-fdisk	http://alioth.debian.org/projects/amig...	2012-07-23 0...	amiga-fdisk is a partitioning program f...	2009-0
325	Anagram Generator	http://alioth.debian.org/projects/anag...	2012-07-23 0...	Generates anagrams for a phrase sup...	2012-0
325	apt	http://alioth.debian.org/projects/apt	2012-07-23 0...	This is Debian's next generation front-...	2007-0
325	apt-build	http://alioth.debian.org/projects/apt-...	2012-07-23 0...	This is an apt-get front-end for compili...	2006-0
325	Apt-Cacher NG	http://alioth.debian.org/projects/apt-...	2012-07-23 0...	Apt-Cacher NG is a caching HTTP prox...	2010-1
325	apt-checkpoint	http://alioth.debian.org/projects/apt-...	2012-07-23 0...	apt-checkpoint is a utility to create sn...	2004-0
325	APT HOWTO	http://alioth.debian.org/projects/apt-...	2012-07-23 0...	APT HOWTO is a document that guide...	2004-0

Figura F.4 Consulta general de la tabla y columnas seleccionadas a consultar
Fuente: creación propia

F.1.2 Especificaciones de caso de prueba TC-03

A continuación, se describe el objetivo, la entrada y salida del caso de prueba TC-03.

ID: TC-03

Objetivo: El objetivo de este caso de prueba es validar el cumplimiento del requerimiento Guardar metadatos.

Entradas: La entrada es la selección del tipo de metadatos a consultar, operacionales o de consultar, ver Figura F.5.

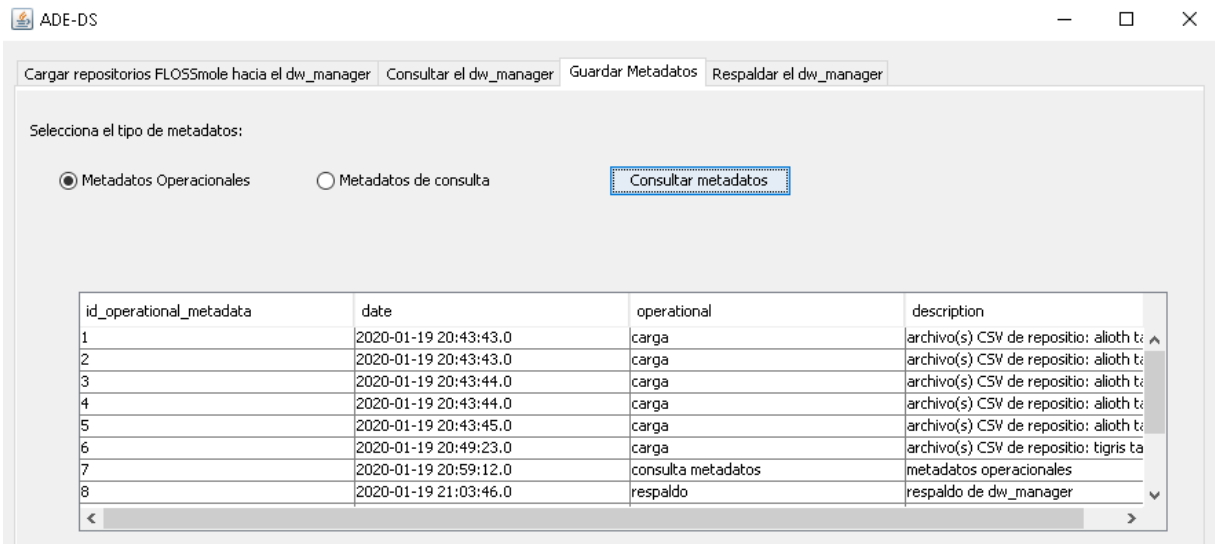
Cargar repositorios FLOSSmole hacia el dw_manager Consultar el dw_manager Guardar Metadatos Respaldar el dw_manager

Selecciona el tipo de metadatos:

Metadatos Operacionales Metadatos de consulta

Figura F.5 Metadatos operacionales o de consulta a seleccionar
Fuente: creación propia

Salidas: La salida es mostrar la información de los metadatos consultados, ver Figura F.6 y F.7.



