



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Tecnologías de la Información

**Sistema para la creación y
mantenimiento de una red de
conocimiento de Tópicos
Musicales**

Autora:

Dña. Ester García Pérez

Tutor:

D. César Llamas Bello

Resumen

“De la Emoción al Sentido” es un proyecto docente de la teoría de tópicos que comienza un equipo de profesores y antiguos alumnos del grado de Historia y Ciencias de la Música de la Universidad de Valladolid para ayudar al alumnado de dicho grado en la asignatura Música y pensamiento en el siglo de las luces (siglo XVIII).

Los tópicos musicales son temas del discurso musical que transmiten significados y emociones reconocibles de nuestra cultura y comparten los mismos códigos de ritmo, dinámica, tempo, armonía, timbre, ...(significante) en las obras de los compositores de Música Clásica occidental desde siglo XVIII. [1]

Este trabajo pretende analizar, comprender y desarrollar una aplicación para la recopilación de información de los distintos tópicos musicales y el favorecimiento hacia la comprensión de dicha información por parte del alumnado. Desde este punto de vista, nuestra aplicación pretende servir de herramienta para el soporte y gestión de una red de conocimiento de expertos y usuarios, y soportar un grafo de conocimiento en forma de tripletas RDF. Por último, en este documento se encuentra todo el material obtenido como consecuencia de los requisitos, el análisis, el diseño, la implementación y las pruebas.

Palabras Clave: Tópicos Musicales, Red de Conocimiento, Grafo de Conocimiento, RDF, Aplicación Software, Musicología.

Abstract

“De la Emoción al Sentido” is a teaching project on topic theory that a team of professors and former students of the History and Music Sciences degree at the University of Valladolid begins to help the students of said degree in the subject Music and thought in the century of enlightenment (18th century).

Musical topics are musical themes of musical discourse that transmit recognizable meanings and emotions of our culture and share the same codes of rhythm, dynamics, tempo, harmony, timbre, ... (signifier) in the works of composers of Western Classical Music since the 18th century.

This work aims to analyze, understand and develop an application for collecting information on different musical topics and promoting the understanding of said information by students. From this point of view, our application aims to serve as a tool for the support and management of a knowledge network of experts and users and support a knowledge graph in the form of RDF triplets. Finally, this document contains all the material obtained because of the requirements, analysis, design, implementation and testing.

Keywords: Musical Topics, Knowledge Network, Knowledge Graph, RDF, Software Application, Musicology.

Tabla de contenidos

| | |
|--|-----------|
| CAPÍTULO 1 - INTRODUCCIÓN | 13 |
| 1.1 MOTIVACIÓN | 13 |
| 1.2 OBJETIVOS..... | 14 |
| 1.3 ESTRUCTURA DE LA MEMORIA..... | 14 |
| 1.4 AGRADECIMIENTOS | 14 |
| CAPÍTULO 2 - PLANIFICACIÓN DEL PROYECTO | 15 |
| 2.1 METODOLOGÍA DE TRABAJO | 15 |
| 2.2 ALCANCE | 16 |
| 2.2.1 Estructura de Descomposición del Trabajo (actividades) – WBS | 16 |
| 2.2.2 Estructura de Descomposición del Producto y Entregables – PBS..... | 18 |
| 2.3 CRONOGRAMAS..... | 18 |
| 2.3.1 Diagrama PERT..... | 18 |
| 2.3.2 Diagrama Gantt..... | 19 |
| 2.4 ASIGNACIÓN DE RECURSOS..... | 20 |
| 2.5 PRESUPUESTO | 21 |
| 2.6 ANÁLISIS DE RIESGOS | 22 |
| CAPÍTULO 3 - CONTEXTO Y ANTECEDENTES DEL PROYECTO | 27 |
| 3.1 INTRODUCCIÓN | 27 |
| 3.2 RDF | 27 |
| 3.2.1 ¿Qué es RDF? | 27 |
| 3.2.2 Grafos de conocimiento | 28 |
| 3.2.3 Herramientas de visualización | 29 |
| 3.2.3 Uso en el proyecto..... | 33 |
| 3.3 UML..... | 34 |
| 3.3.1 Diagramas UML..... | 34 |
| 3.3.2 Estereotipos..... | 35 |
| 3.5 TECNOLOGÍAS DE BASES DE DATOS | 35 |
| 3.5.1 Base de datos relacional..... | 35 |
| 3.5.2 Lenguaje SQL | 36 |
| 3.5.3 MySQL..... | 36 |
| 3.6 LENGUAJE JAVA..... | 37 |
| CAPÍTULO 4 - ANÁLISIS DEL SISTEMA | 39 |
| 4.1 INTRODUCCIÓN | 39 |
| 4.2 ANÁLISIS DE REQUISITOS | 39 |
| 4.3 CASOS DE USO | 41 |
| 4.3.1 Actores..... | 41 |
| 4.3.2 Diagrama de casos de uso | 41 |
| 4.3.3 Especificación de casos de uso..... | 44 |
| CAPÍTULO 5 - ARQUITECTURA Y DISEÑO | 53 |
| 5.1 ARQUITECTURA LÓGICA..... | 53 |
| 5.2 DISEÑO DE LA BASE DE DATOS | 54 |
| 5.2.1 Modelo entidad-relación..... | 54 |
| 5.2.2 Base de datos del sistema | 54 |
| 5.3 DIAGRAMA DE CLASES | 56 |
| 5.4 MÁQUINA DE ESTADOS | 58 |
| 5.4.1 Concepto | 58 |
| 5.4.2 Diagrama de máquina de estados | 58 |
| 5.5 DIAGRAMA DE SECUENCIA | 60 |
| 5.5.1 Inicio de sesión | 60 |
| 5.5.2 Ver tripletas | 62 |
| 5.5.3 Inspeccionar tripletas..... | 63 |
| 5.6 INTERFAZ DE USUARIO..... | 66 |

| | |
|--|------------|
| 5.6.1 Decisiones de diseño | 66 |
| 5.6.2 Bocetos | 66 |
| CAPÍTULO 6 - IMPLEMENTACIÓN | 69 |
| 6.1 HERRAMIENTAS UTILIZADAS | 69 |
| 6.1.1 Librerías utilizadas | 70 |
| 6.2 DESARROLLO | 70 |
| 6.2.1 Fase 1 | 71 |
| 6.2.2 Fase 2 | 71 |
| 6.2.3 Fase 3 | 80 |
| CAPÍTULO 7 - PRUEBAS..... | 83 |
| 7.1 INTRODUCCIÓN | 83 |
| 7.2 LISTADO DE PRUEBAS | 83 |
| 7.3 ESPECIFICACIÓN LISTADO DE PRUEBAS..... | 85 |
| CAPÍTULO 8 - CONCLUSIONES Y LÍNEAS FUTURAS..... | 97 |
| 8.1 CONCLUSIONES | 97 |
| 8.2 LÍNEAS FUTURAS | 97 |
| BIBLIOGRAFÍA | 99 |
| GLOSARIO..... | 107 |
| ANEXO | 109 |
| A1. MANUAL DE USUARIO | 109 |
| A2. INSTALACIÓN | 109 |

Índice de ilustraciones

| | |
|---|----|
| ILUSTRACIÓN 1 - MODELO EN CASCADA DE CINCO NIVELES | 15 |
| ILUSTRACIÓN 2 - ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO (ACTIVIDADES) DEL PROYECTO | 17 |
| ILUSTRACIÓN 3 - ESTRUCTURA DE DESCOMPOSICIÓN DEL PRODUCTO Y ENTREGABLES DEL PROYECTO | 18 |
| ILUSTRACIÓN 4 - DIAGRAMA PROGRAM EVALUATION AND REVIEW TECHNIQUES DEL PROYECTO | 19 |
| ILUSTRACIÓN 5 - DIAGRAMA GANTT DEL PROYECTO | 20 |
| ILUSTRACIÓN 6 - MODELO MATRIZ DE RIESGOS..... | 26 |
| ILUSTRACIÓN 7 - MATRIZ DE RIESGOS DEL PROYECTO | 26 |
| ILUSTRACIÓN 8 - TRIPLETA SEMÁNTICA | 27 |
| ILUSTRACIÓN 9 - EJEMPLO DE TRIPLETA SEMÁNTICA | 27 |
| ILUSTRACIÓN 10 - EJEMPLO DE GRAFO DE CONOCIMIENTO [28] | 28 |
| ILUSTRACIÓN 11 - CATEGORÍAS IMPORTANTES DE INVESTIGACIONES ACTUALES DE LOS GRAFOS DE CONOCIMIENTO | 29 |
| ILUSTRACIÓN 12 - APLICACIONES BASADAS EN GRAFOS DE CONOCIMIENTO EN EL ÁMBITO EDUCATIVO | 29 |
| ILUSTRACIÓN 13 - CÓDIGOS TURTLE DEL PROYECTO | 30 |
| ILUSTRACIÓN 14 - VISUALIZACIÓN DEL CÓDIGO 1 DE TURTLE CON :ISSEMANTIC | 31 |
| ILUSTRACIÓN 15 - RELACIONES DEL RECURSO MARIO DEL EJEMPLO DE CÓDIGO TURTLE | 31 |
| ILUSTRACIÓN 16 - RELACIONES DEL RECURSO LUIGI DEL EJEMPLO DEL CÓDIGO TURTLE..... | 32 |
| ILUSTRACIÓN 17 - RELACIONES DEL LITERAL “FONTANERO” DEL EJEMPLO DEL CÓDIGO TURTLE..... | 32 |
| ILUSTRACIÓN 18 - VISUALIZACIÓN DEL CÓDIGO 1 DE TURTLE CON RDF GRAPHER | 32 |
| ILUSTRACIÓN 19 - VISUALIZACIÓN DEL CÓDIGO 2 DE TURTLE CON RDF GRAPHER | 33 |
| ILUSTRACIÓN 20 - VISUALIZACIÓN DEL CÓDIGO 1 DE TURTLE CON RDF SHAPE | 33 |
| ILUSTRACIÓN 21 - VISUALIZACIÓN DEL CÓDIGO 2 DE TURTLE CON RDF SHAPE | 33 |
| ILUSTRACIÓN 22 - CLASES ESTEREOTIPADAS DE UML | 35 |
| ILUSTRACIÓN 23 - DIAGRAMA DE CASOS DE USO DEL SISTEMA..... | 43 |
| ILUSTRACIÓN 24 - ESQUEMA QUE DESCRIBE LA IGU DEL PATRÓN MVP | 53 |
| ILUSTRACIÓN 25 - MODELO ENTIDAD-RELACIÓN DE LA BASE DE DATOS DEL SISTEMA..... | 56 |
| ILUSTRACIÓN 26 - DIAGRAMA DE CLASES VISTA DEL SISTEMA | 57 |
| ILUSTRACIÓN 27 - DIAGRAMA DE CLASES PRESENTADOR DEL SISTEMA..... | 57 |
| ILUSTRACIÓN 28 - DIAGRAMA DE CLASES MODELO DEL SISTEMA..... | 58 |
| ILUSTRACIÓN 29 - DIAGRAMA DE MÁQUINA DE ESTADOS DEL SISTEMA..... | 59 |
| ILUSTRACIÓN 30 - DIAGRAMA DE SECUENCIA - INICIAR SESI..... | 61 |
| ILUSTRACIÓN 31 - DIAGRAMA DE SECUENCIA - VER TRIPLETS..... | 63 |
| ILUSTRACIÓN 32 - DIAGRAMA DE SECUENCIA - INSPECCIONAR TRIPLETA | 65 |
| ILUSTRACIÓN 33 - BOCETO CU-001..... | 66 |
| ILUSTRACIÓN 34 - BOCETO CU-002..... | 66 |
| ILUSTRACIÓN 35 - BOCETO CU-009..... | 67 |
| ILUSTRACIÓN 36 – ARQUITECTURA DE DOS NIVELES DE JDBC | 70 |
| ILUSTRACIÓN 37 - PARTE DEL SCRIPT DDL Y DML DE LA BASE DE DATOS..... | 71 |
| ILUSTRACIÓN 38 - PARTE DEL CÓDIGO DE LA CLASE CONEXION..... | 72 |
| ILUSTRACIÓN 39 - VISTA DE INICIO DE SESIÓN | 72 |
| ILUSTRACIÓN 40 - VISTA MENÚ (ADMINISTRADOR)..... | 73 |
| ILUSTRACIÓN 41 - VISTA ADMINISTRACIÓN DE USUARIOS | 73 |
| ILUSTRACIÓN 42 - VISTA ADMINISTRACIÓN DE SUGERENCIAS | 73 |
| ILUSTRACIÓN 43 - VISTA DE TRIPLETS (LECTOR)..... | 74 |
| ILUSTRACIÓN 44 - VISTA DETALLES TRIPLETA | 74 |
| ILUSTRACIÓN 45 - VISTA DETALLES OBRA | 75 |
| ILUSTRACIÓN 46 - VISTA DETALLES BIBLIOGRAFÍA..... | 75 |
| ILUSTRACIÓN 47 - VISTA LISTA DE TRIPLETS (COLABORADOR)..... | 76 |
| ILUSTRACIÓN 48 - VISTA CREACIÓN SUGERENCIA | 76 |
| ILUSTRACIÓN 49 - VISTA CREACIÓN TRIPLETA..... | 77 |
| ILUSTRACIÓN 50 - VISTA CREACIÓN OBRA..... | 77 |
| ILUSTRACIÓN 51 - VISTA CREACIÓN BIBLIOGRAFÍA..... | 78 |
| ILUSTRACIÓN 52 - VISTA LISTA DE TRIPLETS (MODERADOR)..... | 78 |
| ILUSTRACIÓN 53 - VISTA MENÚ (MODERADOR) | 79 |
| ILUSTRACIÓN 54 - VISTA OTROS ELEMENTOS..... | 79 |
| ILUSTRACIÓN 55 - VISTA VALIDACIÓN TRIPLETS | 80 |
| ILUSTRACIÓN 56 - VISTA GRAPH STREAM | 81 |

| | |
|--|-----|
| ILUSTRACIÓN 57 - EJEMPLO GRAFO RDF CON DATOS MUSICALES (:ISSEMANTIC) | 82 |
| ILUSTRACIÓN 58 - EJEMPLO GRAFO RDF CON DATOS MUSICALES (RDF GRAPHER) | 82 |
| ILUSTRACIÓN 59 - EJEMPLO GRAFO RDF CON DATOS MUSICALES (RDF SHAPE) | 82 |
| ILUSTRACIÓN 60 - INSTALACIÓN PASO 3.1 | 110 |
| ILUSTRACIÓN 61 - INSTALACIÓN PASO 3.2 | 110 |
| ILUSTRACIÓN 62 - INSTALACIÓN PASO 3.2 - 2..... | 111 |
| ILUSTRACIÓN 63 - INSTALACIÓN PASO 3.3 | 111 |
| ILUSTRACIÓN 64 - INSTALACIÓN PASO 3.4 | 112 |
| ILUSTRACIÓN 65 - INSTALACIÓN PASO 3.5 | 112 |
| ILUSTRACIÓN 66 - INSTALACIÓN PASO 3.6 | 113 |
| ILUSTRACIÓN 67 - INSTALACIÓN PASO 4.1 | 113 |
| ILUSTRACIÓN 68 - INSTALACIÓN PASO 4.2 | 113 |
| ILUSTRACIÓN 69 - INSTALACIÓN PASO 4.3 | 114 |

Índice de tablas

| | |
|--|----|
| TABLA 1 - PRESUPUESTO DEL PROYECTO | 22 |
| TABLA 2 - R-001 | 22 |
| TABLA 3 - R-003 | 23 |
| TABLA 4 - R-003 | 23 |
| TABLA 5 - R-004 | 24 |
| TABLA 6 - R-005 | 24 |
| TABLA 7 - R-006 | 24 |
| TABLA 8 - R-007 | 25 |
| TABLA 9 - R-008 | 25 |
| TABLA 10 - R-009 | 25 |
| TABLA 11 - REQUISITOS FUNCIONALES DEL SISTEMA | 40 |
| TABLA 12 - REQUISITOS NO FUNCIONALES DEL SISTEMA | 41 |
| TABLA 13 - CU-001 | 44 |
| TABLA 14 - CU-002 | 45 |
| TABLA 15 - CU-003 | 45 |
| TABLA 16 - CU-004 | 46 |
| TABLA 17 - CU-005 | 46 |
| TABLA 18 - CU-006 | 47 |
| TABLA 19 - CU-007 | 47 |
| TABLA 20 - CU-008 | 48 |
| TABLA 21 - CU-009 | 48 |
| TABLA 22 - CU-010 | 49 |
| TABLA 23 - CU-011 | 49 |
| TABLA 24 - CU-012 | 50 |
| TABLA 25 - CU-013 | 50 |
| TABLA 26 - CU-014 | 51 |
| TABLA 27 - CU-015 | 51 |
| TABLA 28 - CP001 | 85 |
| TABLA 29 - CP002 | 85 |
| TABLA 30 - CP003 | 86 |
| TABLA 31 - CP004 | 86 |
| TABLA 32 - CP005 | 86 |
| TABLA 33 - CP006 | 87 |
| TABLA 34 - CP007 | 87 |
| TABLA 35 - CP008 | 87 |
| TABLA 36 - CP009 | 88 |
| TABLA 37 - CP010 | 88 |
| TABLA 38 - CP011 | 88 |
| TABLA 39 - CP012 | 88 |
| TABLA 40 - CP013 | 89 |
| TABLA 41 - CP014 | 89 |
| TABLA 42 - CP015 | 89 |
| TABLA 43 - CP016 | 89 |
| TABLA 44 - CP017 | 90 |
| TABLA 45 - CP018 | 90 |
| TABLA 46 - CP019 | 90 |
| TABLA 47 - CP020 | 90 |
| TABLA 48 - CP021 | 91 |
| TABLA 49 - CP022 | 91 |
| TABLA 50 - CP023 | 91 |
| TABLA 51 - CP024 | 91 |
| TABLA 52 - CP025 | 91 |
| TABLA 53 - CP026 | 92 |
| TABLA 54 - CP027 | 92 |
| TABLA 55 - CP028 | 92 |
| TABLA 56 - CP029 | 92 |
| TABLA 57 - CP030 | 93 |

| | |
|-------------------------|----|
| TABLA 58 - CP031 | 93 |
| TABLA 59 - CP032 | 93 |
| TABLA 60 - CP033 | 93 |
| TABLA 61 - CP034 | 93 |
| TABLA 62 - CP035 | 94 |
| TABLA 63 - CP036 | 94 |
| TABLA 64 - CP037 | 94 |
| TABLA 65 - CP038 | 94 |
| TABLA 66 - CP039 | 94 |
| TABLA 67 - CP040 | 95 |
| TABLA 68 - CP041 | 95 |
| TABLA 69 - CP042 | 95 |
| TABLA 70 - CP043 | 95 |
| TABLA 71 - CP044 | 95 |
| TABLA 72 - CP045 | 96 |
| TABLA 73 - CP- 046..... | 96 |

Capítulo 1 - Introducción

1.1 Motivación

“De la Emoción al Sentido” [1] es un proyecto docente de la teoría de tópicos que comienza un equipo de profesores y antiguos alumnos del grado de Historia y Ciencias de la Música de la Universidad de Valladolid para ayudar al alumnado de dicho grado en la asignatura Música y pensamiento en el siglo de las luces (siglo XVIII).

Así pues, ¿qué se entiende por tópicos musicales?

Los tópicos musicales son temas del discurso musical que transmiten significados y emociones reconocibles de nuestra cultura, y que comparten los mismos códigos de ritmo, dinámica, tempo, armonía, timbre, ... (significante) en las obras de los compositores de Música Clásica occidental desde siglo XVIII [2].

Por tanto, este trabajo pretende analizar, comprender y desarrollar una aplicación para la recopilación de información de los distintos tópicos musicales y así mejorar la comprensión y el estudio de dicha información por parte del alumnado. Desde este punto de vista:

- Se pretende añadir capacidad de gestión y mantenimiento de una red de conocimientos que permita la inserción de elementos y de información adicional como bibliografía y referencias musicales por parte de colaboradores.
- Se desea que la red de conocimientos sea alimentada por un grupo selecto de colaboradores y moderadores.
- Finalmente se pretende que esta red de conocimiento pueda ser accedida para su consulta por alumnos e investigadores.

Sin embargo, ¿qué es una red de conocimiento?, ¿qué es un grafo de conocimiento?

Se entiende por red de conocimiento a un grupo de gente, entidades u organizaciones que captan, comparten conocimientos existentes y/o crean nuevos. Puede hacerse, por ejemplo, recopilando los documentos pertinentes y resumirlos para extraer conocimientos nuevos; transfiriendo conocimiento implícito en nuevo conocimiento implícito; mediante la narración de historias o el uso de metáforas; o codificando el conocimiento en documentos para distribuirlos electrónicamente. Está influenciada por su entorno, como el sistema de gestión en el que actúa y la cultura que la rodea; y por su procesamiento del conocimiento como captar, compartir y crear. Está respaldada por herramientas de información y comunicación, así como por tiempo y salas de reuniones.

Un grafo de conocimiento es la representación gráfica del razonamiento del conocimiento; se trata del conocimiento base que utiliza un modelo o topología de datos estructurados en grafos, conjunto de puntos unidos por líneas (aristas o arcos) para representar y operar con datos. Se utiliza a menudo para almacenar descripciones interconectadas de entidades (objetos, eventos, situaciones o conceptos abstractos) y al mismo tiempo codifican la semántica de forma libre o las relaciones subyacentes a estas entidades.

En consecuencia, este Trabajo de Fin de Grado (TFG en adelante) es la propuesta elaborada de un sistema para la creación y gestión de una red de conocimiento de tópicos musicales para el alumnado y el equipo del proyecto docentes.

1.2 Objetivos

El objetivo principal que pretende conseguir este TFG es la realización de un sistema para la creación y gestión de una red de conocimiento de tópicos musicales para la centralización de información y el apoyo al alumnado, presentando la información en estructuras llamadas tripletas RDF.

Otro objetivo de este TFG es la elaboración de un trabajo en el que se apliquen e integren los conocimientos, habilidades y actitudes desarrolladas en las distintas asignaturas del Grado en Ingeniería Informática mención Tecnologías de la Información de la Escuela de Ingeniería Informática de la Universidad de Valladolid.

1.3 Estructura de la memoria

Esta memoria se divide en los siguientes puntos:

- Introducción: en esta sección se presenta la motivación del proyecto y los objetivos de este.
- Planificación del proyecto: en esta sección se realiza una descripción de la metodología elegida, el alcance, el plan del proyecto y su presupuesto estimado.
- Contexto y antecedentes del proyecto: en esta sección se introducen varios conocimientos necesarios para comprender el proyecto.
- Análisis del sistema: en esta sección se detalla las funcionalidades exactas que pretende proporcionar la aplicación, así como sus requisitos y casos de uso.
- Arquitectura y diseño: en esta sección se describen los aspectos más relevantes de la arquitectura y el diseño.
- Implementación: en esta sección se definen los aspectos correspondientes a la fase de construcción de la ingeniería de software seguidas en las fases anteriores.
- Pruebas: en esta sección se exponen las pruebas realizadas para la comprobación de los diferentes casos de uso.
- Conclusiones y líneas futuras: en esta sección se comentan las conclusiones obtenidas a lo largo del desarrollo del proyecto y las líneas futuras de trabajo.
- Bibliografía: en esta sección se muestran las fuentes consultadas a lo largo de la realización de proyecto.
- Glosario: en esta sección se muestra un listado de palabras utilizadas en el documento.
- Anexo: en esta sección se muestra el manual de usuario y la instalación para el despliegue de la aplicación.

1.4 Agradecimientos

En primer lugar, quiero dar las gracias a mi familia por estar y ayudarme siempre en mi camino, sin vosotros no habría llegado hasta aquí.

En segundo lugar, quiero agradecer a todos los profesores que han dedicado su tiempo a mi formación y a la del resto de mis compañeros, y en especial agradecer a mi tutor, D. César Llamas Bello, por toda su ayuda a lo largo del Grado y por aceptar tutorizarme y guiarme en este Trabajo Fin de Grado.

Por último, gracias a mis amigas y amigos por haber estado a mi lado durante todo este tiempo.

Capítulo 2 - Planificación del proyecto

2.1 Metodología de trabajo

La metodología elegida para la planificación del proyecto software es la metodología en cascada

El desarrollo en cascada, también conocida por su nombre en inglés *Waterfall model*, es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente.

El modelo en cascada de cinco niveles (ilustración 1), basado en las propuestas de Winston W. Royce, es el más habitual y divide los procesos de desarrollo en las siguientes fases de proyecto [3]:

- Análisis: planificación, análisis y especificación de los requisitos.
- Diseño: diseño y especificación del sistema.
- Implementación: programación y pruebas unitarias.
- Verificación: integración de sistemas, pruebas de sistema y de integración.
- Mantenimiento: entrega, mantenimiento y mejora.

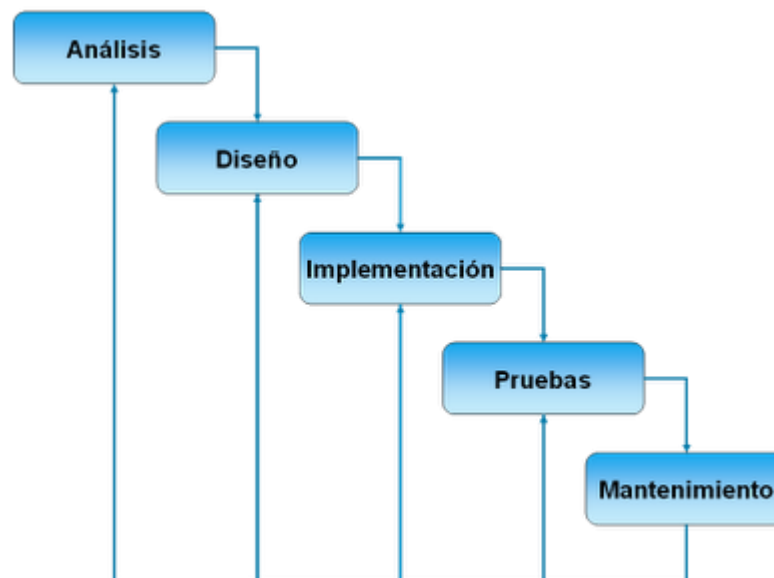


Ilustración 1 - Modelo en cascada de cinco niveles

Aunque es una metodología tradicional, ineficiente para algunos y fuera de moda, se ha elegido porque los requisitos son fijos, están bien documentados y son claros. Además, el proyecto no es de larga duración y no se requiere estrictamente la presencia del cliente después de la fase de requisitos [4].

2.2 Alcance

Para descomponer el alcance del proyecto se han empleado herramientas de gestión de proyectos. En concreto, la estructura de descomposición del trabajo y la estructura de descomposición del producto y entregables [5].

2.2.1 Estructura de Descomposición del Trabajo (actividades) – WBS

El enfoque basado en actividades consiste en crear una lista de todas las actividades que se cree que implica el proyecto. Esto puede requerir una sesión de lluvia de ideas que involucre a todo el equipo del proyecto o puede derivarse de un análisis de proyectos anteriores similares, pero se corre el riesgo de omitir o contar dos veces las tareas. Para evitarlos, existe la estructura de descomposición del trabajo (EDT), también conocida por su nombre en inglés *Work Breakdown Structure* (WBS), que organiza de forma jerárquica el trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de este y crear los entregables requeridos.[6]

En la parte superior de la estructura se encuentra el nivel más general, el cual representa las tareas principales, y por debajo los niveles inferiores que representan una definición con un detalle incrementado del trabajo del proyecto.

Los beneficios de utilizar WBS en la gestión de proyectos son diversos:

- Mejor gestión del alcance. Todas las actividades necesarias para completar el proyecto están identificadas y organizadas.
- Asignación eficiente de recursos y tiempo. Permite la identificación de las actividades críticas y la planificación de los hitos clave del proyecto.
- Ayuda en el establecimiento de una secuencia lógica de las actividades y en la identificación de las dependencias entre ellas.
- Facilita la comunicación y el seguimiento del progreso del proyecto. Proporciona una estructura clara y comprensible para el equipo del proyecto.

Seguidamente, se muestra el esquema WBS (ilustración 2) creado para este proyecto donde el entregable final es el proyecto, en el siguiente nivel las principales fases del proyecto y así descendientemente todas las tareas necesarias para completarlo.

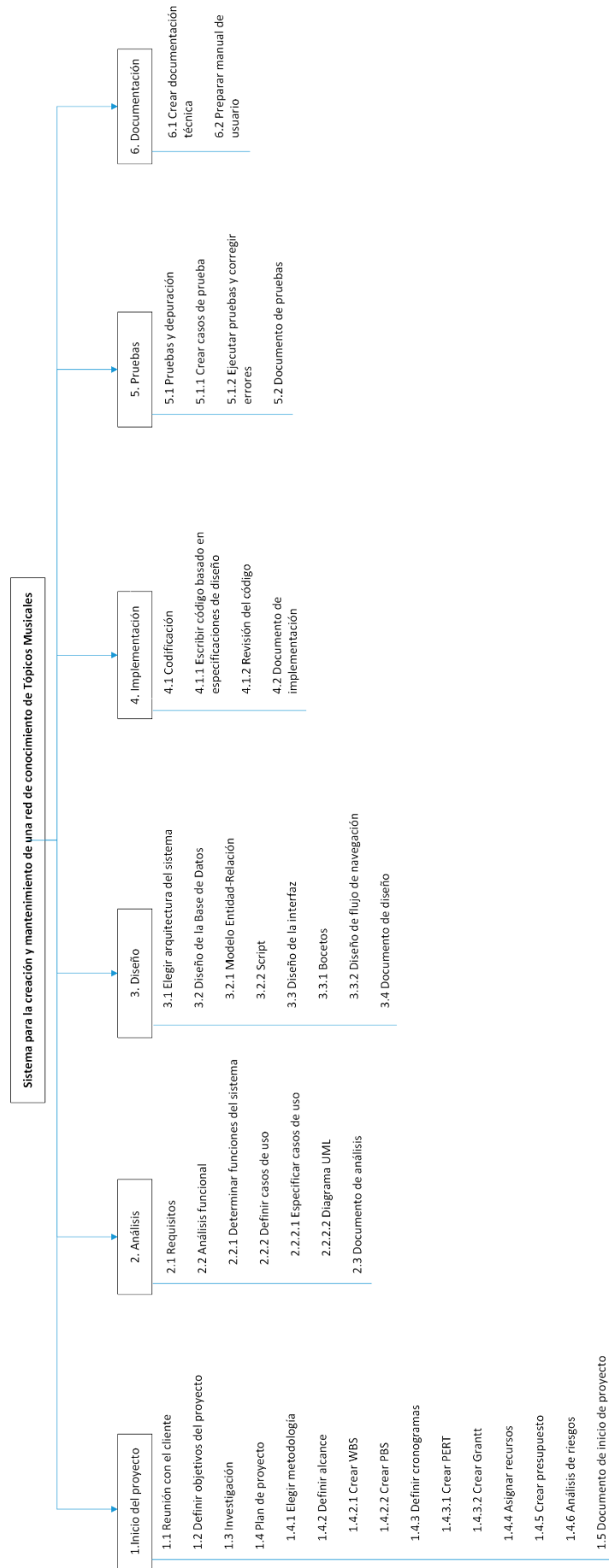


Ilustración 2 - Estructura de Descomposición del Trabajo (actividades) del proyecto

2.2.2 Estructura de Descomposición del Producto y Entregables – PBS

La planificación basada en productos [7] es una parte fundamental del enfoque PRINCE2 para la gestión de proyectos y consiste en identificar todos los productos (entregables del proyecto) que conforman o contribuyen a la consecución de los objetivos del proyecto, y el trabajo asociado necesario para alcanzarlos [8]. Para garantizar que se identifiquen y capturen todos los productos necesarios, existe la estructura de desglose de producto (EDP), también conocida por su nombre en inglés *Product Breakdown Structure* o PBS que organiza de forma jerárquica los entregables del proyecto.