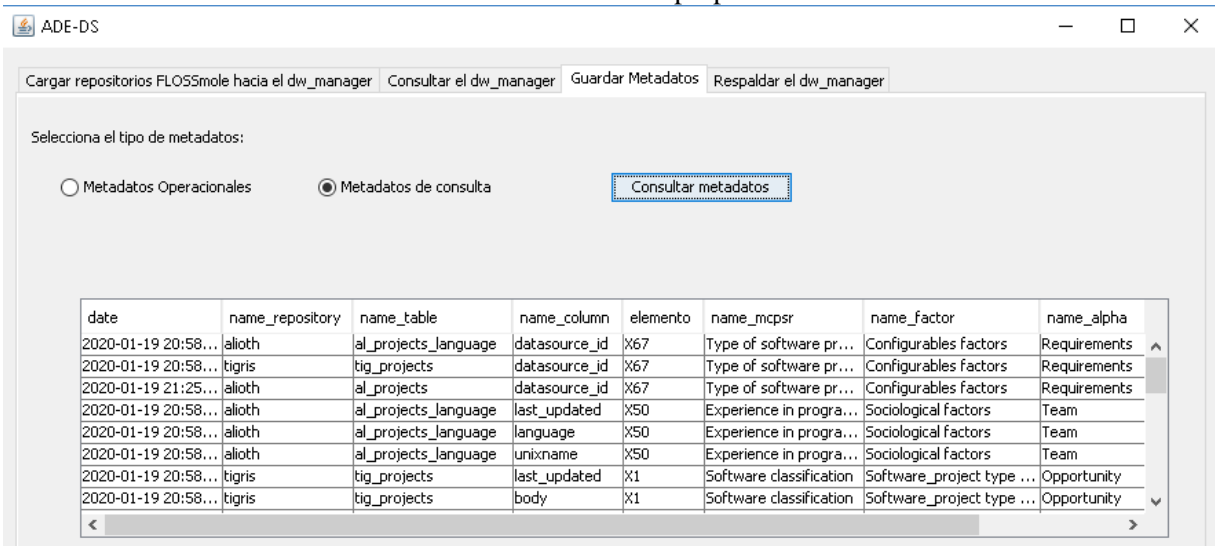
Selección de metadatos operacionales:

Metadatos Operacionales Metadatos de consulta

id_operational_metadatos	date	operational	description
1	2020-01-19 20:43:43.0	carga	archivo(s) CSV de repositorio: alioth t...
2	2020-01-19 20:43:43.0	carga	archivo(s) CSV de repositorio: alioth t...
3	2020-01-19 20:43:44.0	carga	archivo(s) CSV de repositorio: alioth t...
4	2020-01-19 20:43:44.0	carga	archivo(s) CSV de repositorio: alioth t...
5	2020-01-19 20:43:45.0	carga	archivo(s) CSV de repositorio: alioth t...
6	2020-01-19 20:49:23.0	carga	archivo(s) CSV de repositorio: tigris ta...
7	2020-01-19 20:59:12.0	consulta metadatos	metadatos operacionales
8	2020-01-19 21:03:46.0	respaldo	respaldo de dw_manager

Figura F.6 Metadatos operacionales consultados

Fuente: creación propia



Selección de metadatos de consulta:

Metadatos Operacionales Metadatos de consulta

date	name_repository	name_table	name_column	elemento	name_mcpsr	name_factor	name_alpha
2020-01-19 20:58...	alioth	al_projects_language	datasource_id	X67	Type of software pr...	Configurables factors	Requirements
2020-01-19 20:58...	tigris	tig_projects	datasource_id	X67	Type of software pr...	Configurables factors	Requirements
2020-01-19 21:25...	alioth	al_projects	datasource_id	X67	Type of software pr...	Configurables factors	Requirements
2020-01-19 20:58...	alioth	al_projects_language	last_updated	X50	Experience in progra...	Sociological factors	Team
2020-01-19 20:58...	alioth	al_projects_language	language	X50	Experience in progra...	Sociological factors	Team
2020-01-19 20:58...	alioth	al_projects_language	unixname	X50	Experience in progra...	Sociological factors	Team
2020-01-19 20:58...	tigris	tig_projects	last_updated	X1	Software classification	Software_project type ...	Opportunity
2020-01-19 20:58...	tigris	tig_projects	body	X1	Software classification	Software_project type ...	Opportunity

Figura F.7 Metadatos de consulta consultados

Fuente: creación propia

F.1.3 Especificaciones de caso de prueba TC-04

A continuación, se describe el objetivo, la entrada y salida del caso de prueba TC-04.

ID: TC-04

Objetivo: El objetivo de este caso de prueba es validar el cumplimiento del requerimiento Respaldo el dw_manager.

Entradas: La entrada es la ubicación para realizar el respaldo de la base de datos dw_manager, ver Figura F.8.

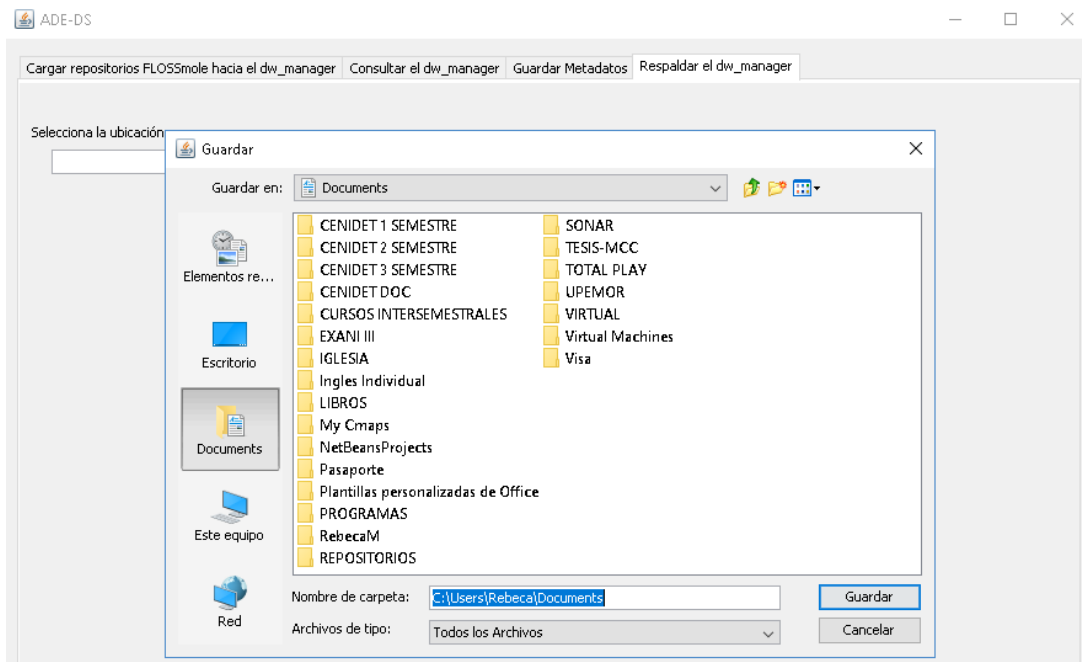


Figura F.8 Seleccionar ubicación para realizar respaldo dw_manager
Fuente: creación propia

Salidas: La salida es la realización del backup de la base de datos dw_manager, ver Figura F.9 y Figura F10.

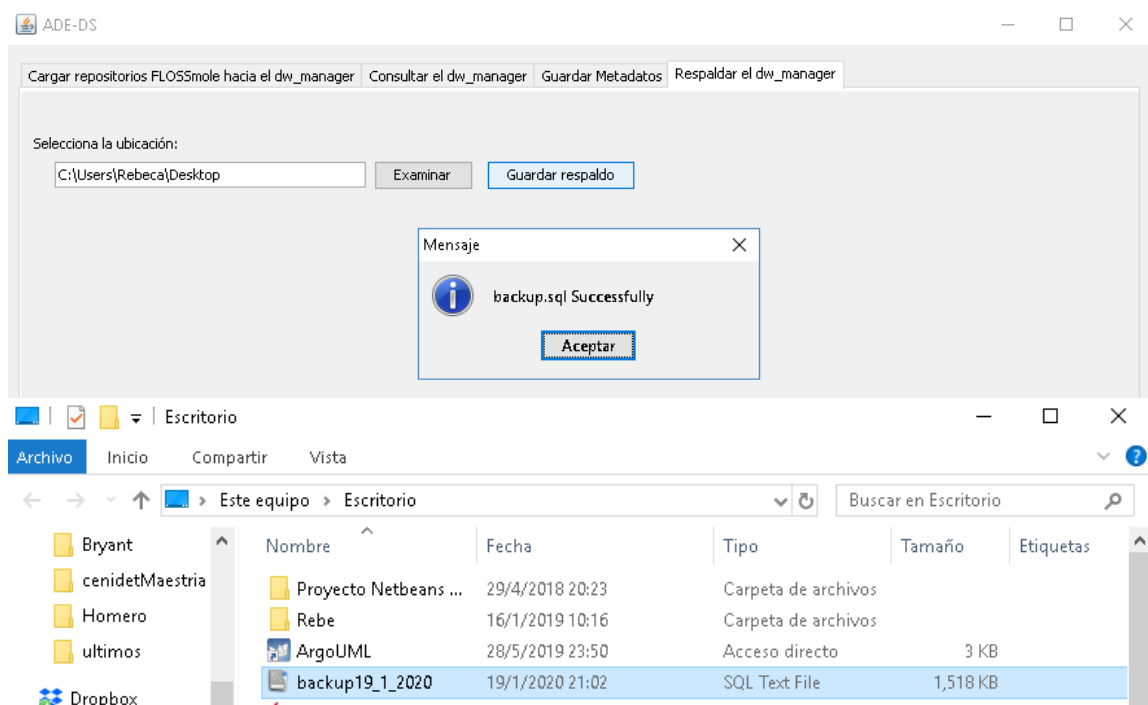


Figura F.9 Respaldo de dw_manager generado
Fuente: creación propia

```

23
24 • CREATE DATABASE /*!32312 IF NOT EXISTS*/ `dw_manager` /*!40100 DEFAULT CHARACTER SET utf8 */;
25
26 • USE `dw_manager`;
27
28 --
29 -- Table structure for table `al_projects`
30 --
31
32 • DROP TABLE IF EXISTS `al_projects`;
33 • /*!40101 SET @saved_cs_client = @@character_set_client */;
34 • /*!40101 SET character_set_client = utf8 */;
35 • CREATE TABLE `al_projects` (
36   `display_name` varchar(255) DEFAULT NULL,
37   `url` varchar(255) DEFAULT NULL,
38   `last_updated` datetime DEFAULT NULL,
39   `short_desc` mediumtext,
40   `datasource_id` int(11) DEFAULT NULL,
41   `registered` date DEFAULT NULL,
42   `group_id` int(11) DEFAULT NULL,
43   `username` varchar(255) DEFAULT NULL,
44   `real_url` varchar(255) DEFAULT NULL
45 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
46 • /*!40101 SET character_set_client = @saved_cs_client */;
47
48 --
49 -- Dumping data for table `al_projects`
50 --

```

Figura F.10 Script del respaldo realizado

Fuente: creación propia

F.2 Procedimiento de Pruebas

En esta sección se describen los diferentes procedimientos a seguir para llevar a cabo la ejecución de cada uno de los casos de prueba, de manera que se cumpla con las funcionalidades de la herramienta ADE-DS.