Como el WBS, los niveles superiores representan lo principal y los niveles inferiores representan mayor detalle.

Los beneficios de utilizar PBS en la gestión de proyectos son diversos [9]:

- Mejora la asignación de recursos. Ayuda a gestionar el proyecto de forma más eficiente, facilitando la identificación de los recursos y el tiempo que puede requerir cada trabajo.
- Apoya la garantía de calidad. Permite identificar con mayor facilidad posibles fallos en el sistema.
- Mejora la comunicación entre los miembros del equipo del proyecto. Permite una comunicación más eficaz entre el equipo.
- Facilita el proceso de planificación de proyectos.

Seguidamente, se muestra el esquema PBS (ilustración 3) creado para este proyecto donde el entregable final es el proyecto.

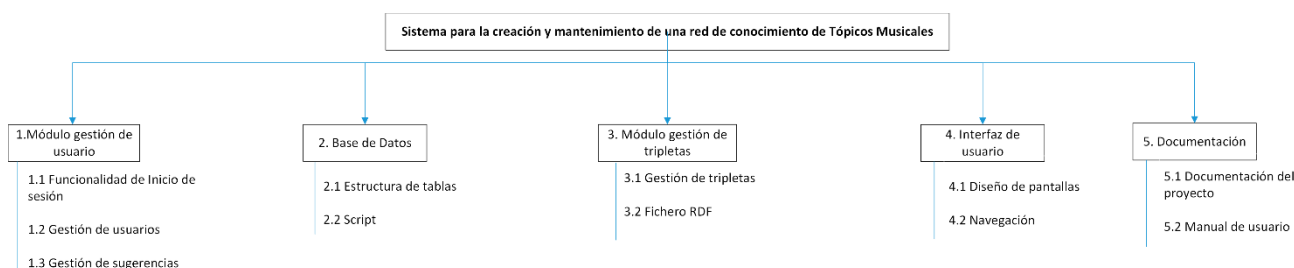


Ilustración 3 - Estructura de Descomposición del Producto y Entregables del proyecto

2.3 Cronogramas

En este apartado se muestran dos cronogramas creados para detallar la programación del proyecto.

2.3.1 Diagrama PERT

El diagrama PERT (*Program Evaluation and Review Techniques*) es una herramienta que sirve para visualizar las actividades relacionadas con un proyecto, cuál es su duración y cómo se vinculan unos con otros [10]. Son una forma habitual de reflejar lo que en otras aproximaciones se denomina red de actividades, aunque incluyen más información de tipo temporal y recursos. Las actividades del proyecto se representan mediante nodos y las dependencias entre actividades se representan mediante flechas que conectan nodos.

Los beneficios de utilizar el diagrama PERT en la gestión de proyectos son diversos:

- Determinación de la ruta crítica.
- Evaluación de los recursos. Muestra fácilmente qué tareas requieren recursos y cuáles no.
- Estimación del tiempo. El diagrama de PERT evalúa tanto la duración de cada tarea como la duración general del proyecto.

La siguiente imagen muestra el diagrama PERT del proyecto (ilustración 4). En rojo se resalta la ruta crítica [11].

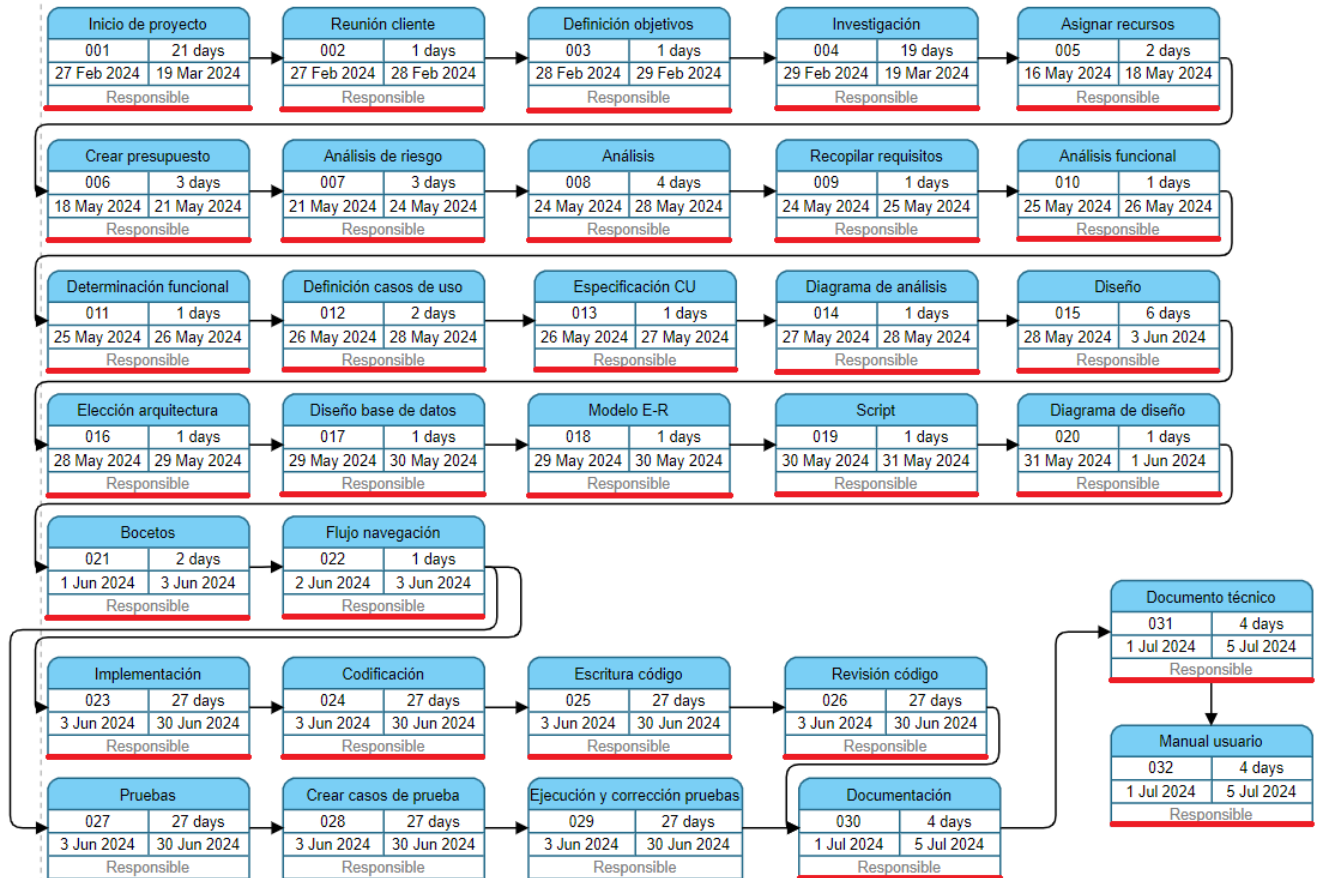


Ilustración 4 - Diagrama Program Evaluation and Review Techniques del proyecto

2.3.2 Diagrama Gantt

El diagrama de Gantt es un instrumento de gestión de proyectos que sirve para esquematizar las tareas que se tienen que realizar desde el inicio del proyecto hasta su conclusión. Consta de dos elementos: las tareas y el calendario. Cada tarea abarca una cantidad de días, por lo cual se forman barras, dependiendo la duración [12].

Los beneficios de utilizar el diagrama Gantt en la gestión de proyectos son diversos [13]:

- Proporciona una descripción general de alto nivel. Ofrece una perspectiva general de un proyecto y su cronograma. Proporciona información crítica sobre las tareas de un vistazo, incluido el orden en que comienzan y su duración.
- Mejora la eficiencia y ayuda a gestionar los recursos. Ayuda a garantizar la gestión del tiempo y recursos de manera efectiva.
- Permite un mejor seguimiento. Permite un seguimiento cercano de las tareas, los puntos de referencia y el flujo de trabajo general.
- Ilustra las superposiciones y dependencias.

- Aumenta la productividad. La visibilidad de alto nivel de un diagrama de Gantt ayuda al equipo a centrarse en las tareas que deben realizar.

A continuación, se muestra el diagrama Gantt (ilustración 5) final realizado con la herramienta Microsoft Project. El lapso de tiempo se debe a la materialización de los riesgos.

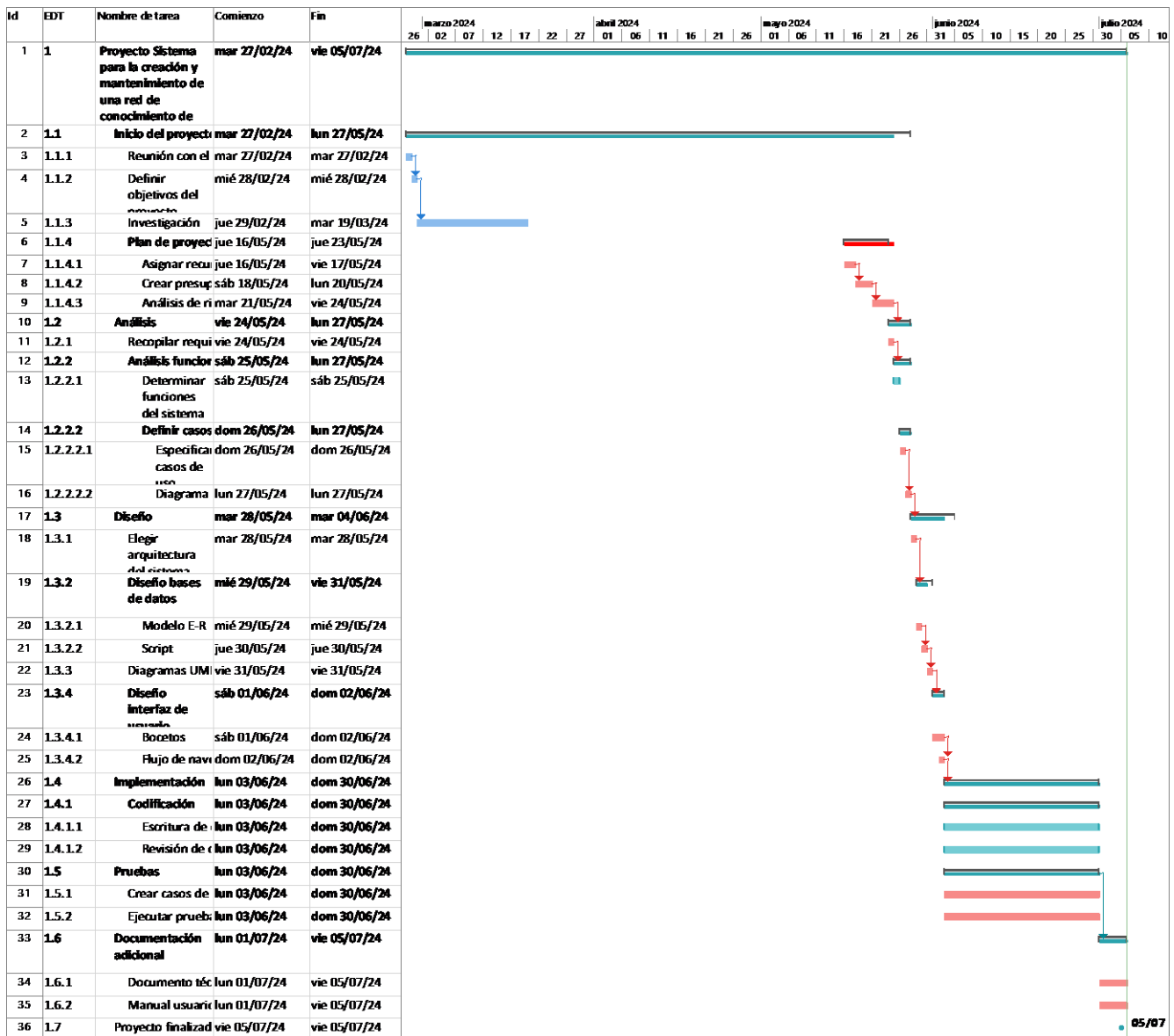


Ilustración 5 - Diagrama Gantt del proyecto

2.4 Asignación de recursos

Los recursos disponibles para el proyecto se han clasificados en dos tipos:

- Recursos humanos. Al tratarse de un TFG, sólo pueden existir los siguientes dos recursos:
 - La autora del TFG. Es la encargada de realizar el proyecto.
 - El tutor del TFG. Es el encargado de supervisar y guiar el proyecto.

- Recursos materiales. Los recursos materiales para el desarrollo del proyecto han sido:
 - Licencias software (especificadas en el apartado 5.2).
 - Ordenador personal.
 - Espacio de trabajo con mobiliario de oficina (mesa, silla y otros elementos para el confort y eficiencia).
 - Disco duro externo. Dispositivo usado para almacenar las copias de seguridad del proyecto.

2.5 Presupuesto

En esta sección se desglosa el plan presupuestario del proyecto en los distintos costes involucrados como el hardware, el software, los recursos humanos. Existe también otros costos como el de la electricidad y el de infraestructura, pero no se han incluido en el presupuesto. Los costos se han extrapolado a lo largo del tiempo total del proyecto, que en este caso ha abarcado un período de casi 2 meses, desde mediados de mayo hasta principios de julio de 2024.

Coste de hardware

El coste hardware contiene el valor de los diversos dispositivos electrónicos y material de oficina empleados en el proyecto. Estos han sido los siguientes:

- Ordenador portátil Hp Pavilion (16GB de RAM y procesador i7) tiene actualmente un coste de 996 euros. Si consideramos una vida útil de 4 años (48 meses), y como el proyecto abarca 2 meses, el coste contabilizado es de $996 / 48 * 2 = 41,5$ euros.
- Memoria USB de 16 GB tiene actualmente un coste que ronda los 11 euros y una vida media de 20 años (240 meses). Por lo que, el coste contabilizado es de $11 / 240 * 2 = 0,09$ euros.

Por tanto, el coste total del hardware ha salido a 41,59 euros.

Coste de software

En este proyecto, se han utilizado herramientas de software libre y de software comercial. Al ser estudiante de la Uva, ésta ha proporcionado las licencias académicas necesarias para el software comercial. Por lo tanto, no se ha contabilizado ningún coste referente al software. Sin embargo, en un entorno empresarial sí existiría tal coste. A continuación, se nombran las herramientas de software y su coste:

- Paquete Office 365 – Word, Excel, Microsoft Visio, Microsoft Project: tiene un coste mensual de 35.7 euros. Si se considera la duración del proyecto que es de 2 meses, el coste contabilizado es de $35.7 * 2 = 71.4$ euros. [14]
- Visual Paradigm 16.2: tiene un coste mensual de 35 dólares para obtener la versión profesional del programa que permite realizar los distintos diagramas empleados. Esta suscripción es posible al realizar una facturación cada 3 meses y cobrando dicho precio para cada mes. Teniendo en cuenta el cambio de divisa a euros con el cambio de 1 euro = 1.08 dólares, el coste mensual es 32.41 euros. Finalmente se obtiene un coste de $32.41 * 3 = 97.23$ euros. [15]

Por lo tanto, el coste total del software ha sido de 168.63 euros.

Coste de recursos humanos

Al tratarse el proyecto de un TFG, no se ha contabilizado ningún coste de recursos humanos. Sin embargo, en un entorno empresarial sí existiría tal coste. Para la realización de este proyecto se ha establecido un plazo de 300 horas, como indica la guía docente de la asignatura TFG mención tecnologías de la información y dado que la autora del TFG es una programadora con perfil junior con menos de tres años de experiencia laboral, tiene un salario medio por hora de 14,62 euros en España [16]. Por tanto, el supuesto coste estimado sería de 4.386 euros.

Coste total

| Concepto | Coste |
|------------------|-------------------|
| Hardware | 41,59 € |
| Software | 168,63 € |
| Recursos Humanos | 4.386,00 € |
| Total | 4.596,22 € |

Tabla 1 - Presupuesto del proyecto

2.6 Análisis de riesgos

Seguidamente, se muestra el análisis de cada riesgo en forma de tabla donde aparecen los apartados:

- ID: identificador único del riesgo.
- Nombre: nombre del riesgo.
- Descripción: explicación del riesgo.
- Probabilidad: posibilidad de que ese riesgo se materialice.
- Impacto: consecuencias o efectos de la materialización del riesgo sobre el proyecto.
- Plan de mitigación: estrategia para minimizar los efectos de la materialización del riesgo sobre el proyecto antes de que ocurra.
- Plan de contingencia: conjunto de medidas y procedimientos diseñados para minimizar el impacto.

| ID | R-001 |
|----------------------|---|
| NOMBRE | Ausencia del autor/a |
| DESCRIPCIÓN | La ausencia temporal del autor/a debido a diferentes causas como enfermedad, lesión, asuntos personales o cansancio visual (debido a la cantidad de horas frente a una pantalla) harán que el proyecto se paralice y en consecuencia aumente la duración de este. |
| PROBABILIDAD | Media |
| IMPACTO | Alto |
| PLAN DE MITIGACIÓN | |
| PLAN DE CONTINGENCIA | |

Tabla 2 - R-001

| ID | R-002 |
|----------------------|--|
| NOMBRE | Fallo del equipo de trabajo del autor/a |
| DESCRIPCIÓN | El fallo del equipo de trabajo del autor debido al hardware (sobrecalentamiento de la batería, rotura de pantalla, etc) o al software (virus, fallos en los programas, etc). |
| PROBABILIDAD | Baja |
| IMPACTO | Baja |
| PLAN DE MITIGACIÓN | Realizar copias de seguridad semanales o incluso diarias de todos los documentos del proyecto y almacenarlas en un repositorio/nube o en otro dispositivo de almacenamiento distinto del que se utiliza habitualmente en el proyecto. Comprobar que esas copias de seguridad sean recuperables. |
| PLAN DE CONTINGENCIA | Recuperar la información de las copias de seguridad realizadas. |

Tabla 3 - R-003

| ID | R-003 |
|----------------------|--|
| NOMBRE | Pérdida del trabajo realizado |
| DESCRIPCIÓN | Pérdida del trabajo realizado hasta el momento debido al R-002 o por la pérdida del equipo personal del autor/a. |
| PROBABILIDAD | Baja |
| IMPACTO | Alto |
| PLAN DE MITIGACIÓN | Mismo plan de mitigación del R-002. Mantener el equipo de trabajo en zonas seguras y de confianza. |
| PLAN DE CONTINGENCIA | Recuperar la información de las copias de seguridad realizadas. |

Tabla 4 - R-003

| ID | | R-004 | |
|----------------------|--|--------------|--|
| NOMBRE | Falta de conocimiento del autor | | |
| DESCRIPCIÓN | El uso de nuevas herramientas para el proyecto implica una falta de familiaridad con el entorno de estas y en consecuencia hacer que el autor/a del proyecto realice acciones fatales. | | |
| PROBABILIDAD | Media | | |
| IMPACTO | Medio | | |
| PLAN DE MITIGACIÓN | Documentarse, por ejemplo consultando los manuales, antes y durante el uso de las herramientas. | | |
| PLAN DE CONTINGENCIA | Consultar manuales oficiales. Buscar otras herramientas más factibles para usar. Asesoría de expertos en el área. | | |

Tabla 5 - R-004

| ID | | R-005 | |
|----------------------|---|--------------|--|
| NOMBRE | Retraso en el proyecto | | |
| DESCRIPCIÓN | Mala planificación del proyecto o imprevistos hacen que la planificación del proyecto se retrase. | | |
| PROBABILIDAD | Media | | |
| IMPACTO | Alto | | |
| PLAN DE MITIGACIÓN | Disponer de un plan detallado y realista que cuente con imprevistos. | | |
| PLAN DE CONTINGENCIA | Aplicar el plan | | |

Tabla 6 - R-005

| ID | | R-006 | |
|----------------------|---|--------------|--|
| NOMBRE | Disponibilidad del equipo | | |
| DESCRIPCIÓN | Debido a compromisos personales o académicos de los miembros del equipo puede hacer que la comunicación sea difícil o imposible, lo que afectaría al proyecto. | | |
| PROBABILIDAD | Media | | |
| IMPACTO | Bajo | | |
| PLAN DE MITIGACIÓN | Usar herramientas de comunicación como correo electrónico o chat para no perder el contacto. Fomentar la comunicación clara y efectiva entre los miembros del equipo para minimizar malentendidos. | | |
| PLAN DE CONTINGENCIA | Establecer un plan de tiempos para evitar retrasos. | | |

Tabla 7 - R-006

| ID | R-007 |
|----------------------|--|
| NOMBRE | Generación incorrecta del proyecto |
| DESCRIPCIÓN | El trabajo desarrollado en el proyecto no cumple con las expectativas de los miembros del equipo, lo que puede llevar a un retraso del proyecto. |
| PROBABILIDAD | Media |
| IMPACTO | Medio |
| PLAN DE MITIGACIÓN | Realizar una documentación detallada de las tareas a realizar para evitar malentendidos. Aprobación en conjunto de las tareas a realizar. |
| PLAN DE CONTINGENCIA | |

Tabla 8 - R-007

| ID | R-008 |
|----------------------|--|
| NOMBRE | Error en la estimación de tiempos |
| DESCRIPCIÓN | La estimación de tiempos para las actividades planificadas resulta ser incorrecto, no ajustándose a la realidad provocando un retraso en el proyecto |
| PROBABILIDAD | Media |
| IMPACTO | Alto |
| PLAN DE MITIGACIÓN | Realizar un buen análisis de las actividades. |
| PLAN DE CONTINGENCIA | Enfocarse en los aspectos más importantes del proyecto y dejar otros más triviales. |

Tabla 9 - R-008

| ID | R-009 |
|----------------------|---|
| NOMBRE | Modificación de los requisitos del proyecto |
| DESCRIPCIÓN | Durante el desarrollo del proyecto, pueden surgir cambios en los requisitos especificados que afecten la planificación. |
| PROBABILIDAD | Baja |
| IMPACTO | Media |
| PLAN DE MITIGACIÓN | Definir y priorizar las funcionalidades para evitar cambios continuos en los requisitos del proyecto Obtener retroalimentación para validar el progreso. |
| PLAN DE CONTINGENCIA | Enfocarse en los requisitos que implican funcionalidades críticas |

Tabla 10 - R-009

Después de detallar posibles riesgos que pueden surgir durante el desarrollo del proyecto, se ha creado una matriz de riesgos basada en la ilustración 6. En la matriz de riesgos creada (ilustración 7) se define el nivel de los riesgos (identificados por su ID) considerando la categoría de probabilidad frente a la categoría de impacto. Se trata de un mecanismo sencillo para aumentar la visibilidad de los riesgos y ayudar a la toma de decisiones de gestión [17] [18].

| Likelihood | Harm severity | | | |
|------------|---------------|----------|-----------|--------------|
| | Minor | Marginal | Critical | Catastrophic |
| Certain | High | High | Very high | Very high |
| Likely | Medium | High | High | Very high |
| Possible | Low | Medium | High | Very high |
| Unlikely | Low | Medium | Medium | High |
| Rare | Low | Low | Medium | Medium |
| Eliminated | Eliminated | | | |

Ilustración 6 - Modelo matriz de riesgos

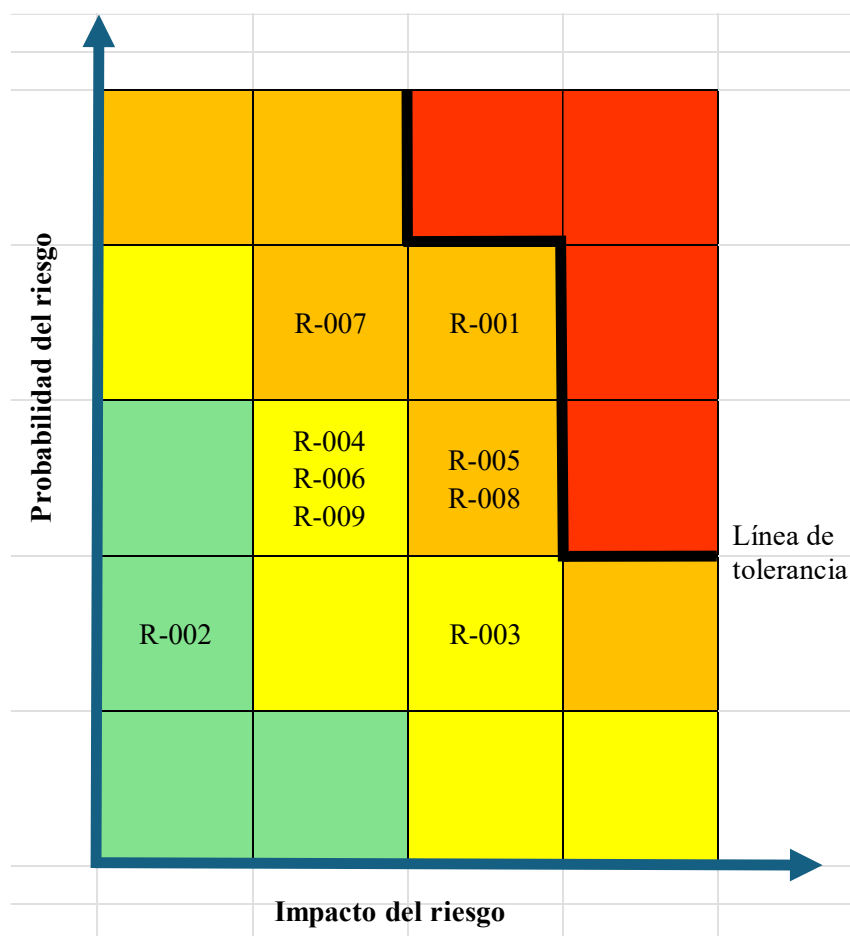


Ilustración 7 - Matriz de riesgos del proyecto

Capítulo 3 - Contexto y antecedentes del proyecto

3.1 Introducción

En esta sección se realiza una explicación sobre distintos conocimientos utilizados en el proyecto: el concepto de RDF, el lenguaje UML, tecnologías de bases de datos relacional y el lenguaje Java.

3.2 RDF

3.2.1 ¿Qué es RDF?

RDF son las siglas de *Resource Description Framework* (Marco de descripción de recursos) y es un modelo estándar desarrollado y estandarizado por el Consorcio *World Wide Web* (W3C) [19] para describir recursos web e intercambio de datos. Si bien existen muchas herramientas convencionales para manejar datos y, más específicamente, para manejar las relaciones entre datos, RDF es un estándar sencillo y poderoso [20].

El modelo de datos RDF es similar a los enfoques de modelado conceptual clásico como los diagramas entidad-relación o clases. Se basa en la idea de hacer declaraciones sobre recursos (cosas tanto concretas como abstractas) en expresiones conocidas como tripletas. Una tripleta semántica [21] (o tripleta RDF o simplemente tripleta) es, como su nombre indica, una estructura formada por sujeto-predicado-objeto (ilustración 8) [22]. El sujeto y el objeto (nodos o cajas) representan dos recursos relacionados y el predicado (arco o flecha) representa la naturaleza de esa relación formulada de manera direccional (del sujeto al objeto). Un objeto puede ser también un literal, definiendo una propiedad para un recurso.



Ilustración 8 - Tripleta semántica

Un ejemplo sencillo (ilustración 9) para representar la noción "La trompeta pertenece a la familia de los instrumentos de viento metal" como tripleta RDF sería:

- Sujeto que denota "trompeta"
- Predicado que denota "pertenece"
- Objeto que denota "instrumento de viento metal".



Ilustración 9 - Ejemplo de tripleta semántica

Para poder describir los recursos web (ya sean accesible o no a través de la red) RDF utiliza las URIs [23]. Un Identificador Uniforme de Recurso o URI (por sus siglas en inglés *Uniform Resource*

La investigación sobre los grafos de conocimiento en los últimos años ha ganado un gran interés. El artículo [27] realiza un estudio exhaustivo sobre los gráficos de conocimiento y enumera siete categorías importantes de investigaciones actuales sobre este tema (ilustración 11), destacando “Campos de aplicación” donde los gráficos de conocimiento tienen numerosas aplicaciones en varios campos, incluida la educación, la investigación científica, las redes sociales y la atención médica.

Ámbito educativo [27]

Muchos estudios se han centrado en implementar aplicaciones inteligentes para mejorar la calidad de la educación. Específicamente, en la era del big data, el procesamiento de datos se convierte en una tarea desafiante debido a la complejidad y falta de estructura de los datos educativos. Por tanto, los sistemas educativos inteligentes tienden a aplicar datos estructurados, como grafos de conocimiento. Varias aplicaciones (ilustración 12) basadas en grafos de conocimiento apoyan el proceso educativo, centrándose en particular en el procesamiento de datos y la difusión del conocimiento.