F.2.1 Procedimiento de prueba TPr-02

Entradas, salidas y requerimientos especiales

Documentos:

Plan de prueba

Diseño de prueba

Procedimiento de Prueba

Reporte de Prueba

Servidores:

MySQL

Aplicaciones:

XAMPP

Datos:

Conexión con la base de datos dw_manager

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

Reportes: La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

Preparación: Iniciar todas las aplicaciones y servidores, antes de comenzar con el procedimiento.

Inicio: En la interfaz de la herramienta, el usuario da clic en sincronizar para realizar la conexión con la base de datos dw_manager.

Proceso: El usuario debe seleccionar el repositorio FLOSSmole y tabla que se encuentren cargados en la base de datos dw_manager a consultar.

Medición: Depende del tiempo que la herramienta establezca la conexión a la base de datos dw_manager.

Suspender: El componente Consultar el dw_manager se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

Reiniciar: La prueba puede ser reiniciada inmediatamente después de finalizar su ejecución.

Cierre: La prueba termina con la información general de las tablas consultadas.

Contingencias: Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

F.2.2 Procedimiento de prueba TPr-03**Entradas, salidas y requerimientos especiales****Documentos:**

Plan de prueba

Diseño de prueba
Procedimiento de Prueba
Reporte de Prueba

Servidores:

MySQL

Aplicaciones:

XAMPP

Datos:

Tipos de metadatos

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

Reportes: La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

Preparación: Iniciar todas las aplicaciones y servidores, antes de comenzar con el procedimiento.

Inicio: En la interfaz de la herramienta, el usuario selecciona el tipo de metadatos a consultar, operacional o consulta.

Proceso: La herramienta se encarga de realizar la consulta hacia la base de datos dw_manager.

Medición: Depende del tiempo que la herramienta establezca la conexión a la base de datos dw_manager.

Suspender: El componente Guardar metadatos se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

Reiniciar: La prueba puede ser reiniciada inmediatamente después de finalizar su ejecución.

Cierre: La prueba termina mostrando la información de los metadatos consultados.

Contingencias: Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

F.2.3 Procedimiento de prueba TPr-04

Entradas, salidas y requerimientos especiales

Documentos:

Plan de prueba

Diseño de prueba

Procedimiento de Prueba

Reporte de Prueba

Servidores:

MySQL

Aplicaciones:

XAMPP

Datos:

Ubicación o dirección para respaldo

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

Reportes: La elaboración de los reportes es de forma manual al hacer una inspección visual de los eventos que ocurren a lo largo de la ejecución de la prueba y serán registrados en el documento de reporte de pruebas, de esta misma forma los errores, problemas o anomalías serán registrados en el mismo documento.

Preparación: Iniciar todas las aplicaciones y servidores, antes de comenzar con el procedimiento.

Inicio: En la interfaz de la herramienta, el usuario selecciona la ubicación para realizar el respaldo de la base de datos dw_manager.

Proceso: La herramienta se encarga de realizar el backup de la base de datos dw_manager.

Medición: Depende del tiempo que la herramienta tarde en realizar el respaldo de la base de datos dw_manager.

Suspender: El componente Respaldo el dw_manager se ejecuta a cierto tiempo por lo que terminado ese plazo se puede detener la prueba sin problemas.

Reiniciar: La prueba puede ser reiniciada inmediatamente después de finalizar su ejecución.

Cierre: La prueba termina con backup generado de la base de datos dw_manager.

Contingencias: Al presentarse anomalías la prueba tiene que ser suspendida y el error o fallo corregidos. Para poder reiniciar la prueba de debe haber corregido la anomalía.

F.3 Registro de Pruebas

En esta sección proporciona un registro cronológico de los detalles relevantes sobre la ejecución de las pruebas ejecutadas, para probar la funcionalidad de los elementos de prueba de la herramienta ADE-DS.

F.3.1 Registro de prueba TL-02

El registro de prueba TL-02 cubre el procedimiento de prueba TPr-02 correspondiente al caso de prueba TC-02. Los datos usados en la ejecución de la prueba son del repositorio Alioth.

Actividades y eventos de la prueba

A continuación, se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-02.

Paso	Actor	Descripción	Verificado
1	Usuario	Hacer clic en el botón sincronizar.	Si
2	Herramienta	Realiza la conexión a base de datos dw_manager	Si
3	Usuario	Seleccionar el repositorio a consultar disponible en la base de datos dw_manager	Si
4	Usuario	Selecciona tabla a consultar	Si
5	Herramienta	Muestra las columnas de la tabla seleccionada con su MCPS-R, alfa y factor correspondiente	Si
6	Usuario	Selecciona la columna o columnas a consultar y da clic en el botón consultar	Si
7	Herramienta	Muestra la información general de la tabla y columnas seleccionadas	Si

Anomalías

No se reportan anomalías.

Ejemplo de consulta para la prueba

A continuación, se muestra consulta de la tabla al_projects del repositorio Alioth que se encuentra en la base de datos dw_manager ver Figura F.11 con el fin de probar el requerimiento funcional Consultar el dw_manager, ver Figura F.12.

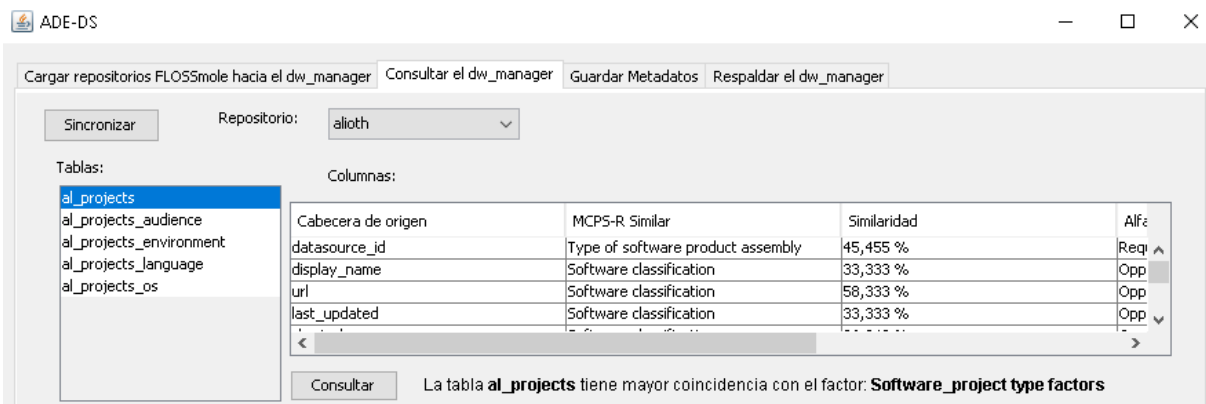


Figura F.11 Columnas de la tabla al_projects a consultar
Fuente: creación propia

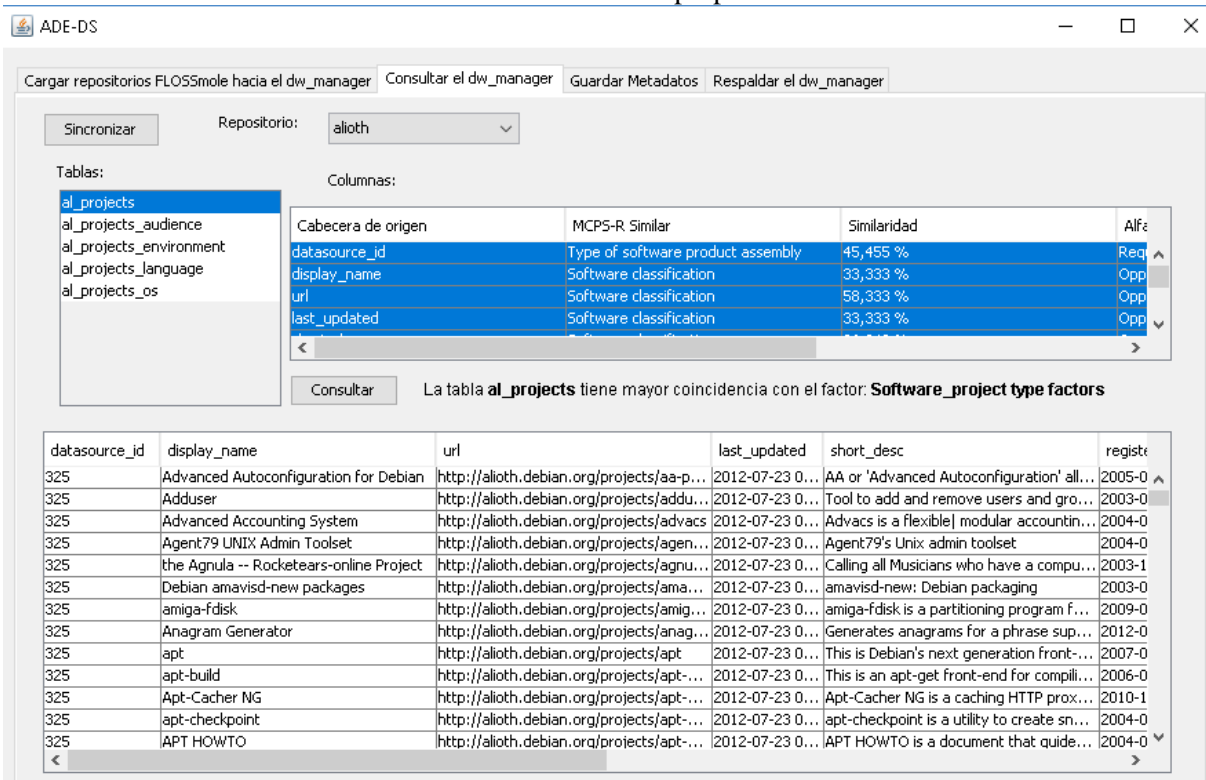


Figura F.12 Consulta general de al_projects y columnas seleccionadas a consultar
Fuente: creación propia