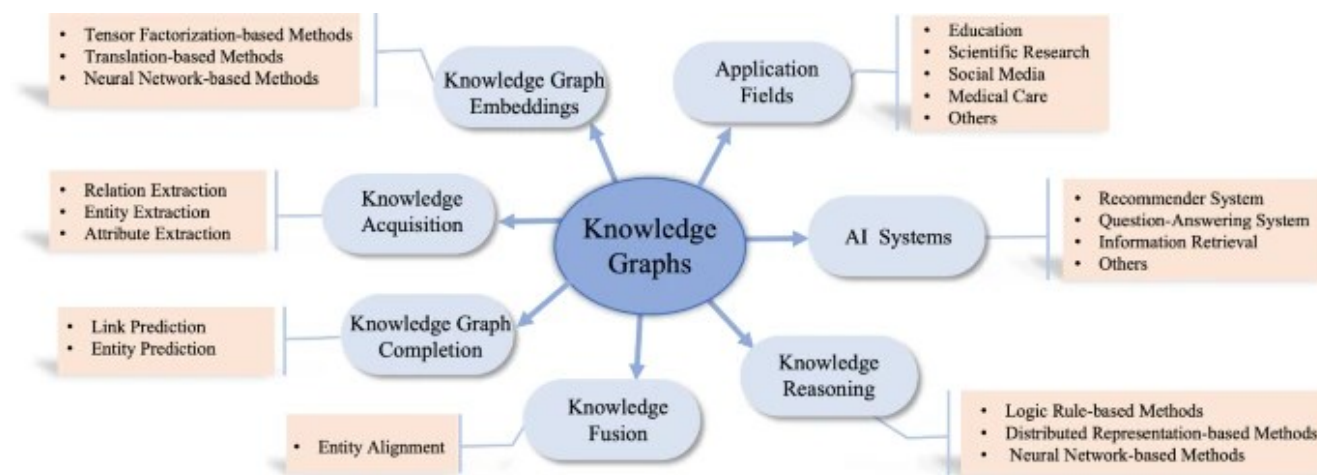


Ilustración 11 - Categorías importantes de investigaciones actuales de los grafos de conocimiento

| Fields | Applications | Methods | Functions |
|-----------|---|---|--|
| Education | Knowledge Graph based Course Management Model (Aliyu et al, 2020) KnowEdu (Chen et al, 2018) | Course knowledge graphs Instructional concepts extraction; Educational relation identification | Courses management; Generation of course allocation schedule Educational knowledge graph construction |

Ilustración 12 - Aplicaciones basadas en grafos de conocimiento en el ámbito educativo

3.2.3 Herramientas de visualización

En este apartado se muestra la búsqueda realizada sobre diferentes herramientas de visualización de gráficos RDF existentes a fecha de este documento. Las herramientas son las siguientes [29]:

- **Protégé:** es un editor de ontologías gratuito y de código abierto y un sistema de gestión del conocimiento. Escrita totalmente en Java. Su última versión es del 14 de marzo de 2019. [30] [31]

Protégé necesita de *plugings* para la visualización de las ontologías como NavigOWL, OWLViz, VOWL. Sin embargo, la gran mayoría de los encontrados se encuentran desactualizados o ya no están disponibles.

- **Online.**

- [:isSemantic \[32\]](#): es herramienta de visualización de gráficos RDF en línea. Es interactiva. Los formatos de serialización RDF admitidos son Turtle, RDF/XML, N-Triples, TriG, N-Quads, mixed, JSON-LD, microdata, RDFa.
- [RDF Grapher \[33\]](#): es un servicio web para analizar datos RDF y visualizarlos como un gráfico. El servicio está basado en Redland Raptor y Graphviz. Los formatos de serialización RDF admitidos son Turtle, RDF/XML, RDF/JSON, N-Triples, TriG y N-Quads. Los formatos de exportación admitidos son PNG, SVG, PDF, PS, EPS, GIF y JPG.
- [RDF Shape \[34\]](#): es una herramienta para la conversión, validación y visualización de datos RDF, entre otras características. Los formatos de serialización RDF admitidos son Turtle, RDF/XML, RDF/JSON, N-Triples, TriG, N-Quads, JSON-LD, mixed, html-rdfa11, html-microdata. El formato de exportación admitido es SVG.

Ejemplo

Se ha creado un ejemplo sencillo con las siguientes tripletas:

- Mario - esHermanoDe - Luigi
- Mario - trabajaComo - fontanero
- Mario - visteDe - rojo
- Luigi - trabajaComo - fontanero
- Luigi - visteDe - verde

Recursos (sujeto y objeto): Mario, Luigi.

Relación (predicado): esHermanoDe, trabajaComo, visteDe.

Literal (objeto): rojo, verde, fontanero.

El formato de serialización RDF utilizado ha sido Turtle [\[35\]](#) [\[36\]](#) por ser más sencillo de lectura y por ende de comprender. En la ilustración 13 se muestra el ejemplo sin y con el uso de prefijos.

```
#CÓDIGO 1
<http://example.org/Mario>
  <http://example.org/esHermanoDe>
    <http://example.org/Luigi> ;
  <http://example.org/trabajaComo>
    "fontanero" ;
  <http://example.org/visteDeColor>
    "rojo" .

<http://example.org/Luigi>
  <http://example.org/trabajaComo>
    "fontanero" ;
  <http://example.org/visteDeColor>
    "verde" .

#CÓDIGO 2
@prefix ex: <http://example.org/> .

ex:Mario ex:esHermanoDe ex:Luigi ; ex:trabajaComo "fontanero" ; ex:visteDeColor "rojo" .
ex:Luigi ex:trabajaComo "fontanero" ; ex:visteDeColor "verde" .
```

Ilustración 13 - Códigos Turtle del proyecto

A continuación se muestran las visualizaciones de los dos códigos usando las herramientas: isSemantic (ilustraciones 14-17), RDF Grapher (ilustraciones 18 y 19) y RDF Shape (ilustraciones 20 y 21).

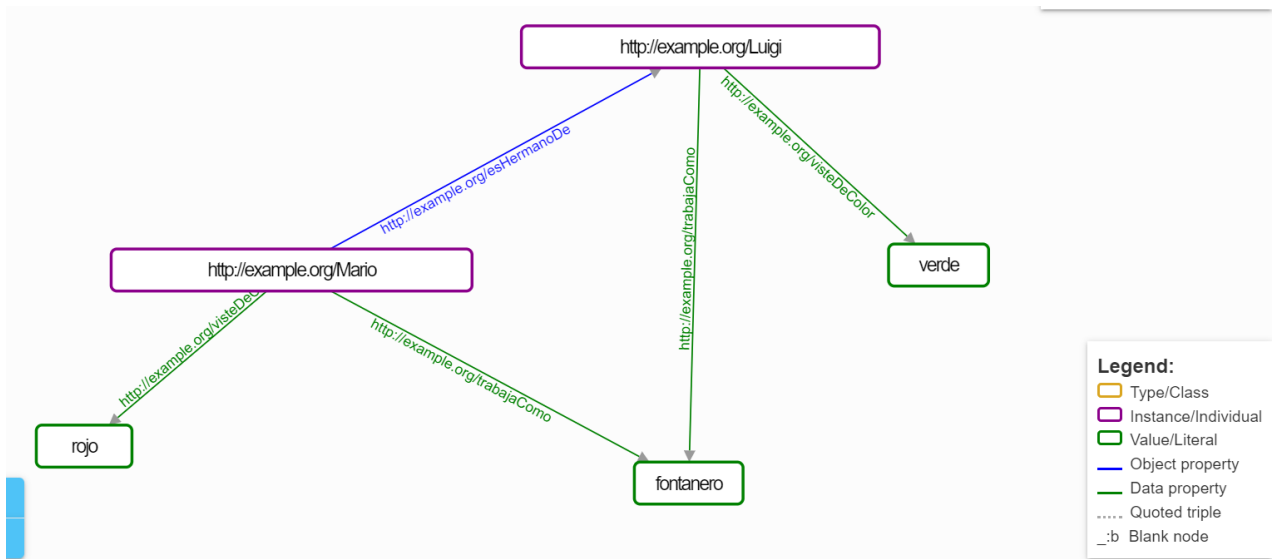


Ilustración 14 - Visualización del código 1 de Turtle con :isSemantic

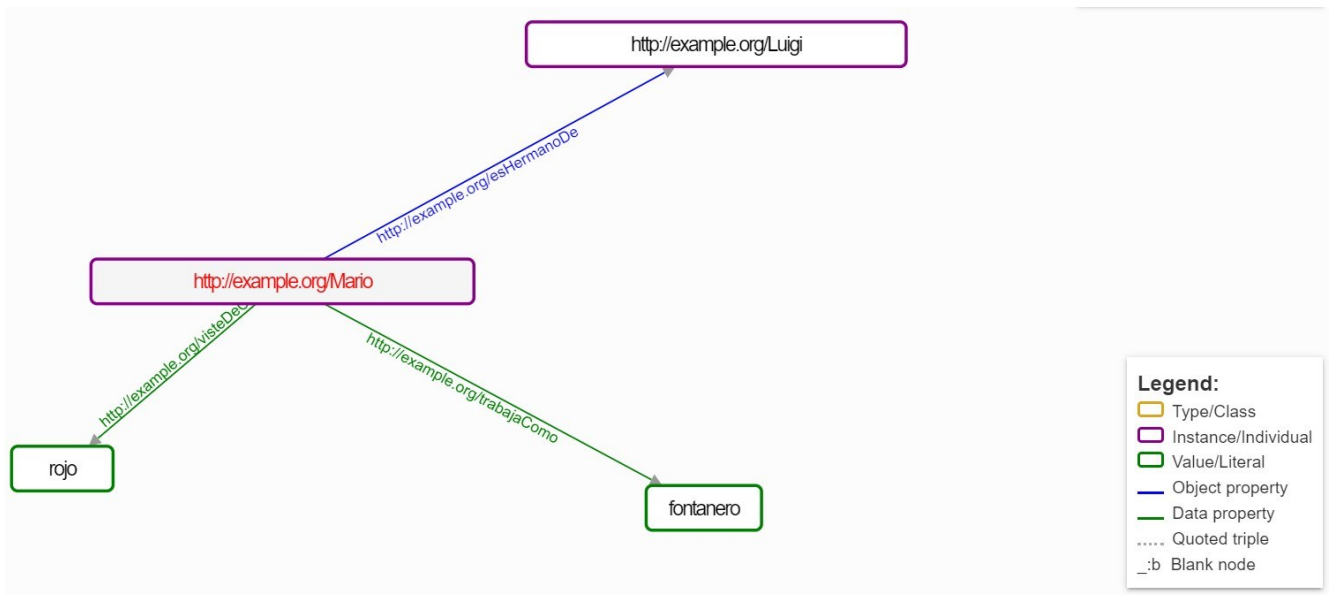


Ilustración 15 - Relaciones del recurso Mario del ejemplo de código Turtle

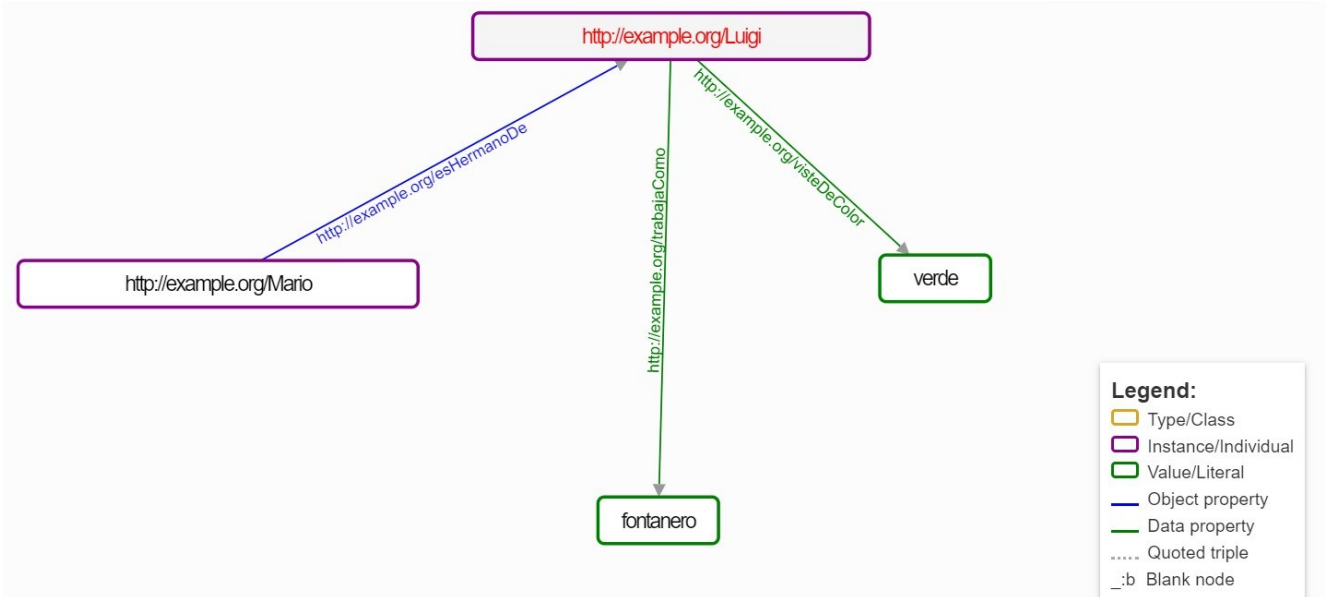


Ilustración 16 - Relaciones del recurso Luigi del ejemplo del código Turtle

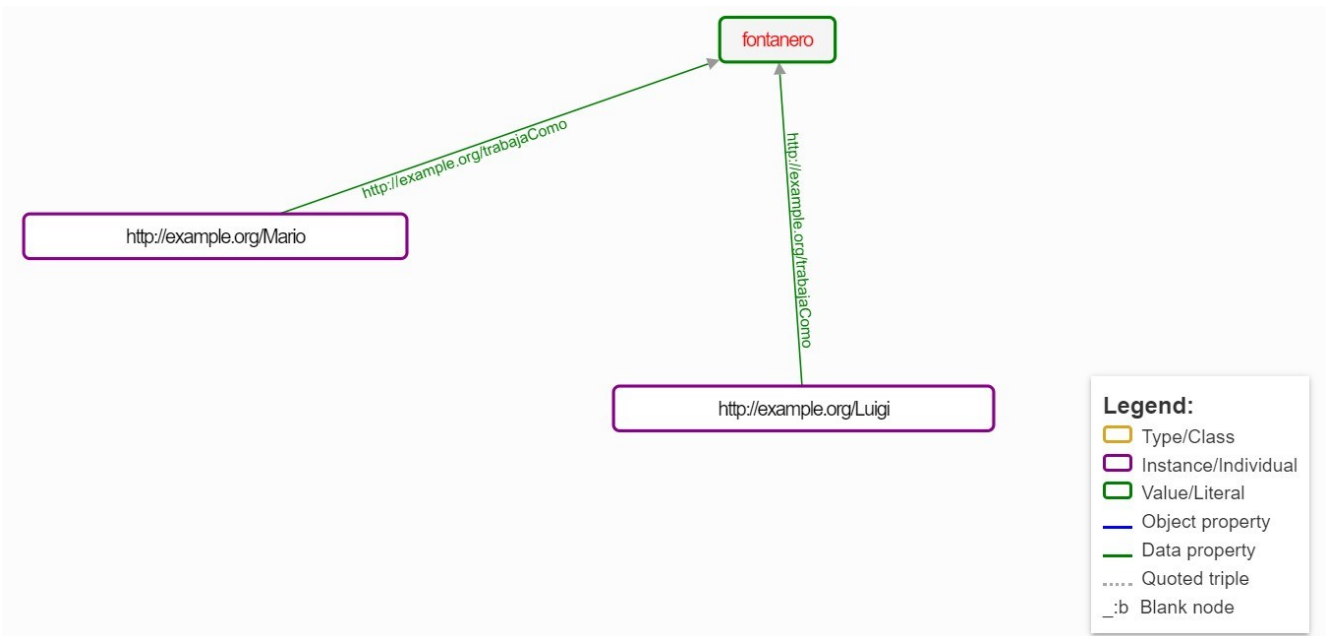


Ilustración 17 - Relaciones del literal "fontanero" del ejemplo del código Turtle

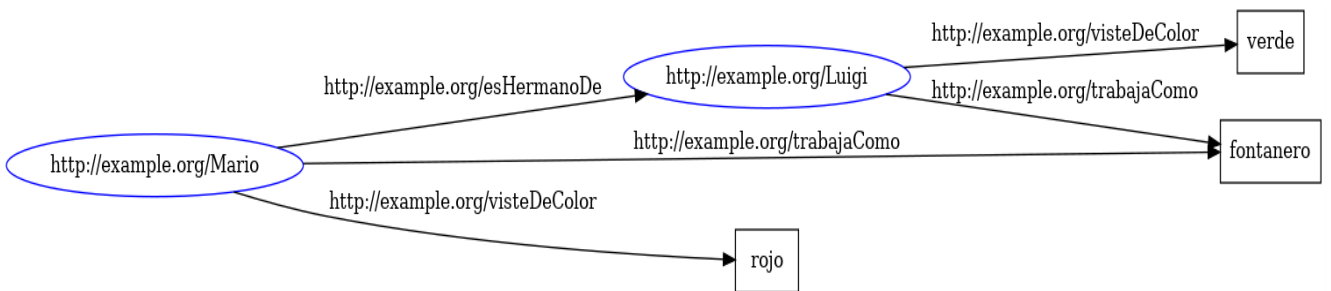
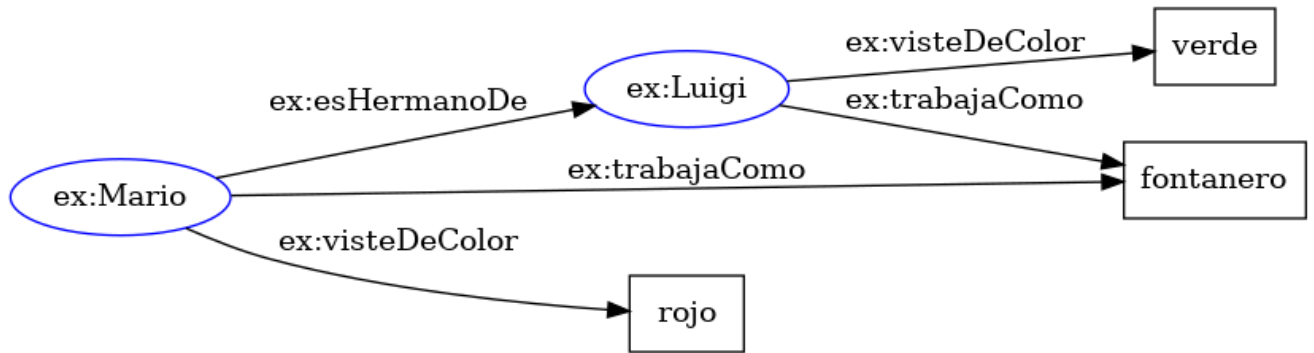


Ilustración 18 - Visualización del código 1 de Turtle con RDF Grapher



Namespaces:
 ex: <http://example.org/>

Ilustración 19 - Visualización del código 2 de Turtle con RDF Grapher

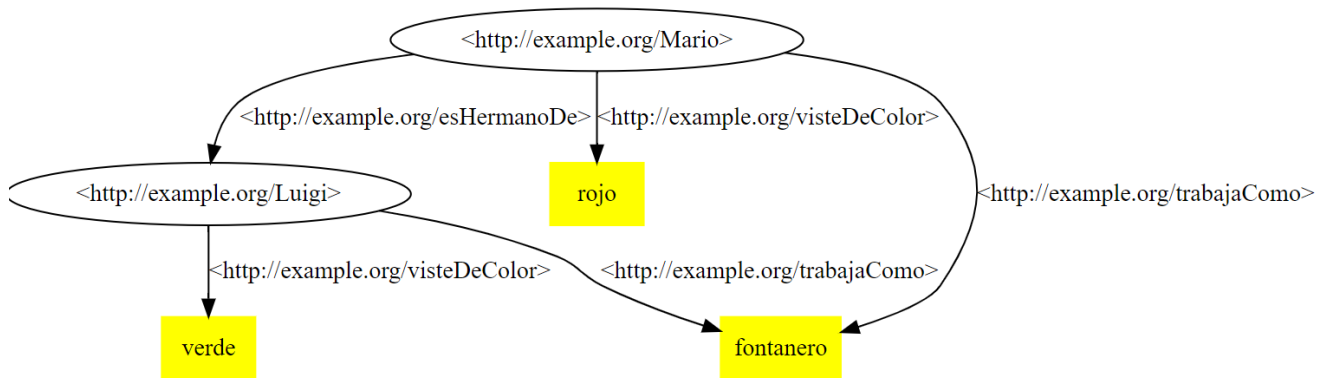


Ilustración 20 - Visualización del código 1 de Turtle con RDF Shape

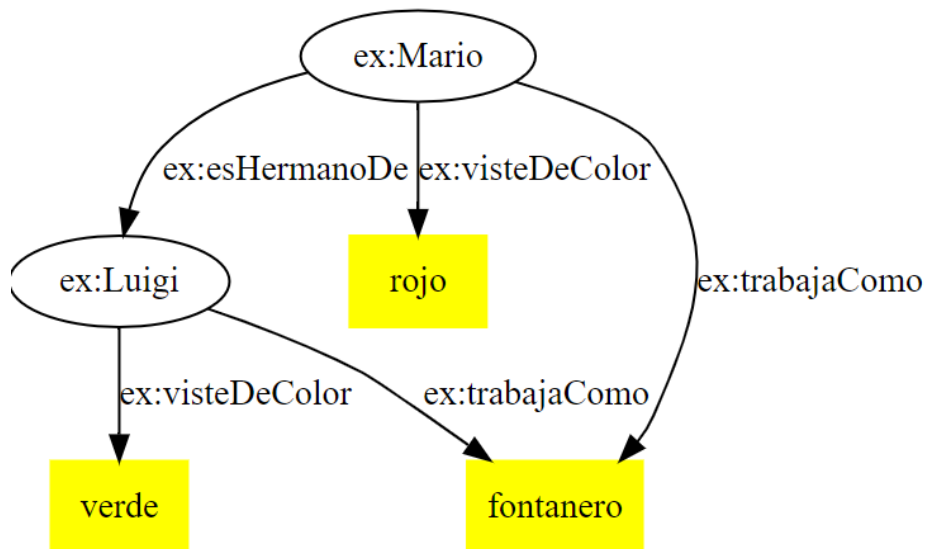


Ilustración 21 - Visualización del código 2 de Turtle con RDF Shape

3.2.3 Uso en el proyecto

Habiendo explicado las características de RDF y las posibles herramientas, se explicará su uso en el proyecto.

El propósito de utilizar RDF tendría varios motivos. El primero, sería facilitar la posibilidad de organizar, describir y compartir información de una manera estructurada y accesible. Otro motivo sería la ayuda a las personas a comprender la vinculación y organización de la información de una manera sencilla mediante la representación de tripletas RDF (sujeto-predicado-objeto) y la visualización de los grafos RDF.

3.3 UML

El lenguaje unificado de modelado o también conocido por su nombre en inglés *Unified Modeling Language* o UML es un estándar para la representación visual de objetos, estados y procesos dentro de un sistema. Por un lado, el lenguaje de modelado sirve de modelo para un proyecto y garantiza así una arquitectura de información estructurada; por otro lado, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. UML se utiliza principalmente en el desarrollo de software orientado a objetos [37] [38].

UML utiliza diagramas para representar los siguientes componentes del sistema:

- Objetos individuales (elementos básicos).
- Clases (combina elementos con las mismas propiedades).
- Relaciones entre objetos (jerarquía y comportamiento/comunicación entre objetos).
- Actividad (combinación compleja de acciones/módulos de comportamiento).
- Interacciones entre objetos e interfaces.

3.3.1 Diagramas UML

UML define 14 tipos de diagramas que se dividen en dos categorías: estructura y comportamiento [37].

- Diagramas de estructura

Los diagramas de estructura representan los elementos individuales de un sistema. Por lo tanto, son especialmente adecuados para la representación de la arquitectura de software.

Los diagramas pertenecientes a esta categoría son (se explican los utilizados en este proyecto):

- Diagrama de clases: muestra clases y objetos y sus interrelaciones.
- Diagrama de objetos.
- Diagrama de componentes.
- Diagrama de estructura compositiva.
- Diagrama de paquete.
- Diagrama de distribución.
- Gráfica de perfil.

- Diagramas de comportamiento

Los diagramas de comportamiento a diferencia de los diagramas estructurales no son estáticos, sino que representan procesos y situaciones dinámicas.

Los diagramas pertenecientes a esta categoría son (se explican los utilizados en este proyecto):

- Diagrama de casos de uso: muestra el comportamiento que se esperará de un sistema más adelante.
- Diagrama de actividades.
- Diagrama de máquina de estados: una máquina de estados, también llamada autómeta finito, representa un conjunto finito de estados en un sistema. Si se cumple una condición definida en el sistema (se detona un desencadenante), se produce una situación correspondiente. Esto puede incluir actividades o interacciones.
- Diagrama de interacción: modelar el comportamiento en el que los elementos intercambian información.
 - Diagrama de secuencia: representa el intercambio de mensajes entre objetos.
 - Diagrama de comunicación.
 - Diagrama de tiempos.

3.3.2 Estereotipos

Un estereotipo en UML Permiten extender la semántica (pero no la estructura) de modelado UML. Un elemento estereotipado puede tener restricciones más allá del elemento sin estereotipar, una imagen visual distinta o propiedades adicionales. Los estereotipos usados en el proyecto se muestran en la ilustración 22 [39].

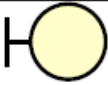


| Estereotipo | Icono | Semántica |
|-------------|---|--|
| «boundary» |  | Mediadora entre el sistema y su entorno. |
| «control» |  | Encapsula en comportamiento descrito en los casos de uso |
| «entity» |  | Modela información persistente (o no). |

Ilustración 22 - Clases estereotipadas de UML

3.5 Tecnologías de bases de datos

En este apartado, se hará una introducción a las bases de datos (BBDD) relaciones, al lenguaje de consulta de BBDD SQL (*Structured Query Language*) así como un motor de bases de datos basado en el lenguaje SQL como MySQL.

3.5.1 Base de datos relacional

Una base de datos relacional es una base de datos basada en el modelo relacional de datos. Un modelo relacional organiza los datos en una o más tablas de columnas y filas, con una clave única que identifica cada fila de una tabla. Generalmente, cada tabla representa un "tipo de entidad". Las filas representan instancias de ese tipo de entidad y las columnas representan valores atribuidos a esa instancia [40].

A la clave única se la conoce como clave primaria y se compone de un atributo o una combinación de atributos. También es importante el término de clave foránea, que es un conjunto de atributos en una tabla que hace referencia a la clave candidata de otra tabla, vinculando estas dos tablas.

Las ventajas de una base de datos relacional son:

- Sencillez. El modelo de datos que subyace a la base de datos relacional se implementa y gestiona más fácilmente que otros modelos.
- Escasa redundancia de datos. Las formas normales del modelo relacional fijan una normativa que tiene como fin evitar duplicaciones.
- Alta consistencia de datos. Las bases de datos relacionales normalizadas permiten almacenar datos sin contradicciones, contribuyendo así a la consistencia de los datos.
- Procesamiento de datos orientado a conjuntos. El sistema de base de datos relacional se apoya en un procesamiento orientado a conjuntos que subdivide cada entidad en valores mínimos.
- Lenguaje de consultas homogéneo. Para la realización de consultas a bases de datos relacionales se ha consolidado el lenguaje SQL, que ha sido estandarizado por la ISO y la IEC. El propósito de tal estandarización es que las aplicaciones puedan desarrollarse y ejecutarse con independencia del SGBD en que se utilicen.

3.5.2 Lenguaje SQL

Idealmente, un lenguaje de base de datos debe permitir al usuario [41]:

- Crear la base de datos y las estructuras de relación.
- Realizar tareas básicas de gestión de datos, como la inserción, modificación y borrado de los datos de las relaciones.
- Realizar consultas tanto simples como complejas.

Un lenguaje de base de datos debe realizar estas tareas requiriendo un esfuerzo mínimo por parte del usuario, y su sintaxis y la estructura de los comandos deben ser relativamente fáciles de aprender. Finalmente, el lenguaje debe ser portable, es decir, debe ajustarse a algún estándar reconocido, para poder utilizar la misma estructura de comandos y la misma sintaxis cuando pasemos de un SGBD a otro. SQL tiene como objetivo satisfacer estos requisitos.

SQL (*Structured Query Language*) es un ejemplo de lenguaje orientado a transformación, es decir, un lenguaje diseñado para usar relaciones con el fin de transformar los datos de entrada en las salidas requeridas. Como lenguaje, el estándar SQL de ISO tiene dos componentes principales:

- Un lenguaje de definición de datos (DDL, *Data Definition Language*) para definir la estructura de la base de datos y controlar el acceso a los datos.
- Un lenguaje de manipulación de datos (DML, *Data Manipulation Language*) para extraer y actualizar los datos.

3.5.3 MySQL

MySQL es un sistema gestor de bases de datos relacionales de código abierto. Un Sistema Gestor de Base de Datos (SGBD) es un paquete de software generalizado para implementar y mantener una base de datos informatizada. El propósito principal de un SGBD es proporcionar a los usuarios una visión abstracta de los datos (esconde detalles de almacenamiento y mantenimiento) [42].

Las principales funciones de un SGBD son:

- Gestionar los datos almacenados.
- Compilar las definiciones de datos.
- Compilar y ejecutar las consultas de los usuarios.
- Cargar ficheros de datos ya existentes en la base de datos.
- Copias de seguridad (*backup*).
- Reorganización de ficheros.
- Control del rendimiento.
- Control de autorización o integridad.
- Control de transacciones.

3.6 Lenguaje Java

Java es un lenguaje de programación con las siguientes características [43]:

1. Lenguaje Simple: se lo conoce como lenguaje simple porque viene de la misma estructura de otros lenguajes de programación como c y c++ .
2. Orientado a Objetos. La programación orientada a objetos (POO) es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen información en forma de campos (a veces también referidos como atributos, cualidades o propiedades) y código en forma de métodos. Los objetos son capaces de interactuar y modificar los valores contenidos en sus campos o atributos (estado) a través de sus métodos (comportamiento) [44].
3. Multihilos: tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos, ya que por cada hilo que el programa tenga se ejecutarán en tiempo real muchas funciones al mismo tiempo.
4. Alto Rendimiento: es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.

Capítulo 4 - Análisis del sistema

4.1 Introducción

En este capítulo se realiza el análisis del sistema a través del análisis de requisitos, definición de los casos de uso y la utilización de distintos diagramas de UML como el de clases, el de secuencia y el de máquina de estados.

4.2 Análisis de requisitos

El análisis de requisitos es el proceso de identificar, clasificar y documentar las necesidades y expectativas de los *stakeholders* para un determinado sistema, producto o proyecto. Se recopila información detallada sobre las funcionalidades, comportamientos y restricciones que debe cumplir el sistema, con el fin de establecer una base sólida para el diseño y desarrollo del mismo.