F.3.2 Registro de prueba TL-03

El registro de prueba TL-03 cubre el procedimiento de prueba TPr-03 correspondiente al caso de prueba TC-03. Los datos usados en la ejecución de la prueba son del repositorio Alioth.

Actividades y eventos de la prueba

A continuación, se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-03.

Id	Actor	Descripción	Verificado
1	Usuario	Selecciona el tipo de metadato a consultar	Si
2	Usuario	Hace clic en el botón consultar metadatos	Si
3	Herramienta	Muestra los metadatos solicitados	Si

Anomalías

No se reportan anomalías.

Ejemplo de selección de metadatos para la Prueba

A continuación, se muestra la selección del tipo de metadatos a consultar ver Figura F.13 con el fin de probar el requerimiento funcional Guardar metadatos, ver Figura F.14 y Figura F.15.

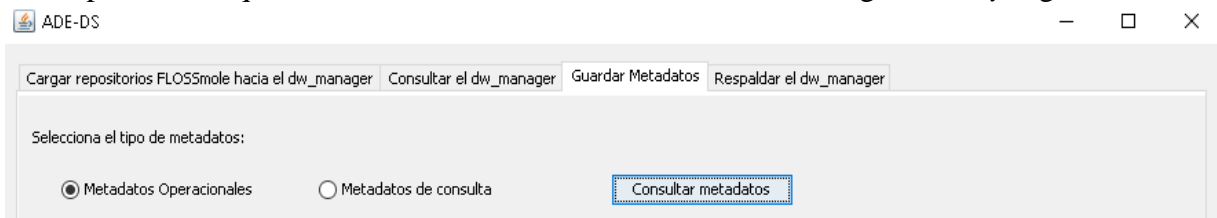


Figura F.13 Metadatos operacionales o de consulta seleccionados
Fuente: creación propia

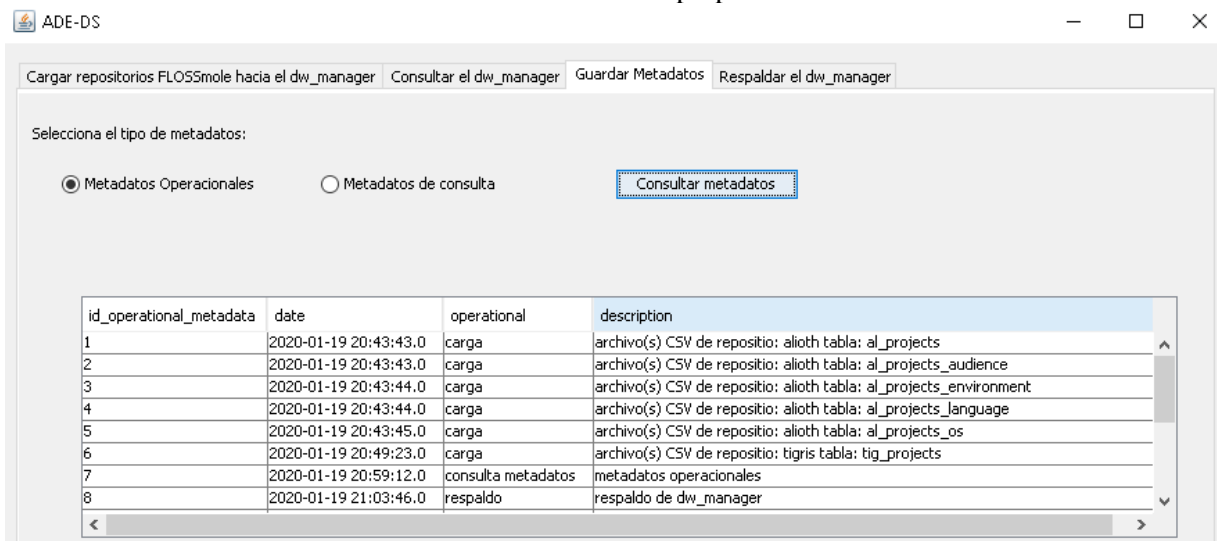


Figura F.14 Metadatos operacionales consultados
Fuente: creación propia

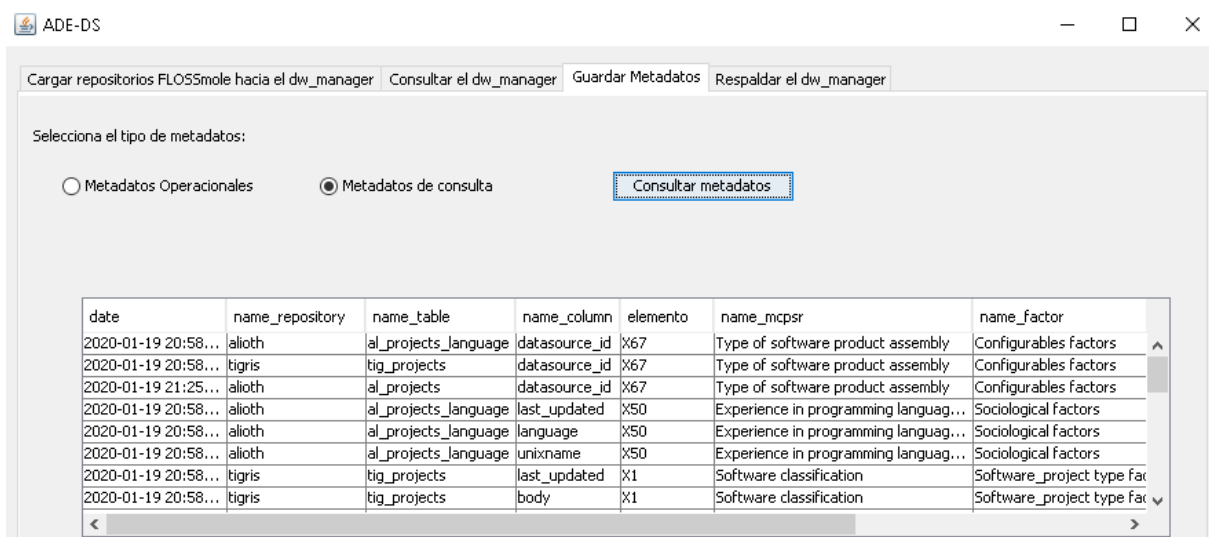


Figura F.15 Metadatos de consulta consultados

Fuente: creación propia

F.3.3 Registro de prueba TL-04

El registro de prueba TL-04 cubre el procedimiento de prueba TPr-04 correspondiente al caso de prueba TC-04. Los datos usados en la ejecución de la prueba son de la base de datos dw_manager.

Actividades y Eventos de la Prueba

A continuación, se describe el registro de eventos y actividades al ejecutar el procedimiento de prueba TPr-04.

Id	Actor	Descripción	Verificado
1	Usuario	Hace clic en el botón examinar	Si
2	Herramienta	Abre ventana para seleccionar la ubicación del respaldo de la base de datos dw_manager	Si
3	Usuario	Elige ubicación del respaldo	Si
4	Usuario	Hace clic en el botón guardar respaldo	Si
5	Herramienta	Genera backup de la base de datos dw_manager	Si

Anomalías

No se reportan anomalías.

Ejemplo de backup realizado para la Prueba

A continuación, se muestra la selección de ubicación para realizar el backup de la base de datos dw_manager ver Figura F.16 con el fin de probar el requerimiento funcional Respaldar el dw_manager, ver Figura F.17 y Figura F.18.