Seguidamente, se muestra el análisis de requisitos en forma de tabla donde aparecen los apartados:

- ID: identificador único del requisito.
- Nombre: nombre del requisito.
- Descripción: explicación del requisito.

| ID | NOMBRE | DESCRIPCIÓN |
|--------|--------------------|---|
| RF-001 | Iniciar sesión | El sistema debe permitir iniciar sesión. |
| RF-002 | Cerrar sesión | El sistema debe permitir cerrar sesión. |
| RF-003 | Crear usuario | El sistema debe permitir crear un usuario. |
| RF-004 | Mostrar usuarios | El sistema debe permitir ver todos los usuarios existentes. |
| RF-005 | Modificar usuario | El sistema debe permitir modificar el usuario deseado. |
| RF-006 | Eliminar usuario | El sistema debe permitir eliminar el usuario deseado. |
| RF-007 | Filtrar usuarios | El sistema debe permitir filtrar por rol los usuarios existentes. |
| RF-008 | Crear objeto | El sistema debe permitir crear un objeto. |
| RF-009 | Mostrar objetos | El sistema debe permitir ver todos los objetos existentes. |
| RF-010 | Modificar objeto | El sistema debe permitir modificar el objeto deseado. |
| RF-011 | Eliminar objeto | El sistema debe permitir eliminar el objeto deseado. |
| RF-012 | Crear predicado | El sistema debe permitir crear un predicado. |
| RF-013 | Mostrar predicados | El sistema debe permitir ver todos los predicados existentes. |

| | | |
|--------|------------------------|---|
| RF-014 | Modificar predicado | El sistema debe permitir modificar el predicado deseado. |
| RF-015 | Eliminar predicado | El sistema debe permitir eliminar el predicado deseado. |
| RF-016 | Crear tripleta | El sistema debe permitir crear una tripleta. |
| RF-017 | Mostrar tripleta | El sistema debe permitir ver los detalles de la tripleta deseada. |
| RF-018 | Modificar tripleta | El sistema debe permitir modificar la tripleta deseada. |
| RF-019 | Eliminar tripleta | El sistema debe permitir eliminar la tripleta deseada. |
| RF-020 | Filtrar tripletas | El sistema debe permitir filtrar por objeto o por predicado las tripletas existentes. |
| RF-021 | Validar tripleta | El sistema debe permitir validar la tripleta sin validar. |
| RF-022 | Crear obra | El sistema debe permitir crear una nueva obra agregada a una tripleta. |
| RF-023 | Mostrar obra | El sistema debe permitir ver los detalles de la obra de la tripleta deseada. |
| RF-024 | Modificar obra | El sistema debe permitir modificar la obra deseada. |
| RF-025 | Eliminar obra | El sistema debe permitir eliminar la obra deseada. |
| RF-026 | Crear bibliografía | El sistema debe permitir crear una bibliografía agregada a una tripleta. |
| RF-027 | Mostrar bibliografía | El sistema debe permitir ver los detalles de la bibliografía de la tripleta deseada. |
| RF-028 | Modificar bibliografía | El sistema debe permitir modificar la bibliografía deseada. |
| RF-029 | Eliminar bibliografía | El sistema debe permitir eliminar la bibliografía deseada. |
| RF-030 | Crear sugerencia | El sistema debe permitir crear una sugerencia. |
| RF-031 | Mostrar sugerencia | El sistema debe permitir ver todas las sugerencias existentes. |
| RF-032 | Eliminar sugerencia | El sistema debe permitir eliminar la sugerencia deseada. |
| RF-033 | Filtrar sugerencias | El sistema debe permitir filtrar las sugerencias existentes por el creador. |

Tabla 11 - Requisitos funcionales del sistema

| ID | NOMBRE | DESCRIPCIÓN |
|---------|--|--|
| RNF-001 | Sistema Gestor de Base de Datos Relacional | La base de datos del sistema estará gestionada por MySQL. |
| RNF-002 | Capacidad de la Base de Datos | La base de datos deberá ser capaz de almacenar y tratar correctamente la información correspondiente. |
| RNF-003 | Campos de entrada | La interfaz de introducción de parámetros debe ser clara y fácil de entender. |
| RNF-004 | Confiable | El sistema deberá ser confiable, asegurando la respuesta esperada. |
| RNF-005 | Lenguaje de Programación | El sistema será implementado con el lenguaje de programación Java. |
| RNF-006 | Usabilidad | El sistema deberá ser fácil de usar e intuitivo. |
| RNF-007 | Escalabilidad | El sistema deberá ser construido de acuerdo a ciertos patrones que permitan el aumento de sus funciones. |

Tabla 12 - Requisitos no funcionales del sistema

4.3 Casos de uso

Un caso de uso es una lista de acciones o pasos de eventos que generalmente definen las interacciones entre un rol (conocido en el Lenguaje de Modelado Unificado (UML) como actor) y un sistema para lograr un objetivo [45]. El actor puede ser un humano u otro sistema externo.

4.3.1 Actores

Los actores definidos en el proyecto son:

- Administrador: rol que se encarga de la gestión de los usuarios de la base de datos del sistema, es decir, de su creación, modificación y eliminación. También se encarga de la gestión de las sugerencias.
- Lector: rol que puede visualizar las tripletas validadas.
- Colaborador: rol que puede crear nuevas tripletas y sugerencias. También puede realizar lo mismo que el rol lector.
- Moderador: rol que se encarga de la validación y gestión de las tripletas y de la gestión de las sugerencias. También puede realizar lo mismo que el rol colaborador.

4.3.2 Diagrama de casos de uso

El diagrama de casos de uso muestra el comportamiento que se esperará de un sistema más adelante. Los elementos del diagrama son el actor, representado por la ilustración de hombre palo, el sistema, representado por un rectángulo, y el caso de uso, representado por una elipse. La relación entre estos elementos se representa con unas líneas de conexión [46].

En el diagrama (ilustración 23) se puede apreciar los cuatro actores del sistema, los cuales referencian a los roles que puede tener un usuario: administrador, moderador, colaborador y lector. Entre los actores existe una herencia: el actor moderador hereda los casos de uso del actor colaborador y este a su vez hereda los casos de uso del actor lector. Este diagrama permite apreciar visualmente la jerarquía entre los actores del sistema. Con relación a los casos de uso, la línea discontinua establece una relación entre diferentes casos de uso junto con un estereotipo [47]:

- *Include*: el caso de uso del que sale la flecha requiere que se realice el caso de uso indicado con la punta de la flecha.
- *Extend*: es un curso alternativo al caso de uso base (del que sale la flecha)

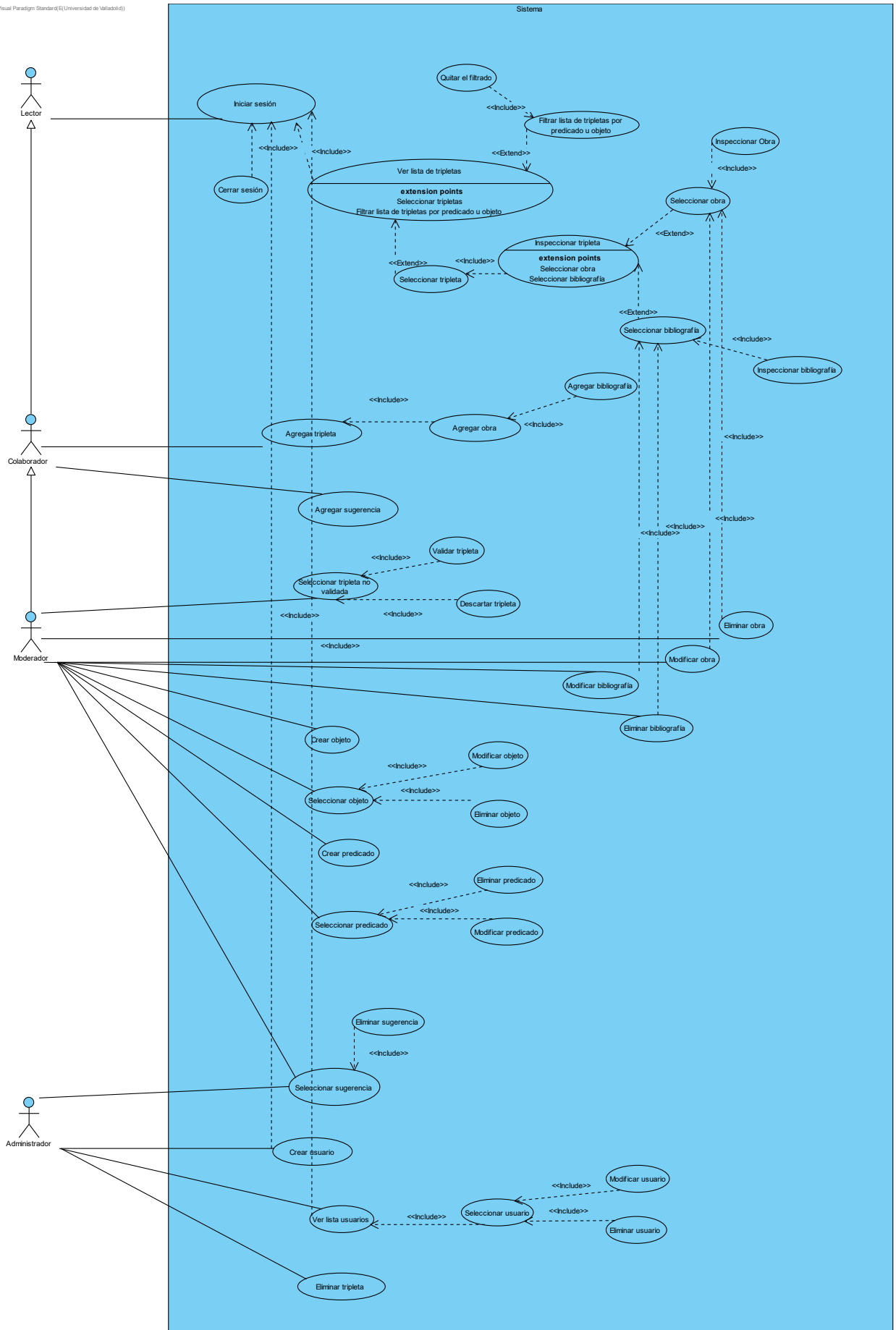


Ilustración 23 - Diagrama de casos de uso del sistema

4.3.3 Especificación de casos de uso

Seguidamente, se muestra la especificación de cada caso de uso en forma de tabla donde aparecen los apartados:

- ID: identificador único del caso de uso.
- Nombre: nombre del caso de uso.
- Dependencias: Requisito con el que se relaciona.
- Actor: actor que interacciona con el sistema.
- Descripción: explicación del caso de uso.
- Precondiciones: condiciones que deben cumplirse antes de que el caso de uso pueda ejecutarse correctamente.
- Postcondiciones: condiciones que deben cumplirse después de que el caso de uso se haya ejecutado con éxito.
- Flujo normal: secuencia de eventos que ocurren dentro del sistema cuando todo funciona según lo esperado.
- Flujos alternativos: diferentes caminos que un sistema puede en caso de que ocurran situaciones divergentes.
- Excepciones: situaciones o condiciones inesperadas que pueden surgir durante la ejecución del caso de uso.
- Frecuencia: cantidad de veces que se espera que se realice una determinada acción o interacción dentro del sistema.

| ID | CU-001 |
|---------------------|--|
| NOMBRE | Iniciar sesión |
| DEPENDENCIAS | RF-001 |
| ACTOR | Administrador, moderador, colaborador, lector. |
| DESCRIPCIÓN | La aplicación muestra los campos que el usuario debe rellenar para iniciar sesión. |
| PRECONDICIONES | |
| POSTCONDICIONES | |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El usuario introduce su nombre de usuario y su contraseña. 2. El usuario pulsa "Enter". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el usuario ha dejado algún campo vacío o los datos introducidos no corresponden a un usuario existe en la base de datos, el sistema le notificará del error y se vuelve al paso 1. |
| FRECUENCIA | Alta. |

Tabla 13 - CU-001

| ID | CU-002 |
|---------------------|---|
| NOMBRE | Crear usuario |
| DEPENDENCIAS | RF-003 |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador inserta un usuario en la base de datos del sistema. |
| PRECONDICIONES | CU-001 |
| POSTCONDICIONES | Los datos de un usuario están agregados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El administrador introduce los datos necesarios en los campos para agregar el usuario a la base de datos del sistema. 2. El administrador pulsa "Crear". 3. El administrador pulsa "Aceptar". 4. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el administrador ha dejado algún campo vacío o el usuario existe, el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Media. |

Tabla 14 - CU-002

| ID | CU-003 |
|---------------------|---|
| NOMBRE | Ver usuarios |
| DEPENDENCIAS | RF-004 |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador ve los datos de los usuarios existentes en la base de datos del sistema. |
| PRECONDICIONES | CU-001 |
| POSTCONDICIONES | |
| FLUJO NORMAL | 1. Se muestran los datos de los usuarios existentes en la base de datos del sistema. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | |
| FRECUENCIA | Media. |

Tabla 15 - CU-003

| ID | CU-004 |
|---------------------|---|
| NOMBRE | Filtrar usuarios por rol |
| DEPENDENCIAS | RF-007 |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador filtra los usuarios existentes por uno de los roles existentes. |
| PRECONDICIONES | CU-003 |
| POSTCONDICIONES | 1. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El administrador selecciona el rol que se usará para aplicar el filtrado. 2. El administrador pulsa “Filtrar”. 3. Se muestra los datos de los usuarios existentes en la base de datos cuyo rol es el seleccionado en el filtro. 4. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | |
| FRECUENCIA | Media. |

Tabla 16 - CU-004

| ID | CU-005 |
|---------------------|--|
| NOMBRE | Seleccionar usuario |
| DEPENDENCIAS | |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador selecciona uno de los usuarios que se muestran. |
| PRECONDICIONES | CU-003 |
| POSTCONDICIONES | |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El administrador selecciona uno de los usuarios que se muestran. 2. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | |
| FRECUENCIA | Alta. |

Tabla 17 - CU-005

| ID | CU-006 |
|---------------------|--|
| NOMBRE | Modificar usuario |
| DEPENDENCIAS | RF-005 |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador modifica los datos del usuario. |
| PRECONDICIONES | CU-005 |
| POSTCONDICIONES | Los datos de un usuario están modificados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El administrador modifica los datos del usuario de la base de datos. 2. El administrador pulsa "Modificar". 3. El administrador pulsa "Aceptar". 4. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1 Si el administrador ha dejado algún campo vacío el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Media. |

Tabla 18 - CU-006

| ID | CU-007 |
|---------------------|---|
| NOMBRE | Eliminar usuario |
| DEPENDENCIAS | RF-006 |
| ACTOR | Administrador |
| DESCRIPCIÓN | El administrador elimina un usuario de la base de datos del sistema. |
| PRECONDICIONES | CU-005 |
| POSTCONDICIONES | Los datos de un usuario se han eliminan de la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El administrador pulsa "Eliminar". 2. El administrador pulsa "Si". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | 2. El administrador pulsa "No". Se avanza hacia el punto 3 del flujo normal y el caso de uso no se completa y queda sin efecto. |
| EXCEPCIONES | |
| FRECUENCIA | Media. |

Tabla 19 - CU-007

| ID | CU-008 |
|---------------------|---|
| NOMBRE | Inspeccionar tripleta |
| DEPENDENCIAS | RF-017 |
| ACTOR | Lector, colaborador, moderador |
| DESCRIPCIÓN | El actor ve los datos de la tripleta existente en la base de datos del sistema. |
| PRECONDICIONES | Seleccionar tripleta. |
| POSTCONDICIONES | |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor pulsa "Ver tripleta". 2. Se muestran los datos de la tripleta existentes en la base de datos del sistema. 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | |
| FRECUENCIA | Alta. |

Tabla 20 - CU-008

| ID | CU-009 |
|---------------------|---|
| NOMBRE | Crear tripleta |
| DEPENDENCIAS | RF-016 |
| ACTOR | Colaborador, moderador |
| DESCRIPCIÓN | El actor inserta una tripleta en la base de datos del sistema. |
| PRECONDICIONES | CU-001 |
| POSTCONDICIONES | Los datos de una tripleta están agregados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor introduce los datos necesarios en los campos para agregar la tripleta a la base de datos del sistema. 2. El actor pulsa "Siguiete". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el actor ha dejado algún campo vacío o la tripleta existe, el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Alta. |

Tabla 21 - CU-009

| ID | CU-010 |
|---------------------|---|
| NOMBRE | Crear obra |
| DEPENDENCIAS | RF-022 |
| ACTOR | Colaborador, moderador |
| DESCRIPCIÓN | El actor inserta una obra en la base de datos del sistema. |
| PRECONDICIONES | CU-009 |
| POSTCONDICIONES | Los datos de una obra están agregados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor introduce los datos necesarios en los campos para agregar la obra a la base de datos del sistema. 2. El actor pulsa "Siguiete". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el actor ha dejado algún campo vacío o la obra existe, el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Alta. |

Tabla 22 - CU-010

| ID | CU-011 |
|---------------------|--|
| NOMBRE | Crear bibliografía |
| DEPENDENCIAS | RF-026 |
| ACTOR | Colaborador, moderador |
| DESCRIPCIÓN | El actor inserta una bibliografía en la base de datos del sistema. |
| PRECONDICIONES | CU-010 |
| POSTCONDICIONES | Los datos de una bibliografía están agregados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor introduce los datos necesarios en los campos para agregar la bibliografía a la base de datos del sistema. 2. El actor pulsa "Crear". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el actor ha dejado algún campo vacío o la bibliografía existe, el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Alta. |

Tabla 23 - CU-011

| ID | CU-012 |
|---------------------|--|
| NOMBRE | Crear sugerencia |
| DEPENDENCIAS | RF-026 |
| ACTOR | Colaborador |
| DESCRIPCIÓN | El actor inserta una sugerencia en la base de datos del sistema. |
| PRECONDICIONES | CU-010 |
| POSTCONDICIONES | Los datos de una sugerencia están agregados en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor introduce los datos necesarios en los campos para agregar la sugerencia a la base de datos del sistema. 2. El actor pulsa "Crear". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | 2.1. Si el actor ha dejado algún campo vacío o la sugerencia existe, el sistema le notificará del error y se procede al paso 1. |
| FRECUENCIA | Alta. |

Tabla 24 - CU-012

| ID | CU-013 |
|---------------------|--|
| NOMBRE | Validar tripleta |
| DEPENDENCIAS | RF-021 |
| ACTOR | Moderador |
| DESCRIPCIÓN | El moderador valida una tripleta de la base de datos del sistema. |
| PRECONDICIONES | Seleccionar tripleta no validada |
| POSTCONDICIONES | Los datos de la tripleta se modifican en la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El moderador pulsa "Validar". 2. El moderador pulsa "Ok". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | |
| EXCEPCIONES | |
| FRECUENCIA | Alta |

Tabla 25 - CU-013

| ID | CU-014 |
|---------------------|---|
| NOMBRE | Descartar tripleta no validada |
| DEPENDENCIAS | RF-019 |
| ACTOR | Moderador |
| DESCRIPCIÓN | El moderador descarta una tripleta de la base de datos del sistema. |
| PRECONDICIONES | Seleccionar tripleta no validada |
| POSTCONDICIONES | Los datos de la tripleta se eliminan de la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El moderador pulsa "Eliminar". 2. El moderador pulsa "Si". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | 2. El moderador pulsa "No". Se avanza hacia el punto 3 del flujo normal y el caso de uso no se completa y queda sin efecto. |
| EXCEPCIONES | |
| FRECUENCIA | Alta |

Tabla 26 - CU-014

| ID | CU-015 |
|---------------------|---|
| NOMBRE | Eliminar sugerencia |
| DEPENDENCIAS | RF-032 |
| ACTOR | Administrador, moderador |
| DESCRIPCIÓN | El administrador elimina una sugerencia de la base de datos del sistema. |
| PRECONDICIONES | CU-005 |
| POSTCONDICIONES | Los datos de una sugerencia se han eliminan de la base de datos del sistema. |
| FLUJO NORMAL | <ol style="list-style-type: none"> 1. El actor pulsa "Eliminar". 2. El actor pulsa "Si". 3. El caso de uso finaliza. |
| FLUJOS ALTERNATIVOS | 2. El actor pulsa "No". Se avanza hacia el punto 3 del flujo normal y el caso de uso no se completa y queda sin efecto. |
| EXCEPCIONES | |
| FRECUENCIA | Media. |

Tabla 27 - CU-015

Capítulo 5 - Arquitectura y diseño

5.1 Arquitectura lógica

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software [48].

Para este proyecto, los patrones utilizados son el patrón cliente-servidor y el patrón arquitectónico Modelo-Vista-Presentador, donde cada vista diferente tiene su propio presentador asociado. Sin embargo, las clases del modelo serán las mismas para todas las vistas, ya que se quiere que la aplicación conserve los cambios realizados en una vista cuando se cambie a otra [49].

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora [50]. En la ilustración 24 se muestra la interacción de la aplicación con la base de datos.

Modelo-Vista-Presentador (MVP) es una derivación del patrón arquitectónico modelo-vista-controlador (MVC), y es utilizado mayoritariamente para construir interfaces de usuario. Facilita las pruebas de unidad automatizada y mejora la separación de inquietudes en lógica de presentación:

- El modelo es una interfaz que define los datos que se mostrarán o sobre los que actuará la interfaz de usuario.
- El presentador actúa sobre el modelo y la vista. Recupera datos de los repositorios (el modelo), y los formatea para mostrarlos en la vista.
- La vista es una interfaz pasiva que exhibe datos (el modelo) y órdenes de usuario de las rutas (eventos) al presentador para actuar sobre los datos.

En la ilustración se muestra cómo se comunican los tres componentes [49].

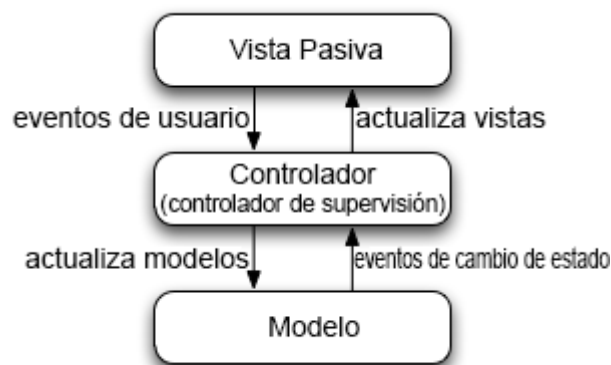


Ilustración 24 - Esquema que describe la IGU del patrón MVP

Las ventajas de la utilización de este patrón son varias [51]:

- Separación mejorada de preocupaciones entre vista y modelo.
- El presentador facilita una mejor capacidad de prueba y modularidad.
- Cada componente puede modificarse o reemplazarse sin afectar a los demás.

- Más adecuado para aplicaciones con requisitos de estado o interacción complejos.

5.2 Diseño de la base de datos

5.2.1 Modelo entidad-relación

El modelo Entidad-Relación (ER) es una técnica de diseño de base de datos de tipo arriba a abajo que comienza identificando los datos más importantes, denominados entidades, y las relaciones entre los datos que deben representarse en el modelo. Después se añaden más detalles, como la información que se quiere almacenar acerca de las entidades y relaciones, lo que se denomina atributos y sobre cualesquiera restricciones que haya que aplicar a las entidades, relaciones y atributos [41].

5.2.2 Base de datos del sistema

El sistema cuenta con una base de datos para la aplicación que se encuentra alojada en un servidor local MySQL.

El diseño lógico de la base de datos del proyecto se representa a través del modelo entidad-relación (ilustración 25) y de él se obtiene la siguiente información:

- Las entidades “Objetos” y “Bibliografías” son las únicas entidades débiles, el resto son entidades fuertes.
- Los atributos de cada entidad.
- Las cardinalidades de las relaciones. Estas representan lo siguiente:
 - Un usuario puede crear varias o ninguna sugerencia.
 - Una sugerencia solo es creada por un usuario.
 - Una tripleta pertenece a uno o dos usuarios, es decir, al creador de la tripleta y al validador de esta. El validador puede ser el propio creador.
 - Un usuario puede ser o no perteneciente a una tripleta, es decir, haber creado/validado una o ninguna.
 - Una tripleta contiene una o varias obras.
 - Una obra pertenece a una sola tripleta.
 - Una tripleta incluye una o varias bibliografías.
 - Una bibliografía pertenece a una sola tripleta.

A continuación, se detallan las tablas con los atributos de la base de datos.

- USUARIOS (id_us, usuario, contrasenia, rol): entidad que representa a las personas que tienen acceso al sistema.
 - id_us: identificador único del usuario (clave primaria).
 - usuario: nombre del usuario.
 - contrasenia: contraseña del usuario.
 - rol: rol adjudicado al usuario.

- OBJETOS (id_o, objeto, color): entidad que representa los sustantivos o adjetivos en las triplas.
 - id_o: identificador único del objeto (clave primaria).
 - objeto: nombre del objeto.
 - color: color asociado al objeto.
- PREDICADOS (id_pr, predicado): entidad que representa los verbos en las triplas.
 - id_pr: identificador único del predicado (clave primaria).
 - predicado: nombre del predicado.
- TRIPLETAS (id_tr, sujeto, predicado, objeto, validada, descripción, colaborador, moderador): entidad que representa información formada por un conjunto de tres elementos relacionados entre sí.
 - id_tr: identificador único de la tripleta (clave primaria).
 - sujeto: nombre del objeto que es el sujeto en la tripleta.
 - predicado: nombre del predicado que es el predicado en la tripleta.
 - objeto: nombre del objeto que es el objeto en la tripleta.
 - validada: valor que indica si la tripleta ha sido validada o no.
 - descripción: breve texto que describe a la tripleta.
 - colaborador: usuario con rol colaborador que ha creado de la tripleta.
 - moderador: usuario con rol moderador que ha validado de la tripleta.
- OBRAS (id_tr, obra, compositor, fragmento): entidad que representa una obra musical en la que se aprecia la información de la tripleta a la que pertenece. La clave primaria se compone de los campos id_tr y obra.
 - id_tr: identificador de la tripleta relacionada a la obra (clave foránea).
 - obra: nombre de la obra.
 - compositor: compositor de la obra.
 - fragmento: parte de la obra que interesa.
 - grabación: grabación de la obra.
 - enlace: enlace a un vídeo de la obra.
- BIBLIOGRAFÍAS (id_tr, título, autor, año, editorial): entidad que representa una bibliografía que justifica la información de la tripleta a la que pertenece. La clave primaria se compone de los campos id_tr y bibliografía.
 - id_tr: identificador de la tripleta relacionada a la bibliografía (clave foránea).

- titulo: nombre de la bibliografía.
 - autor: autor de la bibliografía.
 - año: año de publicación de la bibliografía
 - editorial: nombre de la editorial de la bibliografía.
- SUGERENCIAS (id_su, colaborador): entidad que representa las sugerencias que los usuarios habilitados para crearlas hacen para ampliar y mejorar el sistema.
 - id_su: identificador único de la sugerencia (clave primaria).
 - colaborador: usuario con rol colaborador que ha creado la sugerencia.
 - fecha: fecha en la que se crea la sugerencia.

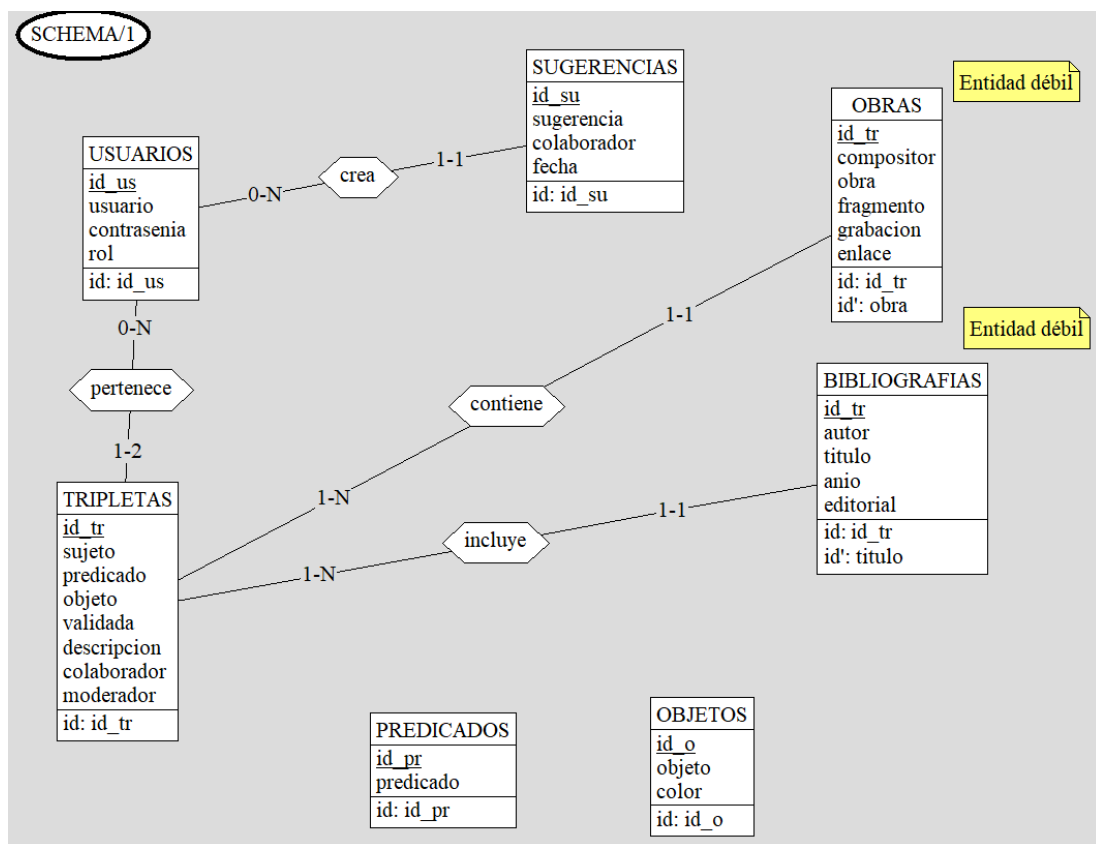


Ilustración 25 - Modelo Entidad-Relación de la base de datos del sistema

5.3 Diagrama de clases

En las ilustraciones 26-28 se muestran los diagramas de clases de diseño divididas en las diferentes capas (Modelo, Vista y Presentador) y omitiendo varias operaciones y atributos para no complicar demasiado la comprensión de estos. Los paquetes con los nombres de los roles se han añadido al diagrama para una mejor estructuración, pero no existen como tal.