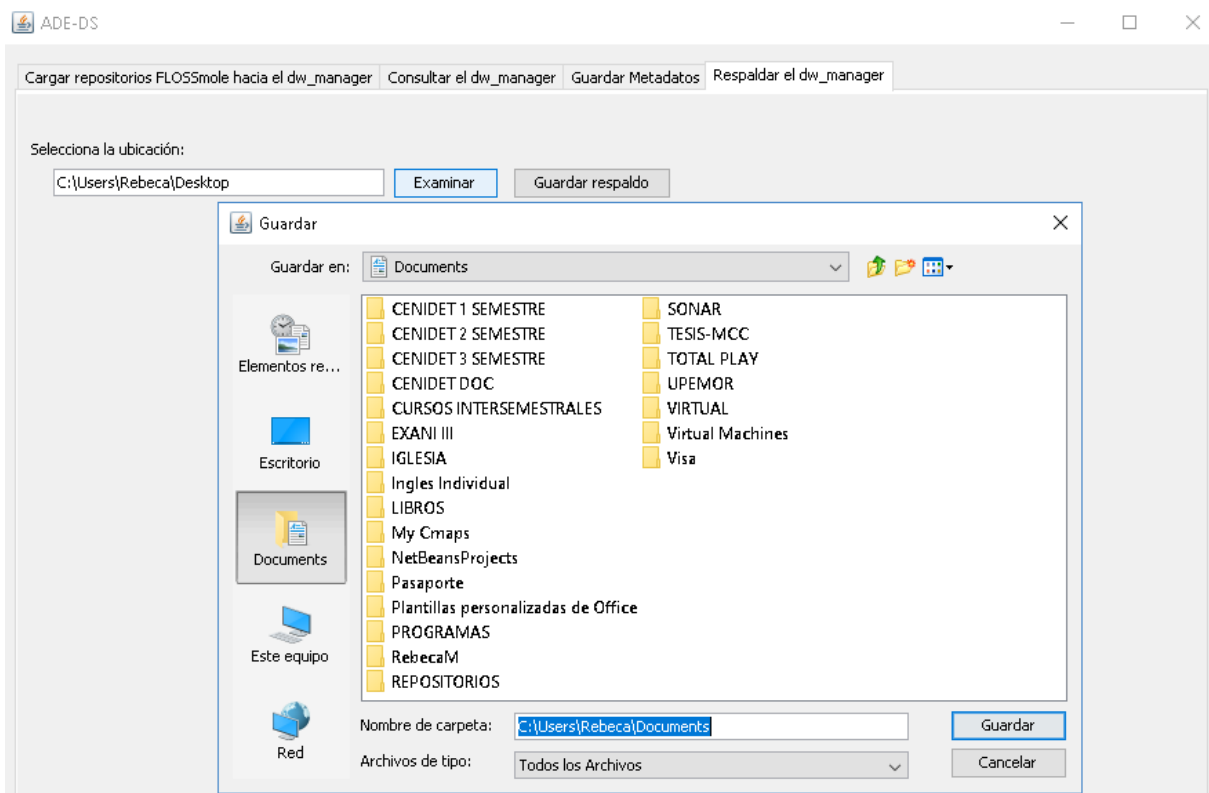


Figura F.16 Seleccionar ubicación para realizar el respaldo de dw_manager
Fuente: creación propia

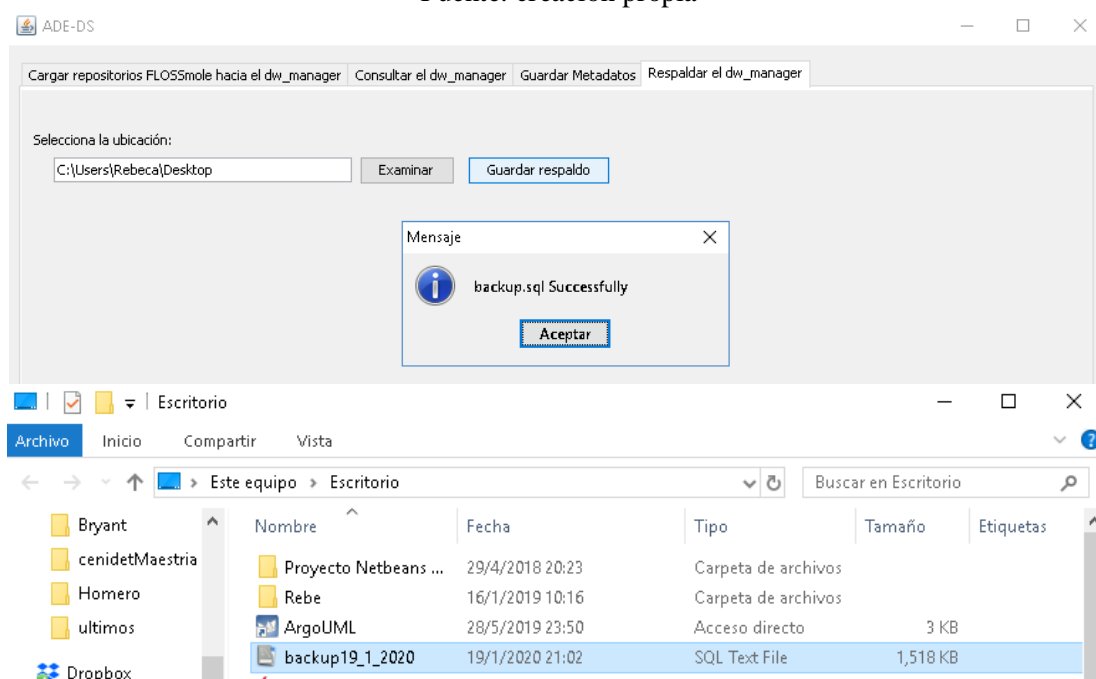


Figura F.17 Respaldo de dw_manager generado
Fuente: creación propia

```

23
24 • CREATE DATABASE /*!32312 IF NOT EXISTS*/ `dw_manager` /*!40100 DEFAULT CHARACTER SET utf8 */;
25
26 • USE `dw_manager`;
27
28 --
29 -- Table structure for table `al_projects`
30 --
31
32 • DROP TABLE IF EXISTS `al_projects`;
33 • /*!40101 SET @saved_cs_client = @@character_set_client */;
34 • /*!40101 SET character_set_client = utf8 */;
35 • CREATE TABLE `al_projects` (
36   `display_name` varchar(255) DEFAULT NULL,
37   `url` varchar(255) DEFAULT NULL,
38   `last_updated` datetime DEFAULT NULL,
39   `short_desc` mediumtext,
40   `datasource_id` int(11) DEFAULT NULL,
41   `registered` date DEFAULT NULL,
42   `group_id` int(11) DEFAULT NULL,
43   `unixname` varchar(255) DEFAULT NULL,
44   `real_url` varchar(255) DEFAULT NULL
45 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
46 • /*!40101 SET character_set_client = @saved_cs_client */;
47
48 --
49 -- Dumping data for table `al_projects`
50 --

```

Figura F.18 Script del respaldo realizado
Fuente: creación propia