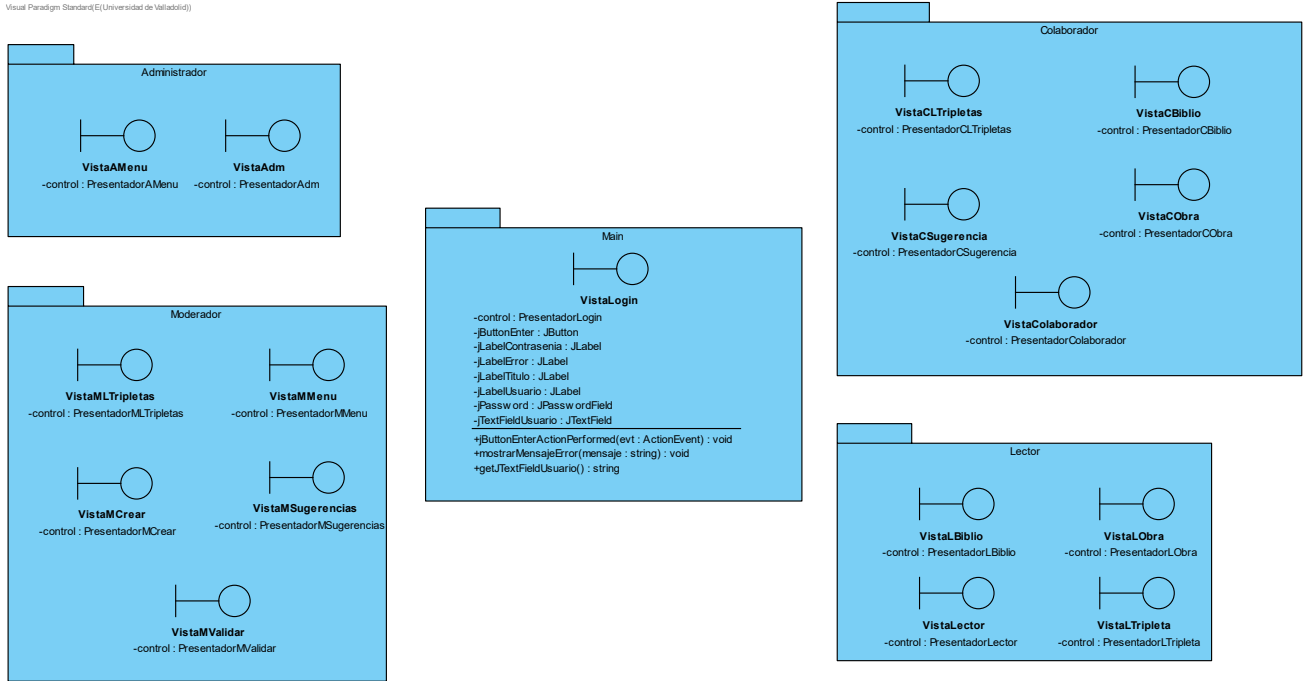


Ilustración 26 - Diagrama de clases Vista del sistema

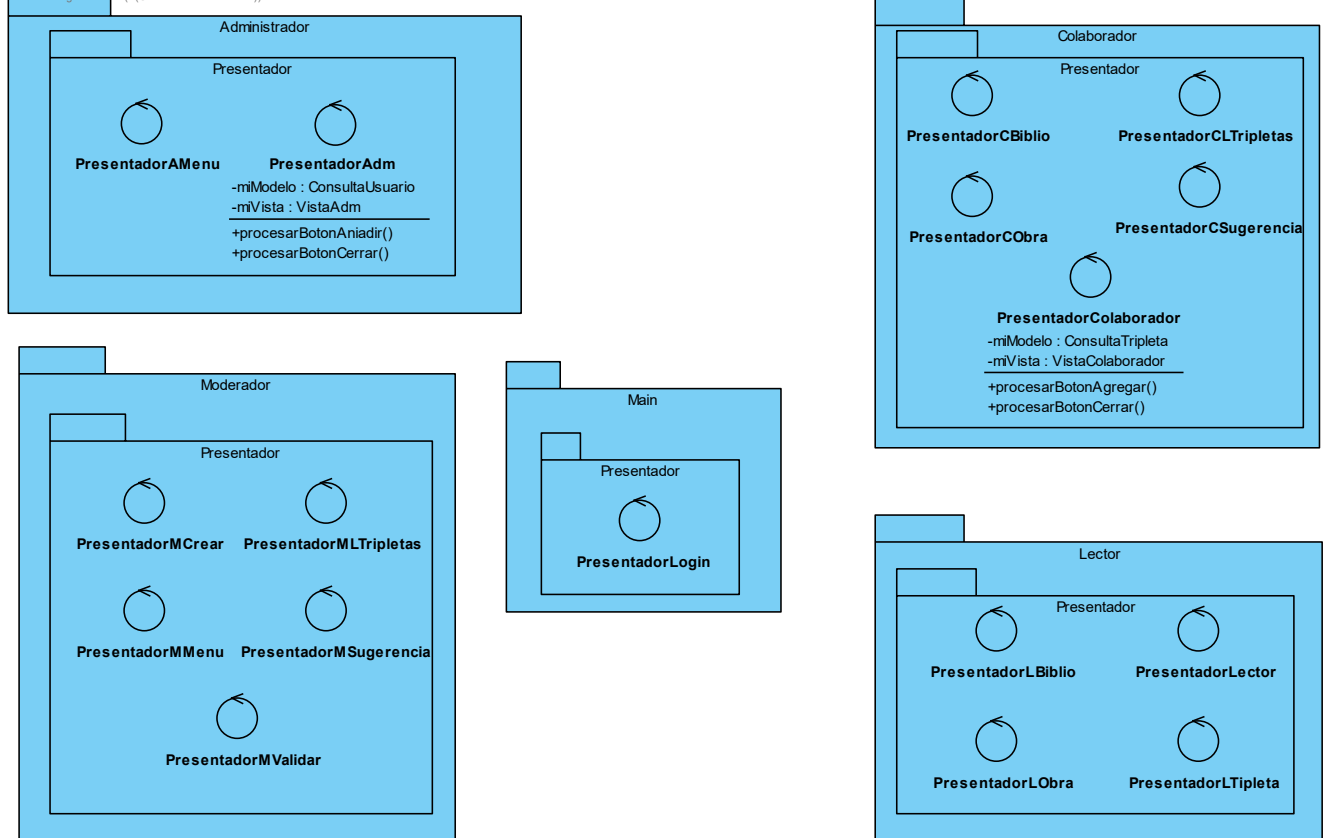


Ilustración 27 - Diagrama de clases Presentador del sistema

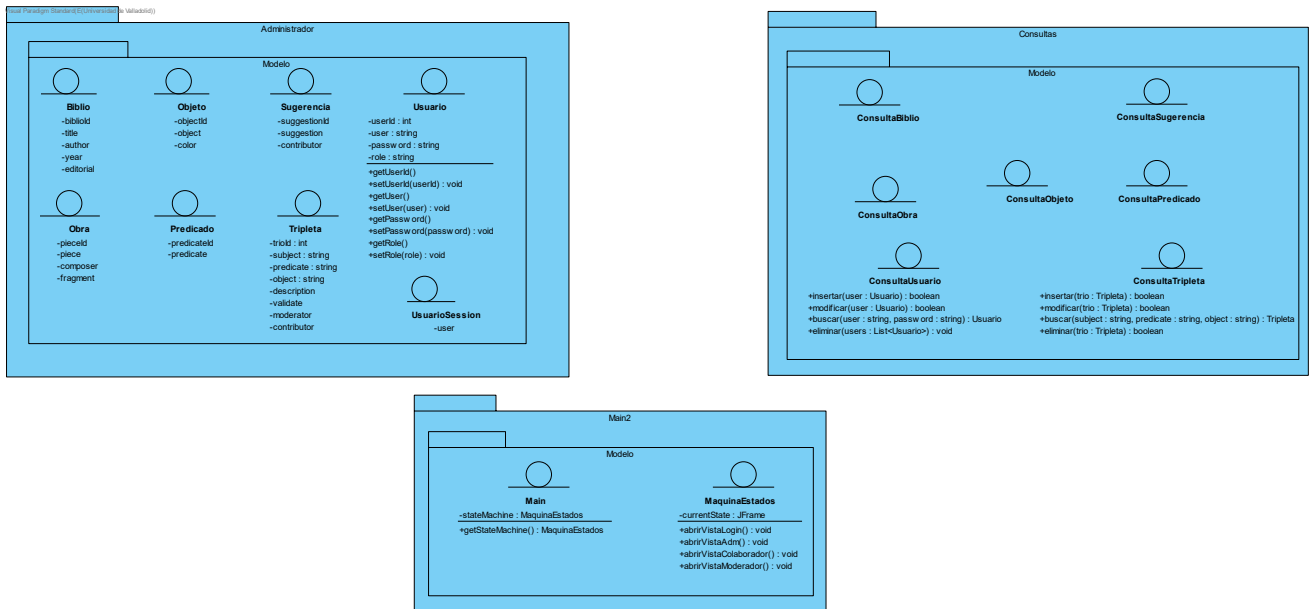


Ilustración 28 - Diagrama de clases Modelo del sistema

5.4 Máquina de estados

5.4.1 Concepto

Una máquina de estado es un modelo matemático de computación que consta principalmente de los siguientes elementos [52]:

- Estados: se definen como un conjunto de valores o comportamientos por lo que puede pasar un objeto. Siempre tenemos que definir un estado inicial que será el primero por el que pasa el objeto.
- Eventos: Son acciones, ya sean internas o externas que hacen que el objeto cambie de estado.
- Transiciones: Son los posibles caminos por los que el objeto cambia de un estado a otro.

Las máquinas de estados se pueden utilizar para modelar la lógica interna de una aplicación y definir los diferentes comportamientos según las entradas que se produzcan en el sistema.

En el proyecto, la máquina de estados se encarga de gestionar las diferentes vistas de la aplicación.

5.4.2 Diagrama de máquina de estados

Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) muestra los estados por los que pasa una máquina de estados finitos, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. El diagrama proporciona un estado inicial y uno final, así como al menos un estado intermedio para cada objeto [53].

Para una mejor visualización del diagrama (ilustración 29), se ha especificado con distintos colores los estados (vistas de la aplicación) dependiendo del rol del usuario (administrador, moderador, colaborador y lector):

- El verde es el común a todos los roles.
- El amarillo pertenece al rol administrador.
- El naranja pertenece al rol moderador.

- El violeta pertenece al rol colaborador.
- El azul pertenece al rol lector.

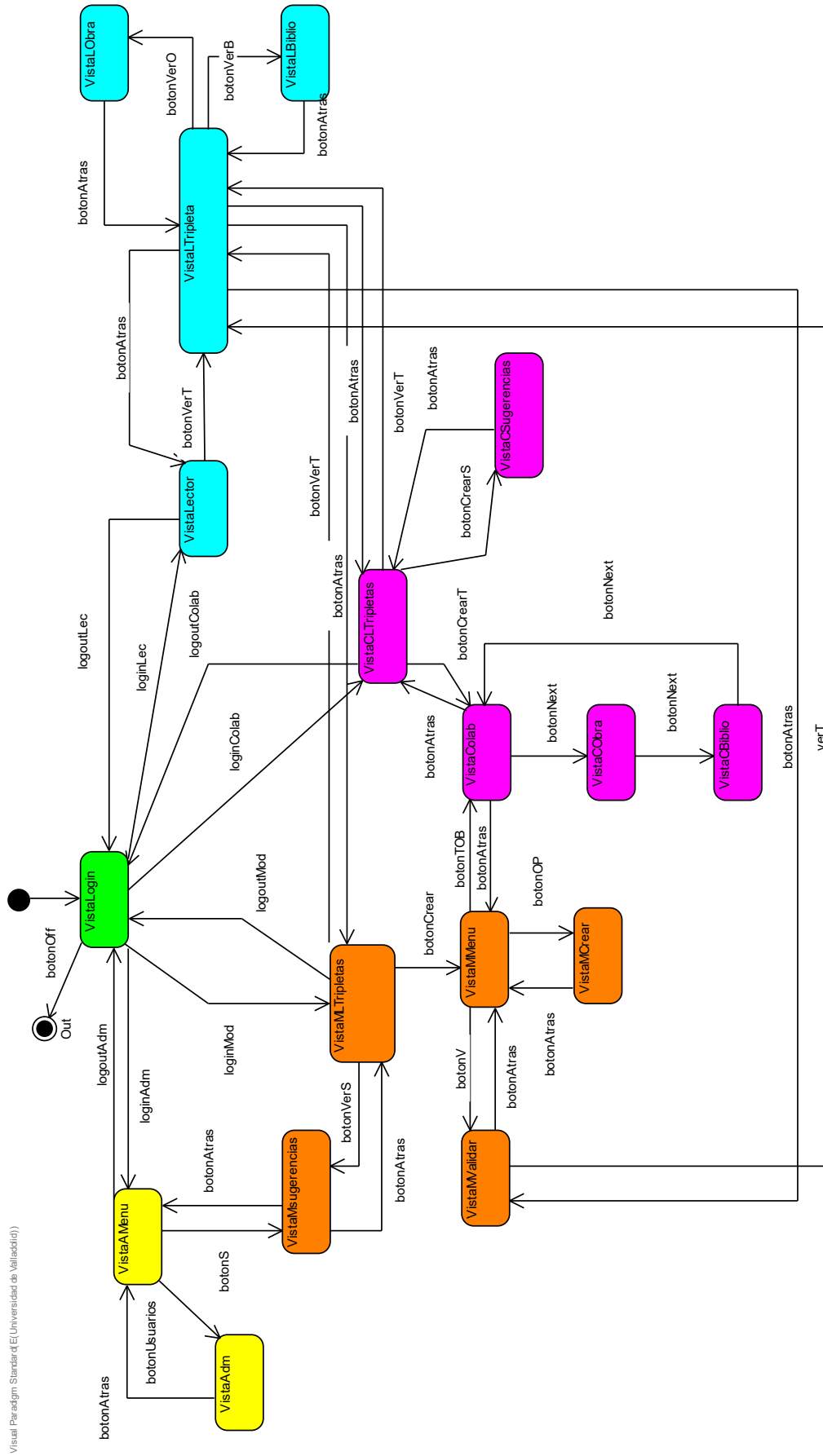


Ilustración 29 - Diagrama de máquina de estados del sistema

5.5 Diagrama de secuencia

El diagrama de secuencia describe básicamente cómo los objetos (e instancias) intercambian mensajes en un orden determinado. El elemento del diagrama de secuencia es la línea de vida de un objeto. [54]

Se han realizado tres diagramas de secuencia con el actor lector de los casos de uso:

- Inicio de sesión: el lector inicia sesión en el sistema.
- Ver tripletas: el lector visualiza las tripletas validadas existentes en la base de datos.
- Inspeccionar tripletas: el lector inspecciona los detalles de una triplete.

5.5.1 Inicio de sesión

A continuación, se detalla el diagrama de secuencia con el caso de uso iniciar sesión con el rol lector (ilustración 30).

- 1. iniciarSesión(userName : String, passwd : String): el lector introduce en la vistaLogin el nombre de usuario y la contraseña.
- 1.1. iniciarSesión(userName : String, passwd : String): la vistaLogin pasa los datos al presentadorLogin.
- 1.1.1. iniciar(userName : String, passwd : String): el presentador pasa los datos al modelo al consultaUsuario para la realización de la consulta a la base de datos.
- 1.1.1.1. devuelveUsuario(usuario : Usuario): consultaUsuario devuelve al presentadorLogin el usuario correspondiente con los datos pasados.
- 2. usuarioSesion(usuario : Usuario): se crea una instancia del usuario que ha iniciado sesión si usuario no es null.
- 3., 4., 5., 6. abrir(): cierra la ventanaLogin y abre la ventana correspondiente al rol del usuario que quiere iniciar sesión.
- 7. mensajeError(mensaje: String): presentadorLogin pasa a la ventanaLogin el mensaje de error si el usuario es null, es decir, los datos introducidos no se corresponden con ningún usuario de la base de datos.

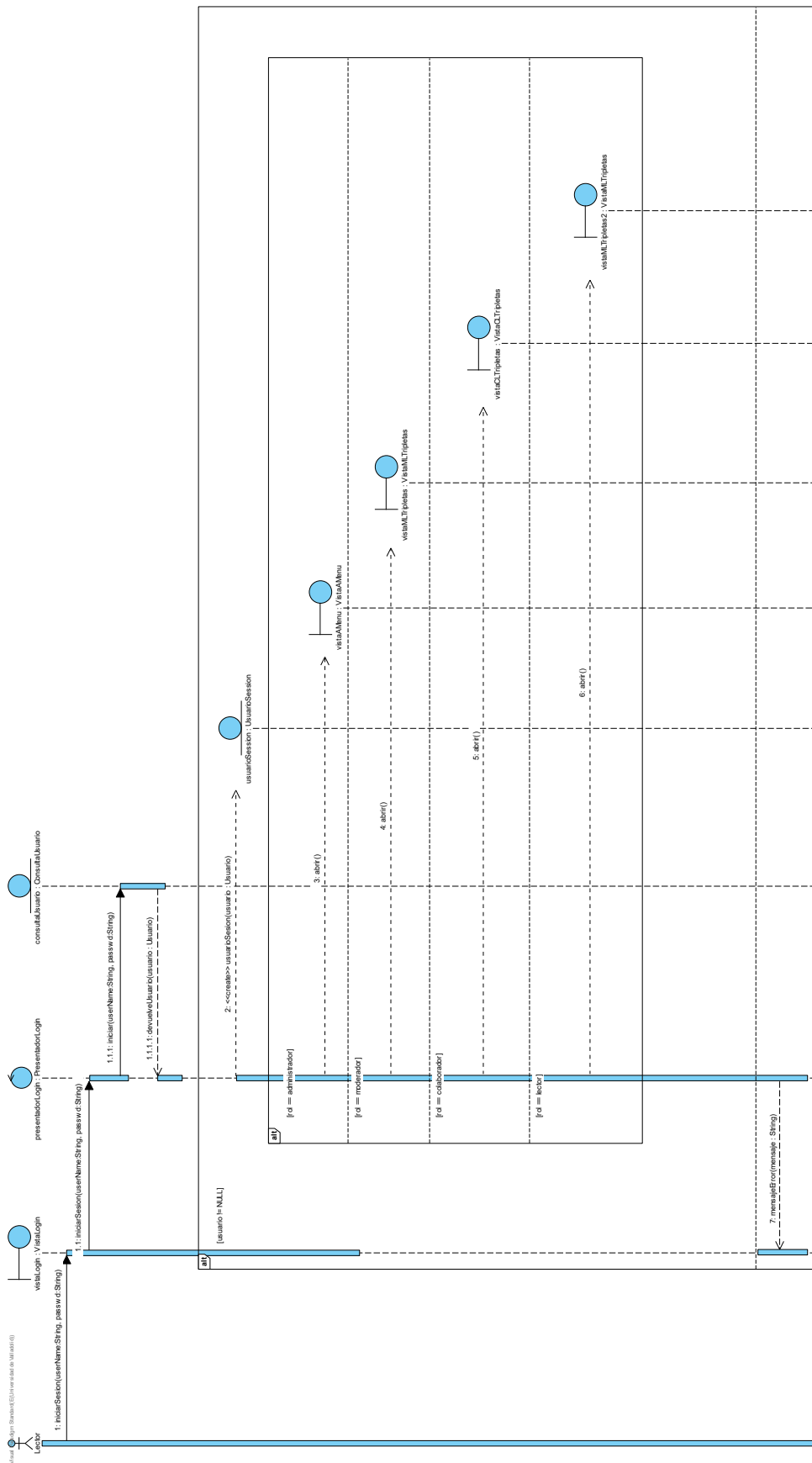


Ilustración 30 - Diagrama de secuencia - Iniciar sesi

5.5.2 Ver tripletas

A continuación, se detalla el diagrama de secuencia con el caso de uso ver tripletas con el rol lector (ilustración 31).

- 1. cargarFiltro2(filtro1 : String): la vistaLector pasa el filtro1 al presentadorLector.
- 1.1. cargarObjetos(): si el valor del argumento pasado es “objetos”, presentadorLector se comunica con ConsultaObjeto para la realización de la consulta a la base de datos.
- 1.2. devuelveObjetos(listaObjetos : ArrayList<<Objeto>>): consultaObjeto devuelve la lista de objetos a presentadorLector.
- 1.3. cargarPredicados(): si el valor del argumento pasado es “predicados”, presentadorLector se comunica con ConsultaPredicado para la realización de la consulta a la base de datos.
- 1.3.1. devuelvePredicados(listaPredicados : ArrayList<<Predicado>>): consultaPredicado devuelve la lista de objetos a presentadorLector.
- 1.3.1.1. devuelveObjetos(listaObjetos : ArrayList<<Objeto>>): presentadorLector pasa a vistaLector la listaObjetos.
- 1.3.2.1. devuelvePredicados(listaPredicados : ArrayList<<Predicado>>): presentadorLector pasa a vistaLector la listaPredicados.
- 1.3.1.2.1. cargarTabla(): vistaLector se comunica con presentadorLector para cargar la tabla.
- 1.3.1.2.1.1. cargarTripletas(): presentadorLector se comunica con ConsultaTripleta para la realización de la consulta a la base de datos.
- 1.3.1.2.1.1.1. devuelveTripletas(listaTripletas : ArrayList<<Tripleta>>): consultaTripleta devuelve la lista de tripletas a presentadorLector.
- 1.3.1.2.1.1.1. devuelveTripletas(listaTripletas : ArrayList<<Tripleta>>): presentadorLector pasa a vistaLector la listaTripletas.

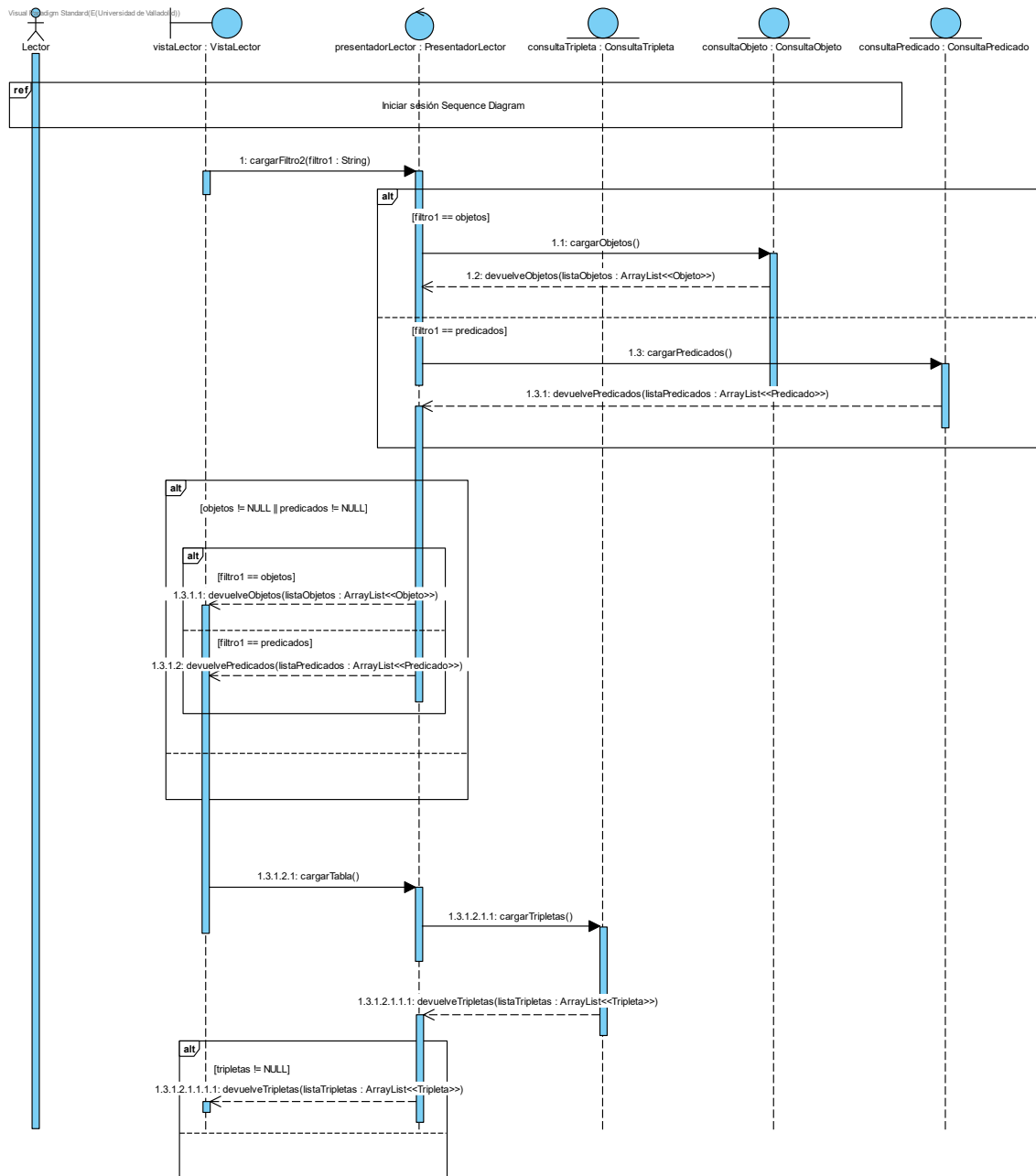


Ilustración 31 - Diagrama de secuencia - Ver tripletas

5.5.3 Inspeccionar tripletas

A continuación, se detalla el diagrama de secuencia con el caso de uso iniciar sesión con el rol lector (ilustración 32).

- **1. verTripleta():** el lector selecciona “Ver Tripleta”.
- **1.1. selecTripleta(numFila: int):** vistaLector pasa el número de filas seleccionadas a presentadorLector.
- **1.1.1. verTripleta(fila: int):** vistaLector pasa la fila seleccionada a presentadorLector.
- **1.2.1. abrir(idTripleta : int):** presentadorLector abre vistaLTripleta y le pasa el idTripleta.
- **2. cargarTripleta(id_tr : int):** vistaLTripleta pasa a presentadorLTripleta el id de la tripleta seleccionada.
- **2.1 cargarTripleta(id_tr : int):** presentadorLTripleta pasa el id_tr a consultaTripleta para la realización de la consulta a la base de datos.

- 2.1.1. devuelveTripleta(tripleta : Tripletta): consultaTripletta devuelve a presentadorLTripletta la tripletta correspondiente al id pasado.
- 2.1.1.1. devuelveTripleta(tripleta : Tripletta): presentadorLector pasa a vistaLector la tripletta correspondiente al id pasado..
- 2.1.1.2. cargarObras(id_tr : int): presentadorLTripletta pasa el id_tr a consultaObras para la realización de la consulta a la base de datos.
- 2.1.1.3. devuelveObras(listaObras : ArrayList<<Obras>>): consultaObras devuelve a presentadorLTripletta la listaObras.
- 2.1.1.4. devuelveObras(listaObras : ArrayList<<Obras>>): presentadorLector pasa a vistaLector la listaObras.
- 2.1.1.5. cargarbiblios(id_tr : int): presentadorLTripletta pasa el id_tr a consultaBiblio para la realización de la consulta a la base de datos.
- 2.1.1.6. devuelveBiblio(listaBiblios : ArrayList<<Biblio>>): consultaBiblio devuelve a presentadorLTripletta la listaBiblios.
- 2.1.1.7. devuelveBiblio(listaBiblios : ArrayList<<Biblio>>): presentadorLector pasa a vistaLector la listaBiblios.
- 1.2.2 aviso(): presentadorLector pasa el mensaje de aviso a vistaLector.

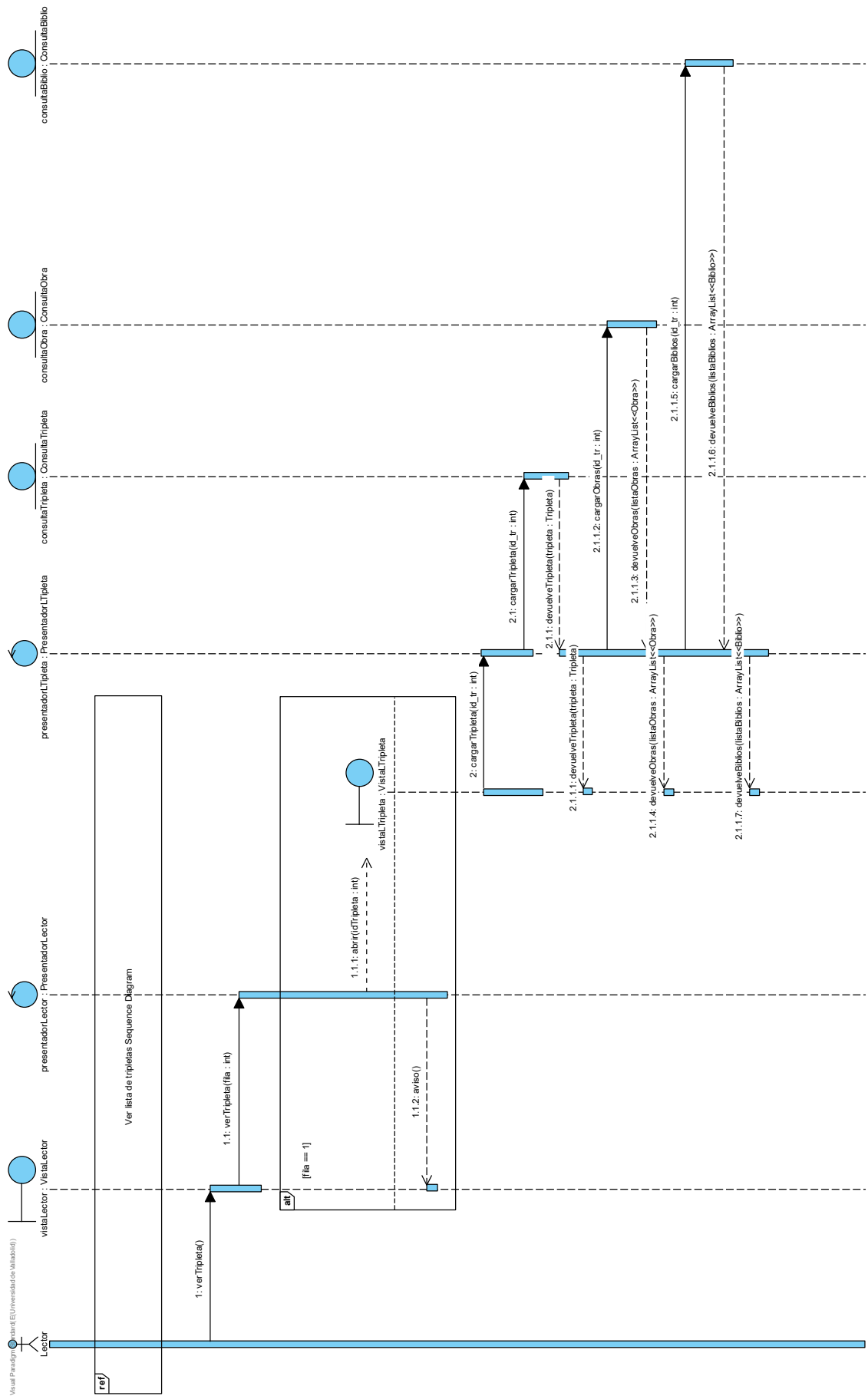


Ilustración 32 - Diagrama de secuencia - Inspeccionar tripleta

5.6 Interfaz de usuario

5.6.1 Decisiones de diseño

A la hora del desarrollo de la interfaz de la aplicación se ha querido dar prioridad a la lógica y por ende se hace uso de un diseño sencillo y claro: elementos visuales de interacción mínimos, con el objetivo de eliminar cualquier distracción.

5.6.2 Bocetos

Para la realización de bocetos de la interfaz de usuario se utilizó la herramienta NetBeans IDE. Se comenzó creando distintas vistas en archivos JFrame con los componentes necesarios como botones, campos para escribir, etiquetas, etc. En las siguientes ilustraciones se muestran algunos de los bocetos con su correspondiente caso de uso (definidos previamente):

- Este boceto (ilustración 33) representa la vista de inicio de sesión que visualiza un usuario, independientemente del rol que tenga.



Ilustración 33 muestra un boceto de la interfaz de usuario para el inicio de sesión. El título de la ventana es "TRIPLETAS DE TÓPICOS MUSICALES". Hay dos campos de texto: "Usuario" y "Contraseña". Debajo de los campos hay un botón "Enter".

Ilustración 33 - Boceto CU-001

- Este boceto (ilustración 34) representa la vista de creación de usuarios que visualiza un usuario con el rol administrador.



Ilustración 34 muestra un boceto de la interfaz de usuario para la creación de nuevos usuarios. El título de la ventana es "CREACIÓN NUEVOS USUARIOS". Hay tres campos de texto: "Nuevo Usuario", "Contraseña" y "Rol". El campo "Rol" tiene un menú desplegable con "Item 1" seleccionado. Hay dos botones: "Cerrar sesión" y "Añadir".

Ilustración 34 - Boceto CU-002

Capítulo 6 - Implementación

6.1 Herramientas utilizadas

Las herramientas utilizadas para la creación del proyecto han sido:

- Windows 10: Sistema operativo sobre el que se ha empleado el entorno de desarrollo de la aplicación.
- Apache NetBeans IDE 12.6: Entorno de desarrollo integrado libre para la creación de aplicaciones Java.
 - JDK (*Java Development Kit*) 11: La aplicación ha sido compilada por la versión 11 de Java instalada en el sistema operativo.
 - Dependencias: Para ampliar algunas funcionalidades de la aplicación se han añadido librerías de código abierto buscadas en el repositorio de Maven. Al generar por defecto el proyecto, NetBeans utiliza Maven, una tecnología que ayuda con la gestión de dependencia y la construcción del proyecto.
- Paquete Office 365 - Word (licencia académica): Herramienta de procesamiento de texto. Se ha utilizado para la creación del documento TFG.
- Paquete Office 365 - Excel (licencia académica): Herramienta que permite editar hojas de cálculo. Se ha utilizado para la creación de todas las tablas y algunas imágenes del documento.
- Microsoft Visio (licencia académica): Herramienta de diagramación y gráficos vectoriales. Se ha utilizado para la creación de los diagramas WBS y PBS.
- Microsoft Project (licencia académica): Herramienta de administración de proyectos y programas de proyectos para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Se ha utilizado para la creación del diagrama de Gantt.
- Visual Paradigm 16.2 (licencia académica): Herramienta de diseño y gestión para el desarrollo de sistemas de TI empresariales. Se ha usado para la creación de los distintos diagramas UML y para el diagrama PERT.
- Mysql Workbench 8.0 CE: Herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL. Se ha usado para la gestión de la base de datos.
- DB Main: Herramienta gráfica CASE libre que admite la mayoría de los procesos de ingeniería de bases de datos. Se ha usado para la creación del modelo entidad-relación de la base de datos.
- Notepad++: Editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Se ha usado para la creación del código Turtle y para la visualización de distintos formatos de serialización de RDF.

6.1.1 Librerías utilizadas

Apache Jena

Es un *framework* Java gratuito y de código abierto para crear aplicaciones de web semántica y datos vinculados [55] [56]. En Jena, toda la información de estado proporcionada por una colección de triples RDF está contenida en una estructura de datos llamada Model.

En términos de Java, Jena utiliza la clase Model como contenedor principal de información RDF contenida en forma de gráfico. Model está diseñado para tener una API (*Application Programming Interface*) rica, con muchos métodos destinados a facilitar la escritura de programas y aplicaciones basados en RDF. Una de las otras funciones de Model es proporcionar una abstracción sobre diferentes formas de almacenar los nodos y las relaciones RDF.

JDBC (Java Database Connectivity)

JDBC es una API de Java que permite acceder a cualquier tipo de datos tabulares, especialmente a los datos almacenados en una base de datos relacional.

JDBC ayuda a escribir aplicaciones Java que gestionan estas tres actividades de programación:

1. Conectarse a una fuente de datos, como una base de datos.
2. Enviar consultas y actualizar declaraciones a la base de datos.
3. Recuperar y procesar los resultados recibidos de la base de datos en respuesta a la consulta enviada.

JDBC Arquitectura

Para el acceso a la base de datos JDBC usa el modelo de procesamiento de dos niveles (*two-tier model*). En este modelo un subprograma o aplicación de Java se comunica directamente con la fuente de datos a través de un controlador JDBC. Las consultas del usuario se entregan a la base de datos u otra fuente de datos, y los resultados de esas declaraciones se envían de vuelta al usuario. La fuente de datos puede estar ubicada localmente o en otra máquina a la que el usuario esté conectado a través de una red (configuración cliente/servidor). En la ilustración 36 se muestra la arquitectura de dos niveles.

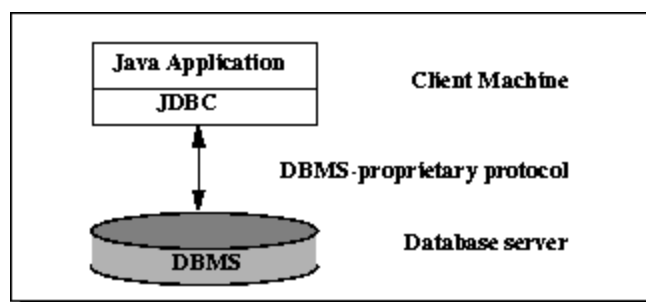


Ilustración 36 – Arquitectura de dos niveles de JDBC

6.2 Desarrollo

La implementación del proyecto se ha dividido en diferentes fases:

- Fase 1: Implementación de la máquina de estados para la gestión de las distintas vistas, de la aplicación del patrón MVP. Se usan datos aleatorios.

- Fase 2: Creación del script de creación y población (con datos aleatorios) de la base de datos, implementación de la aplicación con los datos procedente de la base de datos.
- Fase 3: Implementación de las funcionalidades restantes, uso de datos obtenidos de los vídeos del proyecto docente [2], creación del fichero RDF.

En total, se han implementado 53 clases.

6.2.1 Fase 1

En la herramienta Apache NetBeans IDE se comienza la creación del proyecto completamente en lenguaje Java y siguiendo el patrón MVP junto con la máquina de estados [57]. Para la creación de las distintas interfaces de usuario (vistas) se utiliza el paquete Swing que proporciona widgets como por ejemplo botones, tablas o desplegados (combo box). Se decide que cada vista creada tenga su propio presentador, obteniendo así una separación de responsabilidades y una facilidad para la reutilización del código. Por último, el modelo tiene una clase representativa de cada entidad de la base de datos.

En el tema visual, se decide crear filtro para las tablas con el fin de simplificar el rango de datos. Además, el contenido de los filtros se ordena alfabéticamente para una búsqueda más rápida. El nombre de los botones intenta indicar la acción que realiza, por ejemplo, si un botón se llama “Cerrar sesión” la acción que debe realizar es cerrar la sesión del usuario.

6.2.2 Fase 2

Desde NetBeans se crea el script DDL y DML de la base de datos que se utiliza para crear y poblar (con datos aleatorios correspondientes al tipo definido) la base en MySQL *Workbench* (ilustración 37). Al paquete modelo, se le adjunta una clase encargada específicamente de la conexión con la base de datos (ilustración 38) y una clase por cada tabla de la base de datos, encarga cada una de la interacción con esta. Por ejemplo, teniendo la entidad usuarios se tiene una clase que se encarga de la interacción llamada consulta usuario.

```
-- -Autor: Ester Garcia Perez
create database `TOPICOS_MUSICALES`;
use `TOPICOS_MUSICALES`;

SET foreign_key_checks = 0;

-- -USUARIOS
drop table if exists usuarios;
create table usuarios (
id_us integer not null AUTO_INCREMENT,
usuario varchar(25) not null,
contrasenia varchar(25) not null,
rol varchar(25) not null,
constraint us_id primary key (id_us)
)engine=InnoDB default charset=utf8mb4 collate=utf8mb4_0900_ai_ci;

lock table `usuarios` write;
insert into usuarios(usuario,contrasenia,rol) values ('adm','adm','administrador');
insert into usuarios(usuario,contrasenia,rol) values ('colab','colab','colaborador');
insert into usuarios(usuario,contrasenia,rol) values ('mod','mod','moderador');
insert into usuarios(usuario,contrasenia,rol) values ('lec','lec','lector');
unlock table;
```

Ilustración 37 - Parte del script DDL y DML de la base de datos

En la ilustración 38 se muestra un fragmento de código de la clase Conexión que crea una instancia del objeto *DriverManager* para conectarse al controlador de base de datos e iniciar sesión en la base de datos local (el usuario utilizado es un usuario creado por el administrador *root* y al cual se le han concedido todos los privilegios sobre la base de datos creada). Los drivers JDBC son adaptadores del lado del cliente (instalados en la máquina cliente, no en el servidor) que convierten la petición proveniente del programa JAVA a un protocolo que el SGBD pueda entender [58]. Si se quiere conectar a una base de datos remota únicamente habrá que modificar las constantes (db, ip, port, userDB y passDB) con los datos correspondientes.

```
* @author Ester García Pérez
*/
import java.sql.*;
import javax.swing.JOptionPane;

public class Conexion {

    //constantes
    private static final String db = "temas_musicales"; //nombre de la base de datos
    private static final String ip = "localhost"; // direccion
    private static final String port = "3306"; // numero del puerto
    private static final String userDb = "admBD"; // usuario mysql
    private static final String passDb = "admBD"; // contraseña del usuario
    private static final String cadenaConexion="jdbc:mysql://" + ip + ":" + port + "/" + db;

    public static Connection getConnection() {
        Connection connectionDb = null;
        try {
            Class.forName(className:"com.mysql.cj.jdbc.Driver");
            connectionDb = DriverManager.getConnection(cadenaConexion,userDb,passDb);
        }
    }
}
```

Ilustración 38 - Parte del código de la clase Conexión

A continuación, se muestran las distintas vistas de la aplicación clasificadas por el rol del usuario y con datos aleatorios:

- Todos los roles



The screenshot shows a window titled "Iniciar sesión" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area has the title "RED DE TÓPICOS MUSICALES" in bold. Below the title, there are two text input fields: the first is labeled "Usuario" and the second is labeled "Contraseña". At the bottom of the form, there are two buttons: "OFF" on the left and "Enter" on the right. The entire form is centered on a white background.

Ilustración 39 - Vista de inicio de sesión

- Administrador

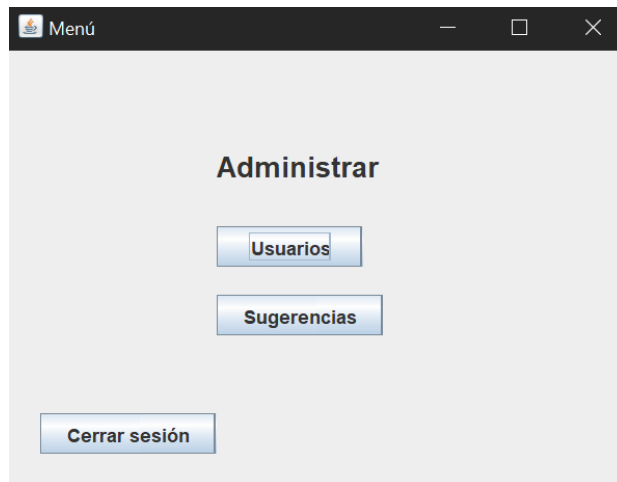


Ilustración 40 - Vista menú (administrador)



Ilustración 41 - Vista administración de usuarios



Ilustración 42 - Vista administración de sugerencias

- Lector



Ilustración 43 - Vista de tripletas (lector)

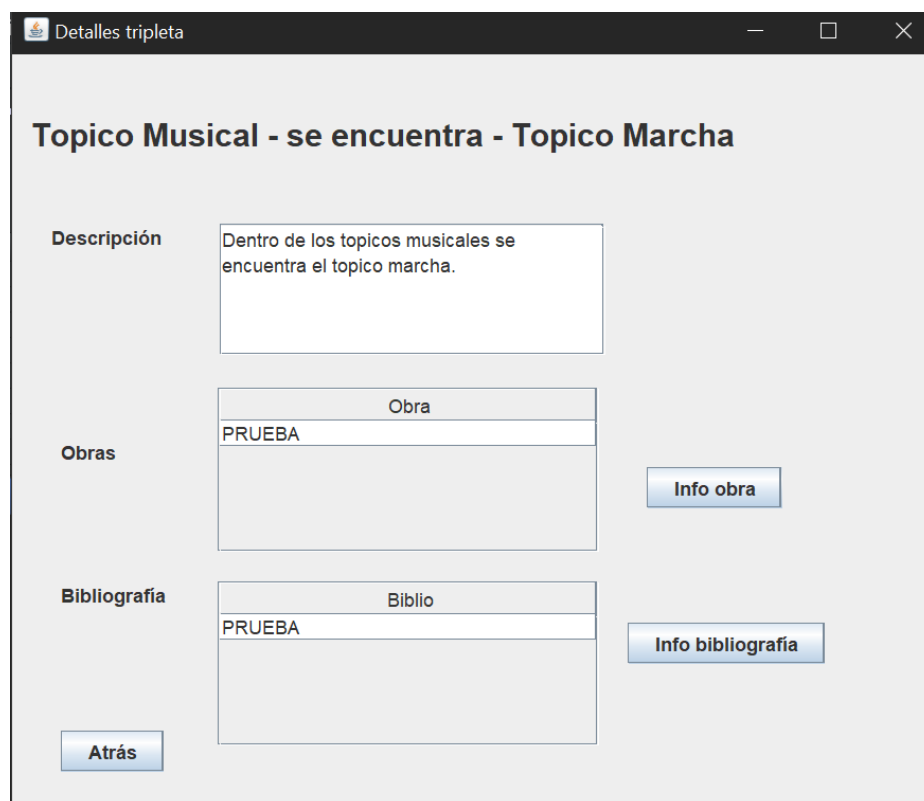


Ilustración 44 - Vista detalles tripleta

Detalles obra

PRUEBA

Compositor prueba

Fragmento 1ºmov

Grabación

Enlace

Atrás

Ilustración 45 - Vista detalles obra

Detalles bibliografía

Bibliografía

Título PRUEBA

Autor prueba

Año 2024

Editorial pruebas

Atrás

Ilustración 46 - Vista detalles bibliografía

- Colaborador



Ilustración 47 - Vista lista de tripletas (colaborador)

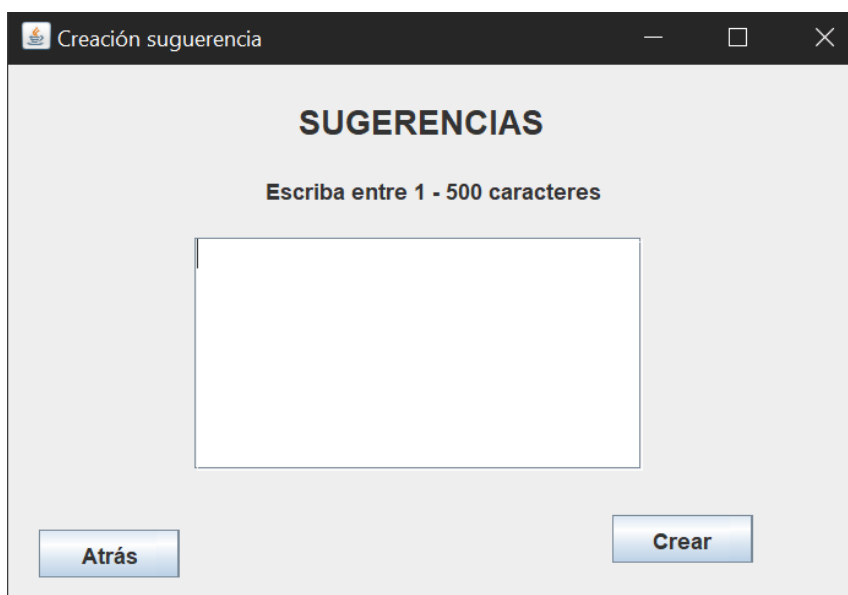


Ilustración 48 - Vista creación sugerencia

Creación bibliografía

CREACIÓN BIBLIOGRAFÍA

Título

Autor

Año

Editorial

Ilustración 51 - Vista creación bibliografía

- Moderador

Lista de tripletas

Objeto

| Sujeto | Predicado | Objeto |
|----------------|--------------|------------------------|
| Topico Musical | se encuentra | Topico Marcha |
| Topico Musical | se encuentra | Topico Ombra |
| Topico Musical | se encuentra | Topico Minuet |
| Topico Musical | se encuentra | Topico Tempesta |
| Topico Marcha | aparece | Trompeta |
| Topico Marcha | aparece | Tambor |
| Topico Marcha | se encuentra | Topico Marcha Solemne |
| Topico Marcha | se encuentra | Topico Marcha Triunfal |
| Topico Marcha | se encuentra | Topico Marcha Fúnembre |
| Topico Ombra | aparece | Trompeta |

Ilustración 52 - Vista lista de tripletas (moderador)



Ilustración 53 - Vista menú (moderador)

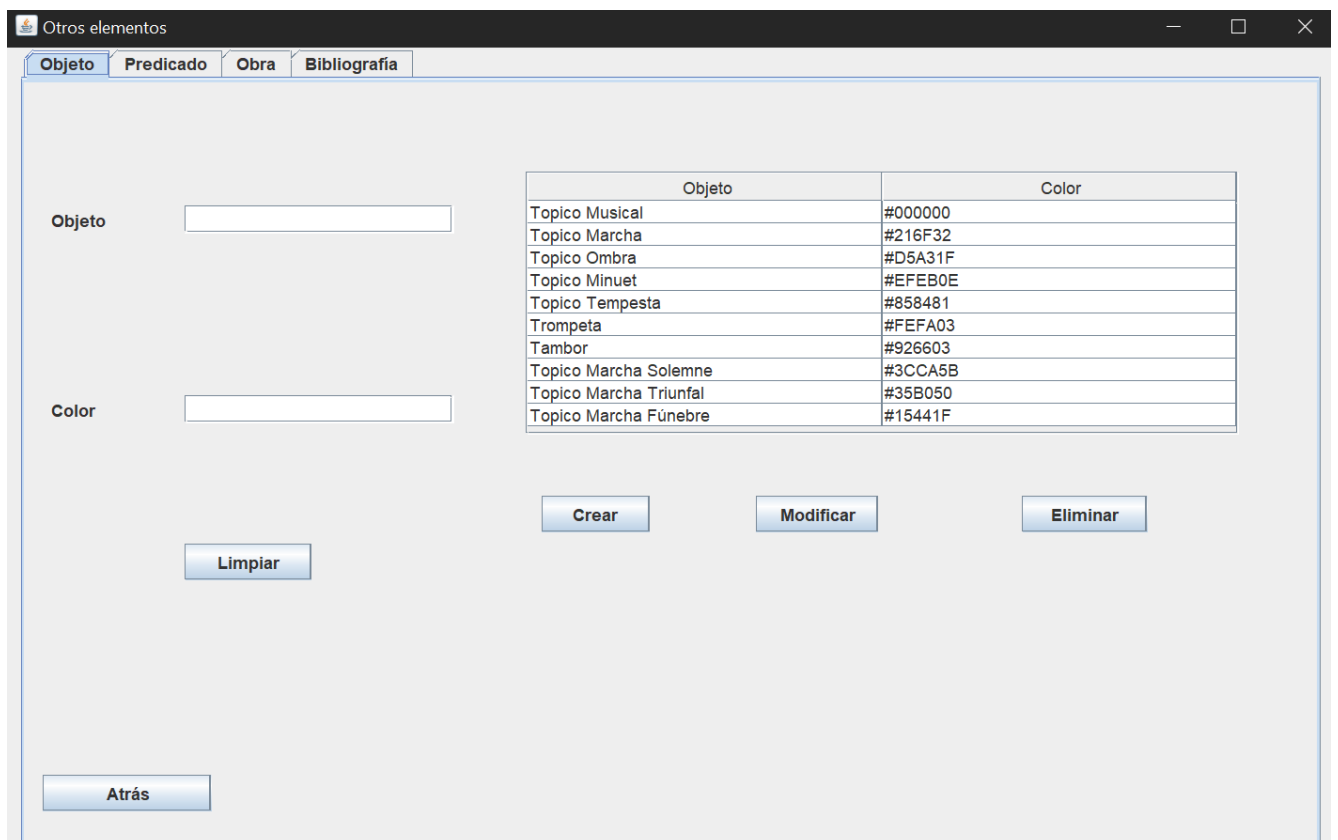


Ilustración 54 - Vista otros elementos



Ilustración 55 - Vista validación tripletas

6.2.3 Fase 3

La lógica de la aplicación se basa en la realización de operaciones CRUD (*Create, Read, Update, Delete*) sobre las tablas de la base de datos y en la creación de un fichero con formato de serialización RDF.

Para la visualización de grafos RDF, además de las herramientas mencionadas en el apartado 3.2.3, se buscó otra alternativa para la propia aplicación. Se encontró la librería GraphStream, una biblioteca Java que maneja grafos y se centra en los aspectos dinámicos de estos. Su principal objetivo es el modelado de redes de interacción dinámicas de varios tamaños: grafos dirigidos y no dirigidos, multigrafos.

GraphStream permite almacenar cualquier tipo de atributo de datos en los elementos del grafo y proporciona una forma de manejar la evolución del gráfico en el tiempo. Esto significa manejar la forma en que se agregan y eliminan nodos y arcos, y la forma en que los atributos de datos pueden aparecer, desaparecer y evolucionar. Además, permite personalizar el grafo: estilo y tamaño de letra, colores, formas de los arcos, etc, lo que ayudaría para el atributo color de la entidad objeto.

Sin embargo, al final por falta de tiempo no se ha podido explorar en profundidad y no se ha implementado en la aplicación final, pero se comenta como una posible línea futura.

A continuación, en la ilustración 56 se muestra el grafo con estilo personalizado resultante de usar el código mostrado en la ilustración 13.

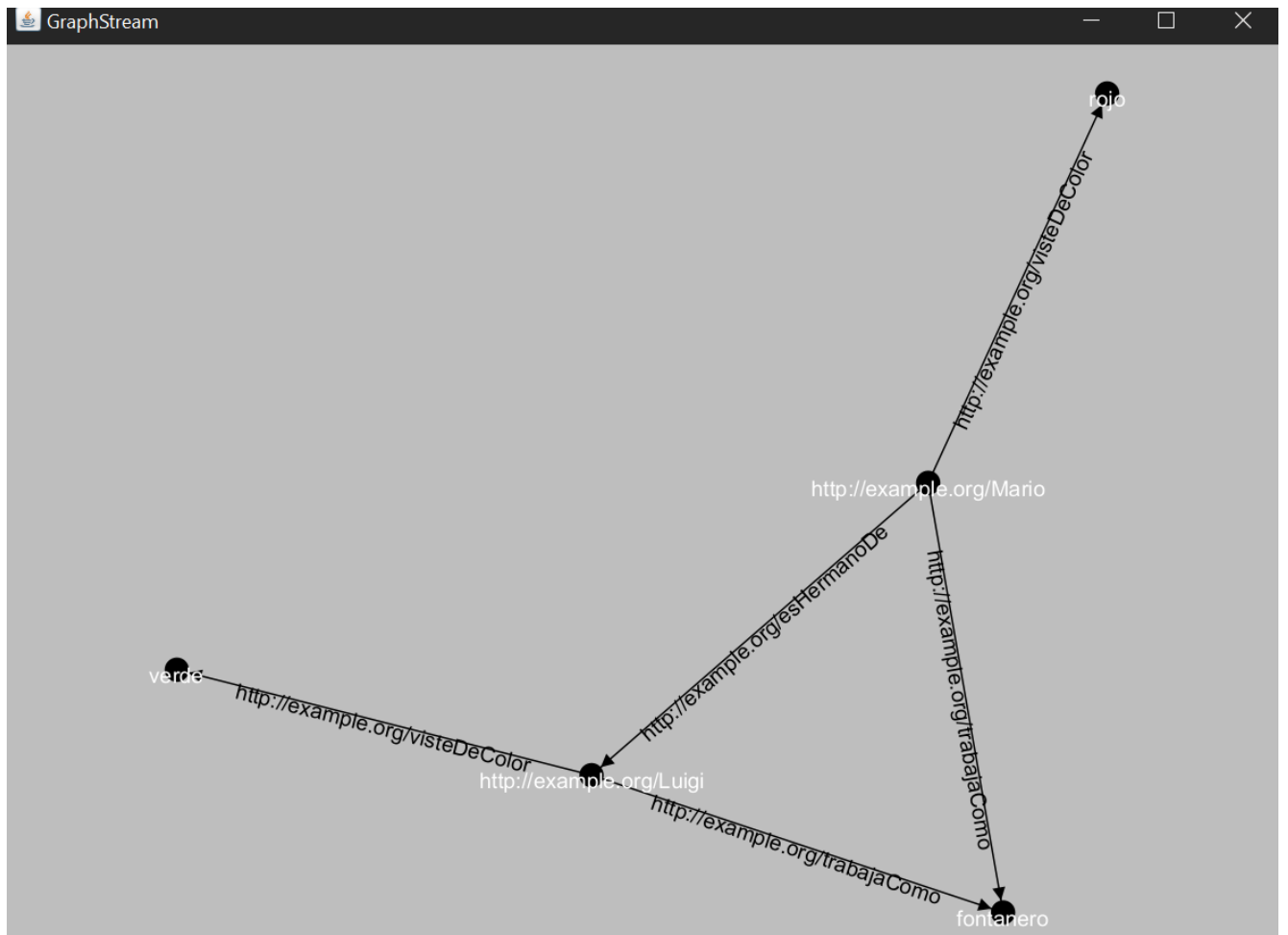


Ilustración 56 - Vista Graph Stream

Por último, se muestran los distintos grafos RDF obtenidos después de utilizar la aplicación para poblar la base de datos con datos verdaderos (ilustraciones 57-59).

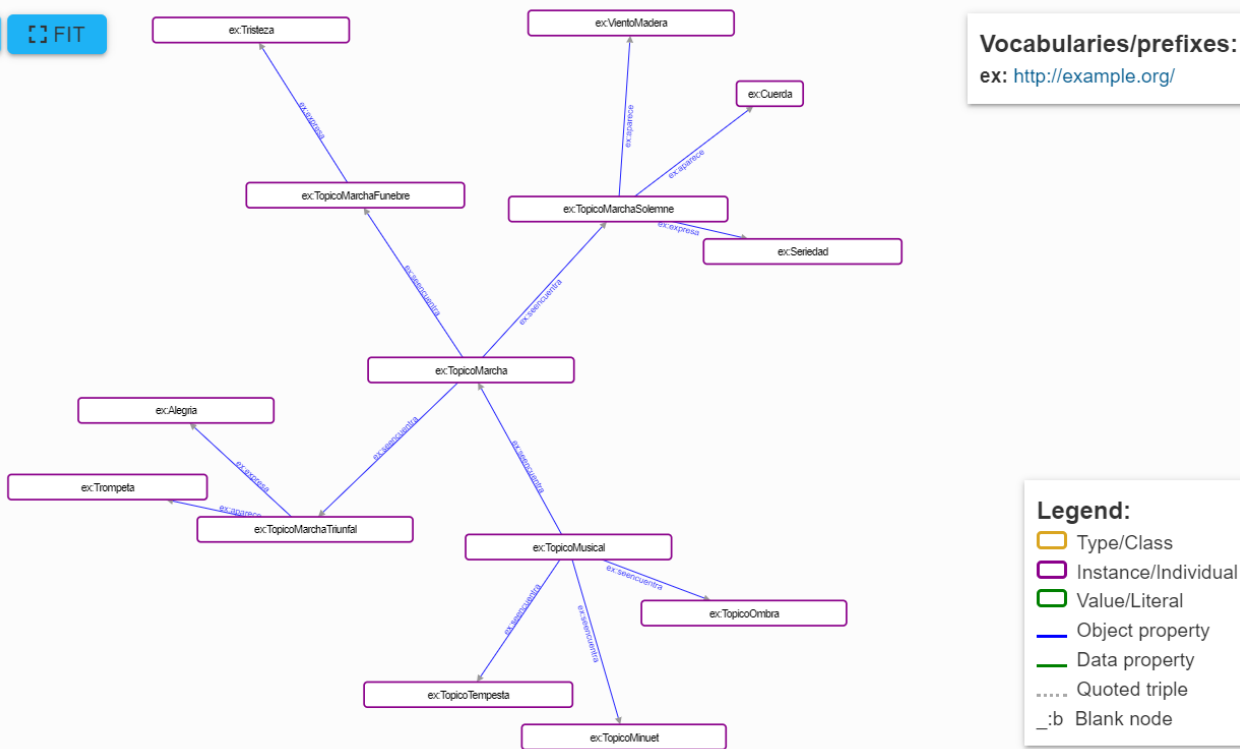


Ilustración 57 - Ejemplo grafo RDF con datos musicales (:isSemantic)

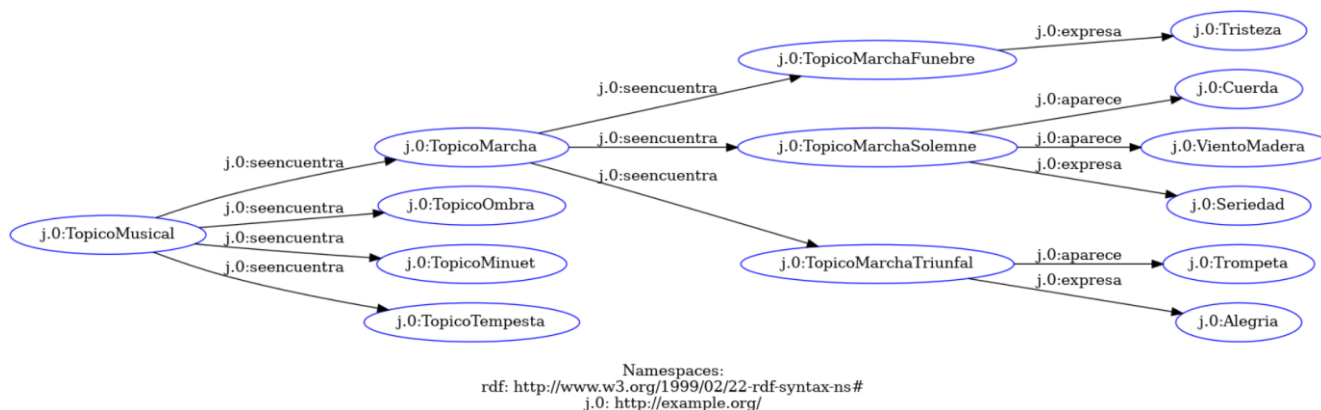


Ilustración 58 - Ejemplo grafo RDF con datos musicales (RDF Grapher)

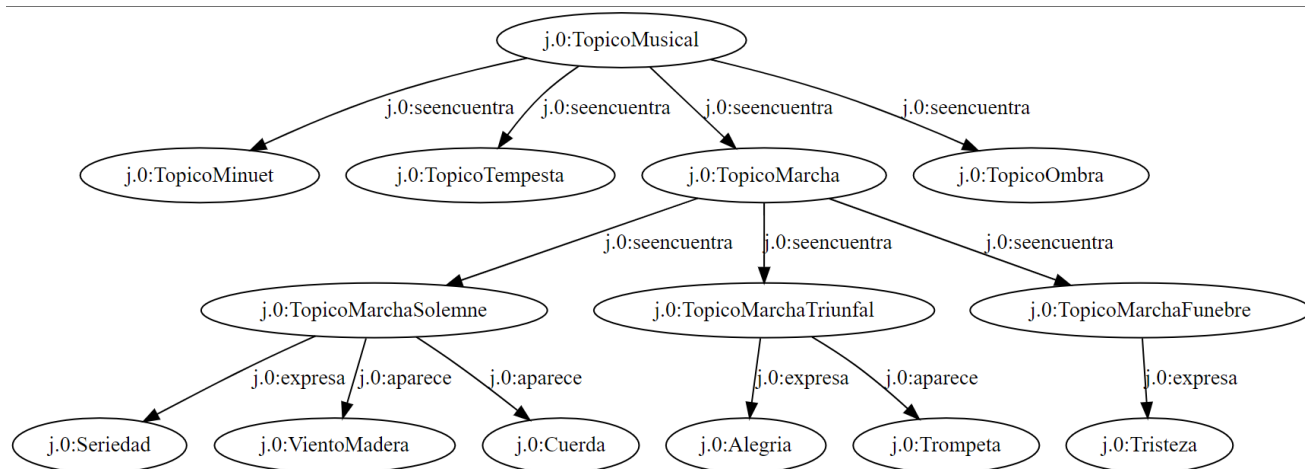


Ilustración 59 - Ejemplo grafo RDF con datos musicales (RDF Shape)

Capítulo 7 - Pruebas

7.1 Introducción

Con el objetivo de proporcionar información objetiva sobre la calidad del producto y sobre posibles errores, durante el desarrollo del proyecto se plantea la realización de pruebas utilizando diferentes métodos:

Pruebas de caja blanca

Las pruebas de caja blanca se refieren a un proceso de prueba en el que el probador tiene acceso completo a toda la información que hay detrás del software. Esto incluye el acceso al código fuente y a los documentos de diseño y las instrucciones para el cliente del proyecto.

Uno de los mejores momentos para utilizar las pruebas de caja blanca es en las tareas principalmente internas. Esto se refiere al desarrollo temprano de la parte funcional de la aplicación, con soluciones rápidas que son ideales, ya que no hay ningún beneficio en ofuscar el código cuando no se está simulando la experiencia del usuario.

Para ayudar con las pruebas de caja blanca se ha utilizado la herramienta de marcos de pruebas unitarias que automatizan la ejecución de casos de prueba en unidades individuales de código. Estas pruebas permiten mejorar la eficiencia de las pruebas de caja blanca, permitiendo a su vez identificar y solucionar problemas más rápidamente [59].

Pruebas de caja negra

Las pruebas de caja negra consisten en probar un sistema o programa informático sin tener conocimiento previo de su funcionamiento interno. Los probadores se limitan a dar entrada y recibir salida como lo haría un usuario final [60].

El objetivo de las pruebas de caja negra es conseguir que los usuarios interactúen con el software de una forma más natural de lo normal, sin tener ningún prejuicio existente derivado de conocer ya el software. Por ello, los responsables de realizar las pruebas son distintos de los que han desarrollado el software,

Este tipo de pruebas se han utilizado porque ayudan a revelar cualquier fallo que afecte a los flujos de trabajo reales del usuario.

7.2 Listado de pruebas

En esta sección se listan algunas de las pruebas que se han realizado con cada rol de usuario.

Listado para el Usuario (cualquier rol)

- El usuario puede iniciar sesión.
- El usuario puede cerrar sesión.
- El usuario puede salir de la aplicación con éxito.
- El usuario no puede iniciar sesión si las credenciales no son correctas.

Listado para el Administrador

- El administrador puede elegir entre administrar usuario o sugerencias.
- El administrador puede crear un usuario.
- El administrador puede modificar un usuario excepto su contraseña.
- El administrador puede eliminar un usuario.

- El administrador puede visualizar todos los usuarios de la base de datos.
- El administrador puede filtrar los usuarios del sistema por el rol.
- El administrador puede limpiar los campos de la vista.
- El administrador no puede crear un usuario si alguno de los campos está vacío.
- El administrador no puede crear un usuario ya existente.
- El administrador puede volver al menú.
- El administrador puede visualizar todas las sugerencias de la base de datos.
- El administrador puede filtrar las sugerencias por el autor de estas.
- El administrador puede quitar el filtro aplicado.
- El administrador puede eliminar una sugerencia.
- El administrador no puede eliminar una sugerencia si no la ha seleccionado antes.

Listado para el Lector

- El lector puede visualizar todas las tripletas validadas de la base de datos.
- El lector puede filtrar por objeto o predicado existentes las tripletas.
- El lector puede quitar el filtrado aplicado.
- El lector puede visualizar la tripleta seleccionada.
- El lector puede crear el fichero RDF con uno de los formatos disponibles.
- El lector puede visualizar la obra seleccionada de la tripleta a la que pertenece.
- El lector puede volver a la visualización de la tripleta estando en la visualización de la obra.
- El lector puede visualizar la bibliografía seleccionada de la tripleta a la que pertenece.
- El lector puede volver a la visualización de la tripleta estando en la visualización de la bibliografía.
- El lector puede volver a la visualización de las tripletas estando en la visualización de la tripleta.

Listado para el Colaborador

- El colaborador puede realizar todas las pruebas del rol lector.
- El colaborador puede visualizar la vista de creación de sugerencias.
- El colaborador puede crear una sugerencia con las condiciones indicadas.
- El colaborador puede visualizar la vista de creación de tripletas.
- El colaborador puede crear una tripleta si no existe.
- El colaborador no puede crear una tripleta si no están todos los campos completos.
- El colaborador puede agregar varias obras siempre que no existan.
- El colaborador no puede visualizar la vista de creación de bibliografía si no ha agregado al menos una obra.
- El colaborador puede agregar varias bibliografías siempre que no existan.
- El colaborador no puede volver la vista de creación de tripletas si no ha agregado al menos una bibliografía.

Listado para el Moderador

- El moderador puede realizar todas las pruebas del colaborador.
- El moderador puede visualizar el listado de tripletas no validadas.
- El moderador puede visualizar la tripleta no validada.
- El moderador puede validar la tripleta no validada seleccionada.
- El moderador puede crear/ver/modificar/eliminar objetos y predicados.
- El moderador puede ver/modificar/eliminar obras y bibliografías.
- El moderador puede ver/eliminar sugerencias.
- El moderador puede ver las fechas de las sugerencias con el formato día-mes-año.

- El moderador puede eliminar tripletas validadas.
- El moderador puede eliminar tripletas no validadas.

También se han realizado pruebas de instalación y despliegue de la aplicación en un dispositivo diferente de donde se ha desarrollado la aplicación.

7.3 Especificación listado de pruebas

Seguidamente, se muestra la especificación de cada prueba del listado de pruebas en forma de tabla donde aparecen los apartados:

- ID: identificador único del caso de prueba.
- Nombre: nombre del caso de prueba.
- Descripción: descripción del caso de prueba.
- Entrada: datos pasados o acción realizada por el usuario.
- Resultado esperado: resultado que se espera obtener al realizar la prueba.
- Resultado: si el resultado es correcto o no.
- Corrección: algunas de las correcciones que se han ido haciendo para que el resultado fuera el esperado.

| ID | CP-001 |
|--------------------|---|
| NOMBRE | Iniciar sesión |
| DESCRIPCIÓN | El usuario intenta iniciar sesión en el sistema después de introducir las credenciales correctas. |
| ENTRADA | Datos introducidos (usuario y contraseña) y accionamiento del botón "Enter". |
| RESULTADO ESPERADO | El usuario inicia sesión en el sistema. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 28 - CP001

| ID | CP-002 |
|--------------------|--|
| NOMBRE | Cerrar sesión |
| DESCRIPCIÓN | El usuario intenta cerrar sesión del sistema pulsando el botón "Cerrar sesión" |
| ENTRADA | Accionamiento del botón "Cerrar sesión". |
| RESULTADO ESPERADO | El usuario cierra sesión. |
| RESULTADO | Incorrecto con el rol moderador, no abre la vista de inicio de sesión. |
| CORRECCIÓN | Modificación en la máquina de estados. |

Tabla 29 - CP002

| ID | CP-003 |
|--------------------|---|
| NOMBRE | Salida exitosa |
| DESCRIPCIÓN | El programa termina con éxito. |
| ENTRADA | Estado de salida 0 y accionamiento del botón "Off". |
| RESULTADO ESPERADO | Se sale correctamente del programa. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 30 - CP003

| ID | CP-004 |
|--------------------|--|
| NOMBRE | Inicio de sesión incorrecto |
| DESCRIPCIÓN | El usuario no puede iniciar sesión si las credenciales son incorrectas. |
| ENTRADA | Posibles entradas: 1. Ningún campo completo. 2. Campo usuario vacío. 3. Campo contraseña vacío. 4. Las credenciales no corresponden. Accionamiento del botón "Enter". |
| RESULTADO ESPERADO | Aviso del error. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Después de añadir la encriptación de la contraseña, se "arregló" la distinción de mayúsculas y minúsculas. |

Tabla 31 - CP004

| ID | CP-005 |
|--------------------|--|
| NOMBRE | Elección menú administrador |
| DESCRIPCIÓN | El administrador puede elegir entre administrar usuario o sugerencias. Para ello debe poder pulsar los botones que te llevan a la administración de usuario o de sugerencia. |
| ENTRADA | |
| RESULTADO ESPERADO | Vista administración de usuarios o de sugerencias. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 32 - CP005

| ID | | CP-006 |
|---------------------------|---|---------------|
| NOMBRE | Crear usuario | |
| DESCRIPCIÓN | El administrador puede insertar un usuario en la base de datos del sistema siempre que este usuario no exista. La contraseña debe estar encriptada. | |
| ENTRADA | Atributos de la entidad usuario y accionamiento del botón “Crear”. | |
| RESULTADO ESPERADO | Creación en la base de datos del nuevo usuario y visualización de este en la tabla de usuarios. | |
| RESULTADO | Correcto. | |
| CORRECCIÓN | <ol style="list-style-type: none"> 1. Creación de restricciones para evitar usuarios duplicados. 2. Realizar actualización en la tabla de usuarios después de insertar el usuario en la base de datos. 3. Encriptación de la contraseña y paso del hash obtenido a la base de datos. | |

Tabla 33 - CP006

| ID | | CP-007 |
|---------------------------|--|---------------|
| NOMBRE | Modificar usuario | |
| DESCRIPCIÓN | El administrador puede modificar los atributos del usuario seleccionado excepto su contraseña. | |
| ENTRADA | Los nuevos datos y accionamiento del botón “Modificar”. | |
| RESULTADO ESPERADO | El usuario se modifica en la base de datos y se visualiza el cambio en la tabla. | |
| RESULTADO | Correcto. | |
| CORRECCIÓN | <ol style="list-style-type: none"> 1. Buscar el usuario correspondiente en la base de datos. 2. Evitar que la contraseña pueda ser modificada. | |

Tabla 34 - CP007

| ID | | CP-008 |
|---------------------------|--|---------------|
| NOMBRE | Eliminar usuario | |
| DESCRIPCIÓN | El administrador puede eliminar un usuario de la base de datos del sistema. | |
| ENTRADA | Selección del usuario a eliminar y accionamiento del botón “Eliminar”. | |
| RESULTADO ESPERADO | El usuario se elimina de la base de datos y se actualiza la tabla. | |
| RESULTADO | Correcto. | |
| CORRECCIÓN | <ol style="list-style-type: none"> 1. Buscar el usuario correspondiente en la base de datos. 2. Actualización de la tabla. | |

Tabla 35 - CP008

| ID | CP-009 |
|--------------------|--|
| NOMBRE | Visualizar usuarios |
| DESCRIPCIÓN | El administrador puede visualizar todos los usuarios de la base de datos del sistema. |
| ENTRADA | Accionamiento del botón “Usuarios” del menú. |
| RESULTADO ESPERADO | Visualización de los atributos de los usuarios existentes en la base de datos. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Carga de datos de la tabla usuarios de la base de datos en la tabla de la vista de administración de usuarios. |

Tabla 36 - CP009

| ID | CP-010 |
|--------------------|---|
| NOMBRE | Filtrar usuarios |
| DESCRIPCIÓN | El administrador puede filtrar los usuarios del sistema por el rol. |
| ENTRADA | El administrador selecciona una de las opciones existentes en el desplegable que se encarga del filtrado. |
| RESULTADO ESPERADO | Visualización de los atributos de los usuarios existentes en la base de datos con el filtro del rol aplicado. |
| RESULTADO | Correcto |
| CORRECCIÓN | <ol style="list-style-type: none"> 1. Visualización correcta en la tabla. 2. Acceso a los datos con un filtro aplicado. |

Tabla 37 - CP010

| ID | CP-011 |
|--------------------|--|
| NOMBRE | Limpiar campos |
| DESCRIPCIÓN | El administrador puede limpiar(vaciar) los campos de la vista. |
| ENTRADA | Accionamiento del botón “Limpiar”. |
| RESULTADO ESPERADO | Los campos de texto se limpian. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 38 - CP011

| ID | CP-012 |
|--------------------|--|
| NOMBRE | Creación usuario incompleta |
| DESCRIPCIÓN | El administrador no puede crear un usuario si alguno de los campos está vacío. |
| ENTRADA | Posibles entradas: <ol style="list-style-type: none"> 1. Ningún campo completo. 2. Campo usuario vacío. 3. Campo contraseña vacío. Accionamiento del botón “Crear”. |
| RESULTADO ESPERADO | Aviso del error. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Detección de los campos vacíos. |

Tabla 39 - CP012

| ID CP-013 | |
|--------------------|--|
| NOMBRE | Creación usuario existente |
| DESCRIPCIÓN | El administrador no puede crear un usuario ya existente. |
| ENTRADA | Posibles entradas: 1. Mismo usuario y rol pero distinta contraseña. 2. Mismo usuario y contraseña pero distinto rol. Accionamiento del botón "Crear". |
| RESULTADO ESPERADO | Aviso del error. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 40 - CP013

| ID CP-014 | |
|--------------------|--|
| NOMBRE | Regresar al menú |
| DESCRIPCIÓN | El administrador puede volver al menú. |
| ENTRADA | Accionamiento del botón "Atrás". |
| RESULTADO ESPERADO | Visualizar la vista del menú. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Ajustes en la máquina de estado. |

Tabla 41 - CP014

| ID CP-015 | |
|--------------------|--|
| NOMBRE | Visualizar sugerencias |
| DESCRIPCIÓN | El administrador puede visualizar todas las sugerencias de la base de datos. |
| ENTRADA | Accionamiento del botón "Sugerencias" del menú. |
| RESULTADO ESPERADO | Vista de la administración de sugerencias. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Ajustes en la máquina de estados. |

Tabla 42 - CP015

| ID CP-016 | |
|--------------------|--|
| NOMBRE | Filtrar sugerencias |
| DESCRIPCIÓN | El administrador puede filtrar las sugerencias por el autor de estas o por la fecha de creación. |
| ENTRADA | Elección de los filtros. |
| RESULTADO ESPERADO | Visualización de la tabla con los filtros aplicados. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Actualización de la tabla con los eventos de los desplegados. |

Tabla 43 - CP016

| ID CP-017 | |
|--------------------|---|
| NOMBRE | Quitar filtro |
| DESCRIPCIÓN | El administrador puede quitar el filtro aplicado. |
| ENTRADA | Accionamiento del botón “Quitar filtro”. |
| RESULTADO ESPERADO | Actualización de la tabla sugerencias. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 44 - CP017

| ID CP-018 | |
|--------------------|---|
| NOMBRE | Eliminar sugerencia |
| DESCRIPCIÓN | El administrador puede eliminar una sugerencia. |
| ENTRADA | Selección de la sugerencia a eliminar y accionamiento del botón “Eliminar”. |
| RESULTADO ESPERADO | Sugerencia eliminada de la base de datos y actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 45 - CP018

| ID CP-019 | |
|--------------------|---|
| NOMBRE | Eliminación sugerencia incorrecta |
| DESCRIPCIÓN | El administrador no puede eliminar una sugerencia si no la ha seleccionado antes. |
| ENTRADA | Accionamiento del botón “Eliminar”. |
| RESULTADO ESPERADO | Eliminación de la sugerencia de la base de datos y actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 46 - CP019

| ID CP-020 | |
|--------------------|--|
| NOMBRE | Visualizar tripletas validadas |
| DESCRIPCIÓN | El lector, colaborador o moderador puede visualizar todas las tripletas validadas de la base de datos. |
| ENTRADA | Accionamiento del botón “Enter”. |
| RESULTADO ESPERADO | Tabla de las tripletas validadas de la base de datos del sistema. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 47 - CP020

| ID | CP-021 |
|--------------------|--|
| NOMBRE | Filtrar tripletas |
| DESCRIPCIÓN | El lector puede filtrar por objeto o predicado existentes las tripletas. |
| ENTRADA | Selección del filtro a aplicar y accionamiento del botón “Filtrar”. |
| RESULTADO ESPERADO | Actualización de la tabla con el filtro aplicado. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 48 - CP021

| ID | CP-022 |
|--------------------|--|
| NOMBRE | Quitar filtro en tripletas |
| DESCRIPCIÓN | El lector puede quitar el filtrado aplicado. |
| ENTRADA | Accionamiento del botón “Quitar filtro”. |
| RESULTADO ESPERADO | Actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 49 - CP022

| ID | CP-023 |
|--------------------|--|
| NOMBRE | Visualizar tripleta seleccionada |
| DESCRIPCIÓN | El lector puede visualizar la tripleta seleccionada. |
| ENTRADA | Selección de la tripleta a ver y accionamiento del botón “Ver tripleta”. |
| RESULTADO ESPERADO | Ventana de detalles de tripleta. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 50 - CP023

| ID | CP-024 |
|--------------------|---|
| NOMBRE | Crear fichero RDF |
| DESCRIPCIÓN | El lector puede crear el fichero RDF con uno de los formatos disponibles. |
| ENTRADA | Selección del botón “Crear fichero RDF” y elección del formato. |
| RESULTADO ESPERADO | Fichero RDF con el formato elegido. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 51 - CP024

| ID | CP-025 |
|--------------------|--|
| NOMBRE | Visualizar obra seleccionada |
| DESCRIPCIÓN | El lector puede visualizar la obra seleccionada de la tripleta a la que pertenece. |
| ENTRADA | Selección de la obra a ver y accionamiento del botón “Info obra”. |
| RESULTADO ESPERADO | Vista detalles obra. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 52 - CP025

| ID CP-026 | |
|--------------------|--|
| NOMBRE | Regreso a detalles tripleta desde obra |
| DESCRIPCIÓN | El lector puede volver a la visualización de la tripleta estando en la visualización de la obra. |
| ENTRADA | Accionamiento del botón “Atrás”. |
| RESULTADO ESPERADO | Vista detalles de tripleta. |
| RESULTADO | Correcto |
| CORRECCIÓN | Ajustes máquina de estados. |

Tabla 53 - CP026

| ID CP-027 | |
|--------------------|--|
| NOMBRE | Visualizar bibliografía seleccionada |
| DESCRIPCIÓN | El lector puede visualizar la bibliografía seleccionada de la tripleta a la que pertenece. |
| ENTRADA | Selección de la bibliografía a ver y accionamiento del botón “Info bibliografía”. |
| RESULTADO ESPERADO | Vista detalles bibliografía. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Ajustes máquina de estados. |

Tabla 54 - CP027

| ID CP-028 | |
|--------------------|--|
| NOMBRE | Regreso a detalles tripleta desde bibliografía |
| DESCRIPCIÓN | El lector puede volver a la visualización de la tripleta estando en la visualización de la bibliografía. |
| ENTRADA | Accionamiento del botón “Atrás”. |
| RESULTADO ESPERADO | Vista detalles de tripleta. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Ajustes máquina de estados. |

Tabla 55 - CP028

| ID CP-029 | |
|--------------------|--|
| NOMBRE | Regreso al listado de tripletas desde detalles tripleta |
| DESCRIPCIÓN | El lector puede volver a la visualización de las tripletas estando en la visualización de la tripleta. |
| ENTRADA | Accionamiento del botón “Atrás”. |
| RESULTADO ESPERADO | Vista lista de tripletas. |
| RESULTADO | Correcto |
| CORRECCIÓN | Ajustes máquina de estados. |

Tabla 56 - CP029

| ID CP-030 | |
|--------------------|--|
| NOMBRE | Creación sugerencia |
| DESCRIPCIÓN | El colaborador puede crear una sugerencia con las condiciones indicadas. |
| ENTRADA | Accionamiento del botón “Crear”. |
| RESULTADO ESPERADO | Sugerencia agregada a la base de datos del sistema. |
| RESULTADO | Correcto. |
| CORRECCIÓN | Ajustes en las fechas de la sugerencia. |

Tabla 57 - CP030

| ID CP-031 | |
|--------------------|---|
| NOMBRE | Creación tripleta |
| DESCRIPCIÓN | El colaborador puede crear una tripleta si no existe. |
| ENTRADA | Accionamiento del botón “Siguiente”. |
| RESULTADO ESPERADO | Vista creación de obra. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 58 - CP031

| ID CP-032 | |
|--------------------|--|
| NOMBRE | Creación tripleta incompleta |
| DESCRIPCIÓN | El colaborador no puede crear una tripleta si no están todos los campos completos. |
| ENTRADA | Falta de datos en los campos. |
| RESULTADO ESPERADO | Aviso del error. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 59 - CP032

| ID CP-033 | |
|--------------------|---|
| NOMBRE | Agregar obra |
| DESCRIPCIÓN | El colaborador puede agregar varias obras siempre que no existan. |
| ENTRADA | Accionamiento del botón “Guardar”. |
| RESULTADO ESPERADO | Obra agregada a la base de datos. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 60 - CP033

| ID CP-034 | |
|--------------------|--|
| NOMBRE | Continuar al siguiente paso incorrecto |
| DESCRIPCIÓN | El colaborador no puede visualizar la vista de creación de bibliografía si no ha agregado al menos una obra. |
| ENTRADA | Accionamiento del botón “Siguiente”. |
| RESULTADO ESPERADO | Aviso del error. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 61 - CP034

| ID | CP-035 |
|--------------------|---|
| NOMBRE | Agregar bibliografía |
| DESCRIPCIÓN | El colaborador puede agregar varias bibliografías siempre que no existan. |
| ENTRADA | Accionamiento del botón “Guardar”. |
| RESULTADO ESPERADO | Bibliografía agregada a la base de datos. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 62 - CP035

| ID | CP-036 |
|--------------------|---|
| NOMBRE | Visualización tripletas no validadas |
| DESCRIPCIÓN | El moderador puede visualizar el listado de tripletas no validadas. |
| ENTRADA | Accionamiento del botón “Validación” |
| RESULTADO ESPERADO | Ver tabla de las tripletas no validadas asignadas al usuario. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 63 - CP036

| ID | CP-037 |
|--------------------|--|
| NOMBRE | Ver tripleta no validada |
| DESCRIPCIÓN | El moderador puede visualizar la tripleta no validada. |
| ENTRADA | Accionamiento del botón “Ver tripleta”. |
| RESULTADO ESPERADO | Vista detalles tripleta. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 64 - CP037

| ID | CP-038 |
|--------------------|--|
| NOMBRE | Validar tripleta |
| DESCRIPCIÓN | El moderador puede validar la tripleta no validada seleccionada. |
| ENTRADA | Accionamiento del botón “Validar” |
| RESULTADO ESPERADO | Modificación de la matrícula en la base de datos. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 65 - CP038

| ID | CP-039 |
|--------------------|--|
| NOMBRE | CRUD objetos |
| DESCRIPCIÓN | El moderador puede crear/modificar/eliminar objetos. |
| ENTRADA | Accionamiento de los diferentes botones. |
| RESULTADO ESPERADO | Acción a realizar en la base de datos y actualización de la tabla. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 66 - CP039

| ID | CP-040 |
|--------------------|--|
| NOMBRE | CRUD predicados |
| DESCRIPCIÓN | El moderador puede crear/modificar/eliminar predicados. |
| ENTRADA | Accionamiento de los diferentes botones. |
| RESULTADO ESPERADO | Acción a realizar en la base de datos y actualización de la tabla. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 67 - CP040

| ID | CP-041 |
|--------------------|--|
| NOMBRE | RUD obras |
| DESCRIPCIÓN | El moderador puede modificar/eliminar obras. |
| ENTRADA | Accionamiento de los diferentes botones. |
| RESULTADO ESPERADO | Acción a realizar en la base de datos y actualización de la tabla. |
| RESULTADO | Incorrecto. |
| CORRECCIÓN | No se ha corregido. Al dar a modificar si se visualiza en la tabla la modificación pero no se guarda bien el cambio en la base de datos. |

Tabla 68 - CP041

| ID | CP-042 |
|--------------------|--|
| NOMBRE | RUD bibliografía |
| DESCRIPCIÓN | El moderador puede modificar/eliminar bibliografías. |
| ENTRADA | Accionamiento de los diferentes botones. |
| RESULTADO ESPERADO | Acción a realizar en la base de datos y actualización de la tabla. |
| RESULTADO | Correcto |
| CORRECCIÓN | |

Tabla 69 - CP042

| ID | CP-043 |
|--------------------|--|
| NOMBRE | RD sugerencias |
| DESCRIPCIÓN | El moderador puede ver/eliminar sugerencias. |
| ENTRADA | Accionamiento de los diferentes botones. |
| RESULTADO ESPERADO | Acción a realizar en la base de datos y actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 70 - CP043

| ID | CP-044 |
|--------------------|---|
| NOMBRE | Descartar tripleta |
| DESCRIPCIÓN | El moderador puede eliminar tripletas no validadas. |
| ENTRADA | Accionamiento del botón "Eliminar". |
| RESULTADO ESPERADO | Eliminación de la tripleta de la base de datos y actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 71 - CP044

| ID CP-045 | |
|--------------------|---|
| NOMBRE | Eliminar tripleta |
| DESCRIPCIÓN | El moderador puede eliminar tripletas validadas. |
| ENTRADA | Accionamiento del botón “Eliminar”. |
| RESULTADO ESPERADO | Eliminación de la tripleta de la base de datos y actualización de la tabla. |
| RESULTADO | Correcto. |
| CORRECCIÓN | |

Tabla 72 - CP045

| ID CP-046 | |
|--------------------|---|
| NOMBRE | Despliegue aplicación |
| DESCRIPCIÓN | El usuario puede ejecutar en la herramienta NetBeans la aplicación obtenida del repositorio de la autora. |
| ENTRADA | Accionamiento de “Ejecutar”. |
| RESULTADO ESPERADO | Despliegue de la aplicación |
| RESULTADO | Correcto. |
| CORRECCIÓN | Antes de ejecutar hay que seleccionar “Limpiar y Generar”. |

Tabla 73 - CP- 046

Capítulo 8 - Conclusiones y líneas futuras

8.1 Conclusiones

Desde una perspectiva personal, la realización del TFG ha supuesto experimentar en primera persona todo el proceso de planificación y desarrollo de un proyecto, lidiar con las dificultades y riesgos que han surgido a lo largo de la duración del proyecto y tener conciencia sobre los diferentes papeles existentes en cada fase: jefe de proyecto, analista, desarrollador, etc. También, ha implicado la adquisición de nuevos conocimientos sobre RDF y todo lo relacionado con él (grafos de conocimiento, web semántica, ontologías, etc) y la aplicación y afianzamiento de los conocimientos obtenidos en las asignaturas cursadas en la mención, siendo las más implicadas Panificación y gestión de plataformas informáticas, Fundamentos de software, Diseño, integración y adaptación de software, Interacción persona-computadora. Fundamentos de programación, Programación orientada a objetos, Tecnología y diseño de bases de datos y Administración de bases de datos.

En relación con la investigación realizada, me ha permitido aprender sobre los grafos de conocimiento y RDF: qué son, las posibilidades que ofrecen y las investigaciones y aplicaciones que se están realizando en diferentes ámbitos.

Finalmente, como resultado de la realización del TFG se ha obtenido el presente documento y una infraestructura para una red de tópicos musicales.

8.2 Líneas futuras

Se presentan las posibles líneas futuras para mejorar la funcionalidad del proyecto actual. Estas ideas se basan en los resultados y conclusiones obtenidos hasta el momento, así como en las necesidades y oportunidades identificadas durante la ejecución del proyecto.

A continuación, se presentan algunas:

- Agregar iconos representativos de la acción en algunos botones.
- Agregar algunas comprobaciones de los datos introducidos, por ejemplo el atributo año.
- Agregar obras o bibliografía a tripletas validadas.
- Cambiar el estilo de fondo haciendo uso de colores para que sea más atractiva visualmente.
- Configurar los tamaños de las distintas vistas para que se auto ajusten.
- Agregar captación de eventos al presionar una tecla.
- Agregar límite en los campos en los que se introduce texto.
- Agregar opción de cambio de contraseña a los usuarios.
- Agregar restricciones para crear contraseñas más robustas (número de caracteres, distinción entre mayúsculas y minúsculas, etc).
- Agregar visualización de grafos directamente desde la aplicación, por ejemplo usando la librería Graph Stream.
- Cambiar la base de datos local por una remota o una específica para almacenar tripletas como GraphDB o neo4j.

Bibliografía

- [1] M. D.-E. Águeda Pedrero-Encabo, «DE LA EMOCIÓN AL SENTIDO: APLICACIÓN DOCENTE DE LA TEORÍA DE TÓPICOS,» [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/60273/HUMAN-M32-A11.pdf?sequence=1>. [Último acceso: 2 Julio 2024].
- [2] UVA, «De la emoción al sentido,» Universidad de Valladolid Departamento de Didáctica de la Expresión Musical, Plástica y Corporal, [En línea]. Available: <https://topicos.uva.es/>. [Último acceso: 2 julio 2024].
- [3] E. e. d. IONOS, «El modelo en cascada: desarrollo secuencial de software,» IONOS, 11 marzo 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>. [Último acceso: 2 julio 2024].
- [4] Deloitte, «¿Cuál es la metodología más adecuada para tu proyecto?,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>. [Último acceso: 2 julio 2024].
- [5] B. H. a. M. Cotterellh, Software Project Management, M. Companies, Ed., 1968.
- [6] c. d. Wikipedia, «Estructura de descomposición del trabajo,» Wikipedia, La enciclopedia libre, 25 marzo 2024. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Estructura_de_descomposici%C3%B3n_del_trabajo&oldid=159018546. [Último acceso: 1 julio 2024].
- [7] W. contributors, «Product-based planning,» Wikipedia, The Free Encyclopedia, 28 diciembre 2023. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Product-based_planning&oldid=1192223635. [Último acceso: 1 julio 2024].
- [8] W. contributors, «Product breakdown structure,» Wikipedia, The Free Encyclopedia, 8 julio 2023. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Product_breakdown_structure&oldid=1164248494. [Último acceso: 1 julio 2024].
- [9] K. D. Davis, «What is Product Breakdown Structure (PBS) in Project Management?,» upGrad, 19 febrero 2024. [En línea]. Available: <https://www.knowledgehut.com/blog/project-management/product-breakdown-structure-in-project-management#advantages-and-disadvantages-of-product-breakdown-structure%C2%A0>. [Último acceso: 2 julio 2024].
- [10] S. Pursell, «Diagrama de PERT: qué es, cómo hacerlo y ejemplos,» HubSpot, 20 enero 2023. [En línea]. Available: <https://blog.hubspot.es/marketing/diagrama-pert>. [Último acceso: 2 julio 2024].
- [11] T. Asana, «El diagrama de PERT: qué es y cómo crearlo,» Asana, 7 febrero 2024. [En línea]. Available: <https://asana.com/es/resources/pert-chart>. [Último acceso: 1 julio 2024].
- [12] D. Santos, «Diagrama de Gantt: qué es y cómo crearlo paso a paso,» HubSpot, 19 marzo 2024. [En línea]. Available: <https://blog.hubspot.es/marketing/crear-diagrama-gantt>. [Último acceso: 2 julio 2024].

- [13] D. Ramos, «Ventajas y limitaciones de los diagramas de Gantt en la gestión de proyectos,» Smartsheet, 3 mayo 2021. [En línea]. Available: <https://es.smartsheet.com/content/gantt-chart-pros-cons>. [Último acceso: 2 julio 2024].
- [14] Microsoft, «Encontrar el plan de Microsoft 365 Enterprise más adecuado para la organización,» Microsoft, [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/enterprise/microsoft365-plans-and-pricing>. [Último acceso: 1 julio 2024].
- [15] V. Paradigm, «Visual Paradigm Pricing,» Visual Paradigm, [En línea]. Available: <https://www.visual-paradigm.com/shop/vp.jsp?license=perpetual>. [Último acceso: 1 julio 2024].
- [16] Talent, «Salario medio para Programador Junior en España, 2024,» 2024. [En línea]. Available: <https://es.talent.com/salary?job=programador#:~:text=%C2%BFcu%C3%A1nto%20gana%20un%20Programador%20en%20Espa%C3%B1a%3F&text=El%20salario%20programador%20promedio%20en,hasta%20%E2%82%AC%2040.000%20al%20a%C3%B1o..> [Último acceso: 2 julio 2024].
- [17] X. s. Martín, «MATRIZ DE RIESGOS: GUÍA COMPLETA PARA LA GESTIÓN EMPRESARIAL,» Cloud, Ao Data, 8 enero 2024. [En línea]. Available: <https://aodatacloud.es/blog/que-es-y-como-hacer-la-matriz-de-riesgos/>. [Último acceso: 1 julio 2024].
- [18] W. contributors, «Risk matrix,» Wikipedia, The Free Encyclopedia, 9 mayo 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Risk_matrix&oldid=1222967219. [Último acceso: 1 julio 2024].
- [19] R. W. Group, «Resource Description Framework (RDF),» W3C, 25 febrero 2014. [En línea]. Available: <https://www.w3.org/RDF/>. [Último acceso: 2 julio 2024].
- [20] ontotext, «What is RDF?,» [En línea]. Available: <https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf/>. [Último acceso: 2 julio 2024].
- [21] c. d. Wikipedia, «Tripleta semántica,» Wikipedia, La enciclopedia libre, 11 julio 2019. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Tripleta_sem%C3%A1ntica&oldid=117335384. [Último acceso: 1 julio 2024].
- [22] W. contributors, «Resource description framework,» Wikipedia, The Free Encyclopedia, 25 abril 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Resource_Description_Framework&oldid=1220688807. [Último acceso: 1 julio 2024].
- [23] G. Vasco, «RDF (Resource Description Framework),» [En línea]. Available: https://www.euskadi.eus/contenidos/informacion/opendata_rdf_euskadi/es_info/adjuntos/RDF.pdf. [Último acceso: 2 julio 2024].
- [24] c. d. Wikipedia, «Identificador de recursos uniforme,» Wikipedia, La enciclopedia libre, 12 abril 2024. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Identificador_de_recursos_uniforme&oldid=159398558. [Último acceso: 1 julio 2024].
- [25] B. S. a. D. A. Sean Martin, «The Rise of the Knowledge Graph. Toward Modern Data Integration and the Data Fabric Architecture,» 19 Marzo 2021. [En línea]. Available:

- https://info.cambridgesemantics.com/hubfs/The_Rise_of_the_Knowledge_Graph.pdf. [Último acceso: 3 Julio 2024].
- [26] ontotext, «What is a Knowledge Graph?,» [En línea]. Available: <https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/>. [Último acceso: 3 Julio 2024].
- [27] F. X. M. N. F. O. Ciyuan Peng, «Knowledge Graphs: Opportunities and Challenges,» ResearchGate, [En línea]. Available: https://www.researchgate.net/publication/369540824_Knowledge_Graphs_Opportunities_and_Challenges. [Último acceso: 3 Julio 2024].
- [28] A. K. Pandey, «What is knowledge graph?,» AtulHost, 18 Diciembre 2020. [En línea]. Available: <https://www.atulhost.com/what-is-knowledge-graph>. [Último acceso: 3 Julio 2024].
- [29] W3C, «RDF Data Visualization,» septiembre 2018. [En línea]. Available: <https://www.w3.org/2018/09/rdf-data-viz/>. [Último acceso: 2 julio 2024].
- [30] Protégé, «Protégé,» [En línea]. Available: <https://protege.stanford.edu/>. [Último acceso: 2 julio 2024].
- [31] Protégé, «Protége visualization,» 27 marzo 2007. [En línea]. Available: <https://protegewiki.stanford.edu/wiki/Visualization>. [Último acceso: 2 julio 2024].
- [32] isSemantic, «Visualize RDF graph linked data as a connected diagram,» [En línea]. Available: <https://issemantic.net/rdf-visualizer>. [Último acceso: 1 julio 2024].
- [33] «RDF Grapher,» [En línea]. Available: <https://www.ldf.fi/service/rdf-grapher>. [Último acceso: 1 julio 2024].
- [34] RDFShape, «Data analysis,» [En línea]. Available: <https://rdfshape.weso.es/dataInfo>. [Último acceso: 2 julio 2024].
- [35] T. B.-L. W. E. P. G. C. L. M. I. David Beckett, «RDF 1.1 Turtle,» W3C, 25 febrero 2014. [En línea]. Available: <https://www.w3.org/TR/turtle/>. [Último acceso: 2 julio 2024].
- [36] W. W. Draft, «RDF 1.2 Concepts and Abstract Syntax,» W3C, 27 junio 2024. [En línea]. Available: <https://www.w3.org/TR/rdf12-concepts/>. [Último acceso: 2 julio 2024].
- [37] E. e. d. IONOS, «UML, lenguaje de modelado gráfico,» IONOS, 26 octubre 2018. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>. [Último acceso: 2 julio 2024].
- [38] Academia.edu, «UML gota a gota,» [En línea]. Available: https://www.academia.edu/29502209/UML_gota_a_gota. [Último acceso: 1 julio 2024].
- [39] D. G. Álvarez, *Tema 2: Especificación y análisis de requisitos del software. Parte II: Modelo de Análisis - Clases de Análisis*, 2021-2022.
- [40] W. contributors, «Relational database,» Wikipedia, The Free Encyclopedia, 6 abril 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Relational_database&oldid=1217554613. [Último acceso: 1 julio 2024].

- [41] T. a. B. C. Connolly, *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*, 4ª edición ed., P. E. S.A., Ed., 2005.
- [42] M. Martínez, *Tema 1 - Introducción a los sistemas de bases de dato*, 2021-2022.
- [43] c. d. Wikipedia, «Java (lenguaje de programación),» Wikipedia, La enciclopedia libre., 3 Julio 2024. [En línea]. Available: [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=161108422](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=161108422). [Último acceso: 3 Julio 2024].
- [44] c. d. Wikipedia, «Programación orientada a objetos,» Wikipedia, La enciclopedia libre., 2 Julio 2024. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_orientada_a_objetos&oldid=161080073. [Último acceso: 3 Julio 2024].
- [45] W. contributors, «Use Case,» Wikipedia, The Free Encyclopedia, 19 abril 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Use_case&oldid=1219721783. [Último acceso: 1 julio 2024].
- [46] E. e. d. IONOS, «El diagrama de casos de uso en UML,» IONOS, 24 julio 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/#:~:text=El%20diagrama%20de%20casos%20de%20uso%20es%20una%20forma%20de,de%20programaci%C3%B3n%20orientada%20a%20objetos..> [Último acceso: 2 julio 2024].
- [47] V. paradigm, «Casos De Uso “Incluir” Y “Extender”,» 9 febrero 2022. [En línea]. Available: <https://blog.visual-paradigm.com/es/include-and-extend-use-cases/>. [Último acceso: 2 julio 2024].
- [48] c. d. Wikipedia, «Arquitectura de software,» Wikipedia, La enciclopedia libre, 16 mayo 2024. [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Arquitectura_de_software&oldid=160156007. [Último acceso: 1 julio 2024].
- [49] c. d. Wikipedia, «Modelo-vista-presentador,» Wikipedia, La enciclopedia libre, 28 abril 2024. [En línea]. Available: <https://es.wikipedia.org/w/index.php?title=Modelo%E2%80%93vista%E2%80%93presentador&oldid=159762254>. [Último acceso: 1 julio 2024].
- [50] c. d. Wikipedia, «Cliente-servidor,» Wikipedia, La enciclopedia libre., 17 Junio 2024. [En línea]. Available: <https://es.wikipedia.org/w/index.php?title=Cliente-servidor&oldid=160801785>. [Último acceso: 3 Julio 2024].
- [51] AppMaster.io, «Patrones arquitectónicos: explicación de MVC, MVP y MVVM,» 28 agosto 2023. [En línea]. Available: <https://appmaster.io/es/blog/patrones-arquitectonicos-mvc-mvp-y-mvvm>. [Último acceso: 1 julio 2024].
- [52] A. M. M. Mario Corrales Astorgano, *Introducción al diseño de interfaces gráficas de usuario con JSwing*, 2023.
- [53] E. e. d. IONOS, «Diagrama de estado UML: visualizar secuencias de estados de objetos,» IONOS, 20 julio 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-estado-uml/>. [Último acceso: 2 julio 2024].
- [54] E. e. d. IONOS, «Diagramas de secuencia: mostrar interacciones con UML,» IONOS, 4 marzo 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de->

- [67] c. d. Wikipedia, «Llave candidata,» [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Llave_candidata&oldid=159811504. [Último acceso: 2 Julio 2024].
- [68] Santander, «¿Qué es un stakeholder y cuál es su función?,» [En línea]. Available: <https://www.impulsa-empresa.es/diccionario/stakeholder/>. [Último acceso: 2 Julio 2024].
- [69] c. d. Wikipedia, «Modelo computacional,» [En línea]. Available: https://es.wikipedia.org/w/index.php?title=Modelo_computacional&oldid=129029737. [Último acceso: 2 Julio 2024].
- [70] E. Enkel, The Palgrave Encyclopedia of Strategic Management, M. a. T. D. J. Augier, Ed., Palgrave Macmillan UK, 2018.
- [71] W. contributors, «Knowledge graph,» Wikipedia, The Free Encyclopedia., 15 Junio 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Knowledge_graph&oldid=1229285349. [Último acceso: 3 Julio 2024].
- [72] concepto, «Software de aplicación,» concepto, [En línea]. Available: <https://concepto.de/software-de-aplicacion/>. [Último acceso: 3 Julio 2024].
- [73] c. d. Wikipedia, «Musicología,» Wikipedia, La enciclopedia libre., 15 Mayo 2024. [En línea]. Available: <https://es.wikipedia.org/w/index.php?title=Musicolog%C3%ADa&oldid=160132880>. [Último acceso: 3 Julio 2024].
- [74] W. contributors, «Foreign key,» Wikipedia, The Free Encyclopedia, 14 abril 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Foreign_key&oldid=1218954322. [Último acceso: 1 julio 2024].
- [75] W. contributors, «Rapid application development,» Wikipedia, The Free Encyclopedia, 29 junio 2024. [En línea]. Available: https://en.wikipedia.org/w/index.php?title=Rapid_application_development&oldid=1231686099. [Último acceso: 1 julio 2024].
- [76] U. d. Valladolid, «Universidad de Valladolid Repositorio Documental,» Universidad de Valladolid, [En línea]. Available: <https://uvadoc.uva.es/handle/10324/852>. [Último acceso: 1 julio 2024].
- [77] M. 365, «Buscar el plan de Microsoft 365 más adecuado para tu empresa,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/business/compare-all-microsoft-365-business-products>. [Último acceso: 1 julio 2024].
- [78] J. C. A. Domínguez, «Trabajos Fin de Grado UVa,» Universidad de Valladolid Escuela de Ingeniería Informática de Valladolid, 2019. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/38682>. [Último acceso: 2 julio 2024].
- [79] P. P. Corral, «Trabajos Fin de Grado UVa,» Universidad de Valladolid Escuela de Ingeniería Informática de Valladolid, 2016. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/16952>. [Último acceso: 2 julio 2024].
- [80] M. C. García, «TFGMiguelChaveinte,» 2023. [En línea]. Available: <https://github.com/miguelchaveinte/TFGMiguelChaveinte/blob/master/tfgPresentaci%C3%B3n.pdf>. [Último acceso: 2 julio 2024].

- [81] E. e. d. IONOS, «¿Qué es la programación orientada a objetos (OOP)?», IONOS, 6 octubre 2023. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-orientada-a-objetos-oop/>. [Último acceso: 2 julio 2024].
- [82] MySQL, «Manual MySQL Workbench», [En línea]. Available: <https://dev.mysql.com/doc/workbench/en/>. [Último acceso: 2 julio 2024].
- [83] UVa, «DE LA EMOCIÓN AL SENTIDO: APLICACIÓN DOCENTE DE LA TEORÍA DE TÓPICOS EN MUSICOLOGÍA», 17 diciembre 2022. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/60273/HUMAN-M32-A11.pdf?sequence=1>. [Último acceso: 2 julio 2024].
- [84] M. T. Vazquez, «Trabajos Fin de Grado UVa», Universidad de Valladolid Escuela de Ingeniería Informática de Valladolid, 2023. [En línea]. Available: <https://uvadoc.uva.es/handle/10324/63010>. [Último acceso: 2 julio 2024].

- Tópicos musicales

Los tópicos musicales son temas del discurso musical que transmiten significados y emociones reconocibles de nuestra cultura y comparten los mismos códigos de ritmo, dinámica, tempo, armonía, timbre, ...(significante) en las obras de los compositores de Música Clásica occidental desde siglo XVIII. [1]

- World Wide Web

Red informática mundial que forma un sistema cuyo funcionamiento se realiza a través de internet, por el cual se pueden transmitir diversos tipos de datos a través del Protocolo de Transferencia de Hipertextos o HTTP, que son los enlaces de la página web. [61]

- Gestión del conocimiento

Proceso que continuamente asegura el desarrollo y la aplicación de todo tipo de conocimientos pertinentes de una empresa con objeto de mejorar su capacidad de resolución de problemas y así contribuir a la sostenibilidad de sus ventajas competitivas. [62]

- Web semántica

Extensión de la World Wide Web en la que el significado, es decir, la semántica, de la información y de los servicios está definido de forma correcta, lo que permite entender, en cierta manera, y satisfacer las peticiones de las personas y las máquinas que utilizan el contenido web. [63]

- Multigrafo dirigido

En teoría de grafos, es una generalización de un grafo que permite aristas múltiples, las cuales tienen un sentido definido. [64] [65]

- Modelos relacionales u ontológicos

En ciencia de la computación, es un modelo para describir el mundo que consiste en un conjunto de tipos, propiedades y relaciones entre tipos. [66]

- Clave candidata

En el modelo relacional de bases de datos, una llave candidata (o clave candidata) de una relación es una mínima súper llave de esa relación; es decir, un conjunto de atributos tales que:

1. La relación no tiene dos distintas tuplas (es decir, filas o registros en el lenguaje de base de datos común) con los mismos valores para estos atributos (lo que significa que el conjunto de atributos es una súper llave).

2. No hay un subconjunto propio de estos atributos para los que se cumple la condición anterior (lo que significa que el conjunto es mínimo). [67]

- Stakeholders

Personas, grupos o entidades que constituyen una parte interesada en una organización, proyecto o empresa y que pueden verse afectados por las acciones o decisiones tomadas a su respecto. [68]

- Modelo matemático de computación

En las ciencias de la computación, se trata de un modelo que requiere de extensos recursos computacionales para estudiar el comportamiento de un sistema complejo por medio de la simulación por computadora. [69]

- Red de Conocimiento

Grupo de gente, entidades u organizaciones que captan, comparten conocimientos existentes y/o crean nuevos. Puede hacerse, por ejemplo, recopilando los documentos pertinentes y resumirlos para extraer conocimientos nuevos; transfiriendo conocimiento implícito en nuevo conocimiento implícito; mediante la narración de historias o el uso de metáforas; o codificando el conocimiento en documentos para distribuirlos electrónicamente. Está influenciada por su entorno, como el sistema de gestión en el que actúa y la cultura que la rodea; y por su procesamiento del conocimiento como captar, compartir y crear. Está respaldada por herramientas como herramientas de información y comunicación, así como por tiempo y salas de reuniones [70].

- Grafo de Conocimiento

En la representación y razonamiento del conocimiento, se trata del conocimiento base que utiliza un modelo o topología de datos estructurados en grafos para representar y operar con datos. Se utiliza a menudo para almacenar descripciones interconectadas de entidades (objetos, eventos, situaciones o conceptos abstractos) y al mismo tiempo codifican la semántica de forma libre o las relaciones subyacentes a estas entidades [71].

- Software de Aplicación

Tipo específico de programas, cuya función es permitir llevar a cabo tareas de distinto tipo y finalidad, como pueden ser trabajos, cálculos, navegación web y cualquier otro fin distinto al mantenimiento del sistema [72].

- Musicología

Es el estudio científico o académico de todos los fenómenos relacionados con la música, como sus bases físicas, su historia y su relación con el ser humano y la sociedad [73].

A1. Manual de usuario

Al iniciar la aplicación lo primero que verá es la pantalla de inicio de sesión donde debe ingresar las credenciales aportadas (usuario y contraseña) en los campos indicados. En esta pantalla aparecen dos botones cuyo nombre indica la acción que realizan. Una vez ingresados las credenciales en los campos, si son correctas le llevará a la vista correspondiente con su rol asignado.

Si se trata del rol administrador le aparecerá el menú de administración donde podrá elegir entre la gestión de usuarios, la gestión de sugerencias o cerrar sesión. Si elige la de usuario le abrirá una pantalla donde podrá ver una tabla con los datos de los usuarios existentes, los campos de los datos para poder crear o modificar los datos, una barra desplegable para poder aplicar un filtro a la tabla y los botones correspondientes a la acción que indican. Si elige la de sugerencias le abrirá una pantalla donde podrá ver una tabla con los datos de las sugerencias existentes, dos desplegables para aplicar filtros sobre la tabla, un campo para visualizar la sugerencia y un botón para eliminar. En ambas pantallas aparece un botón que le permite regresar al menú.

Si se trata del rol lector le aparecerá la tabla de tripletas existentes. En esta pantalla puede aplicar filtrados con los desplegables y dirigirse a otra pantalla para visualizar los detalles de la tripleta seleccionada (pulsando el botón “Ver tripleta”). Desde la nueva ventana (detalles tripleta) puede dirigirse o a la ventana de obra o a la de bibliografía, para ver los detalles de la fila seleccionada. Siempre que abra una nueva ventana, podrá regresar a su predecesora.

Si se trata del rol colaborador le aparecerá la tabla de tripletas existentes. En esta, además de poder realizar lo mismo que el rol lector, puede dirigirse o a la pantalla de creación de tripletas o a la de sugerencias. En la de tripletas, le aparecerán los campos para formar la tripleta, para agregar una descripción y para asignarla a un moderador (se encargará de validarla). Una vez que comience con la creación (tres pasos) no podrá regresar hasta que se finalice: El primer paso es crear la tripleta, el segundo es agregarla al menos una obra y el tercero es agregarla al menos una bibliografía. Si todos los pasos se realizan correctamente la tripleta se habrá creado y volverá a la pantalla de creación de tripletas.

Si se trata del rol moderador le aparecerá la tabla de tripletas existentes. En esta, además de poder realizar lo mismo que el rol colaborador, puede ir a la gestión de sugerencias, ir a la gestión de tripletas no validadas o ir a un menú de gestión. En la ventana de gestión de sugerencias podrá realizar lo comentado en el rol de administrador. En la ventana de tripletas no validadas podrá realizar lo mismo que en la pantalla de las tripletas validadas, es decir, inspeccionar la tripleta y aplicar filtrador, y finalmente validarla o descartarla. En el menú de gestión podrá elegir entre crear tripletas (mismas funciones que para rol colaborador) u otros elementos: gestión de los objetos, predicados, obras y bibliografías (abreviado OPOB). En la gestión OPOB podrá realizar las funciones básicas de creación, visualización, modificación y eliminación.

El fichero RDF se guardará en la carpeta de descargas del dispositivo. Para la visualización del archivo creado se recomienda la herramienta Notepad++.

A2. Instalación

Para instalar y desplegar la aplicación se deberán realizar los siguientes pasos:

1. Ir al repositorio <https://gitlab.inf.uva.es/estegar/tfg.git> para descargar el proyecto.
2. Descomprimir el fichero tfg-main.zip en el directorio que se desee mantener su contenido.

3. Tener instalado MySQL Workbench para crear la base de datos (la instalación y configuración de esta herramienta no está en el ámbito de este documento).
- 3.1. Una vez abierto MySQL Workbench se selecciona el icono de añadir, para crear una nueva conexión a una base de datos.

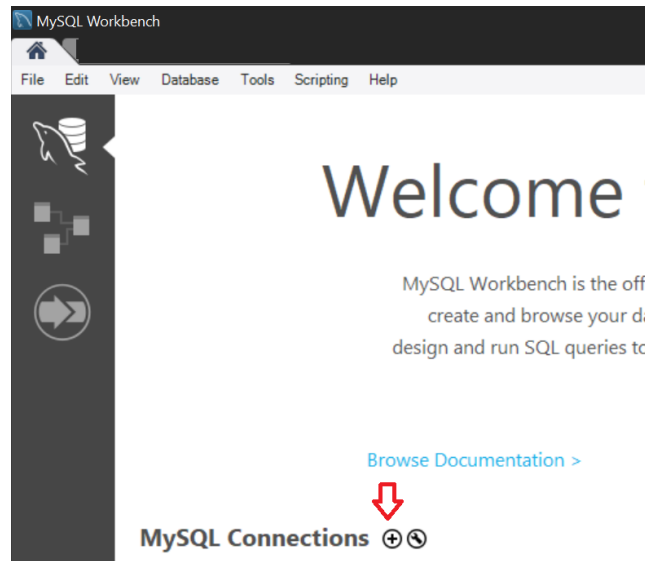


Ilustración 60 - Instalación paso 3.1

- 3.2. En la ventana emergente se deben introducir los datos de la conexión. El primero es el nombre de la conexión (el que desee), los otros vienen por defecto. Si desea crear una conexión a otra base de datos (por ejemplo remota) deberá modificar el código fuente. En concreto en la clase Conexión (paquete Modelo) en los campos db (nombre de la base de datos), ip (dirección ip del servidor. En este caso es un servidor local, por lo que vale con poner *localhost*), port (número de puerto de la conexión), userDB (el nombre de usuario para la conexión a la base de datos), passDB (la contraseña del usuario de la conexión a la base de datos).

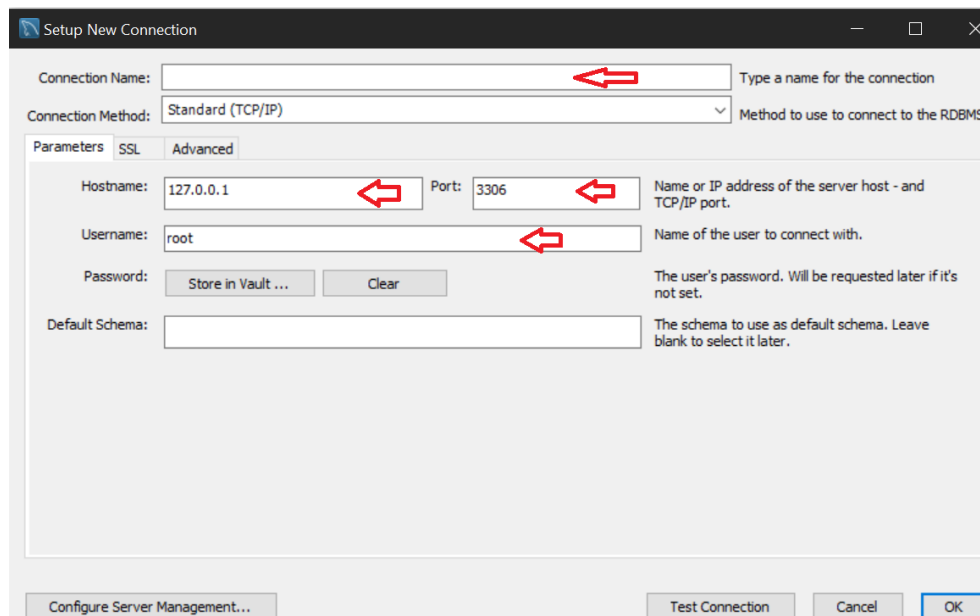


Ilustración 61 - Instalación paso 3.2

```
public class Conexion {  
    //constantes  
    private static final String db = "temas_musicales"; //nombre de la base de datos  
    private static final String ip = "localhost"; // direccion  
    private static final String port = "3306"; // numero del puerto  
    private static final String userDb = "root"; // usuario mysql  
    private static final String passDb = ""; // contraseña del usuario  
    private static final String cadenaConnection="jdbc:mysql://" + ip + ":" + port + "/" + db;
```

Ilustración 62 - Instalación paso 3.2 - 2

3.3. Una vez creada la conexión a la base de datos, seleccione la conexión.

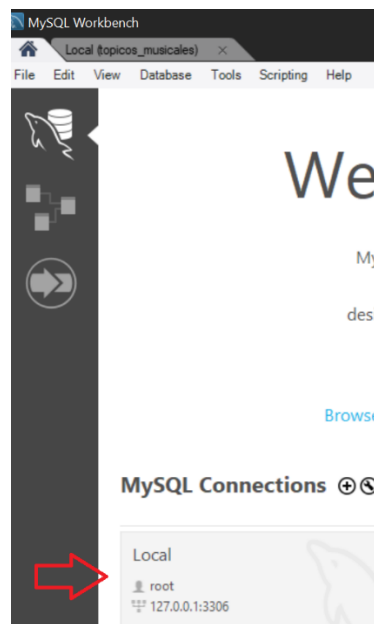


Ilustración 63 - Instalación paso 3.3

3.4. Para crear la base de datos hay que ejecutar el script que se encuentra en el fichero descomprimido. Para poder ejecutarlo hay que seleccionar Archivo y después Abrir SQL script.

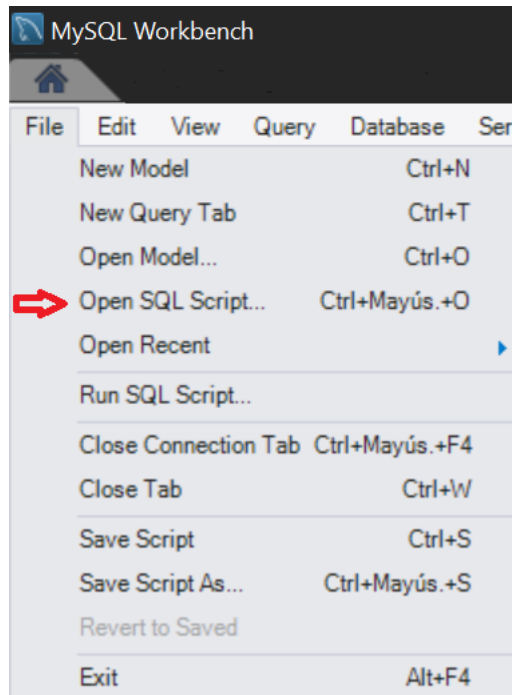


Ilustración 64 - Instalación paso 3.4

3.5. Una vez seleccionado, para ejecutarlo debe pulsar el icono indicado.

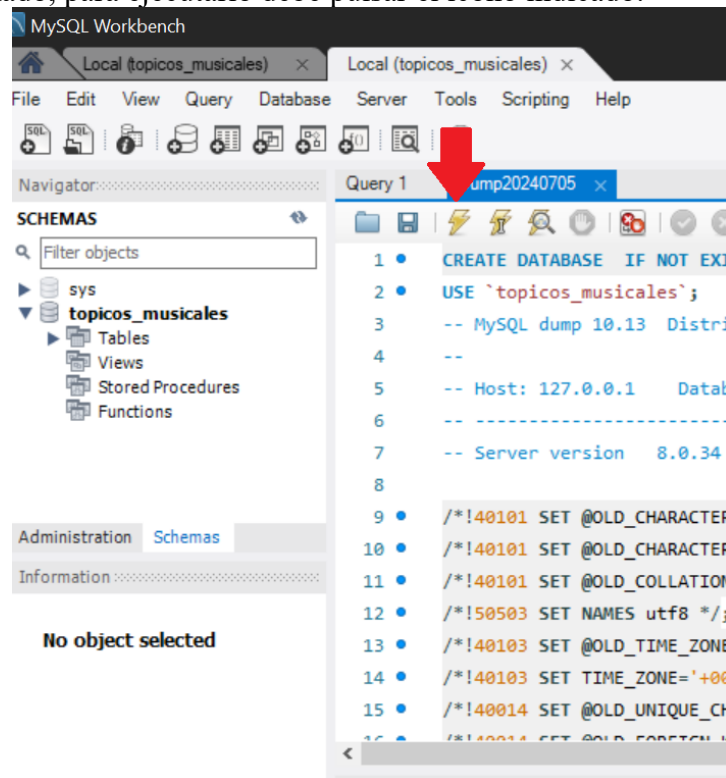


Ilustración 65 - Instalación paso 3.5

3.6. Ya estaría creada la base de datos (listado de la izquierda). Si no ve la base de datos salga y vuelva a entrar.

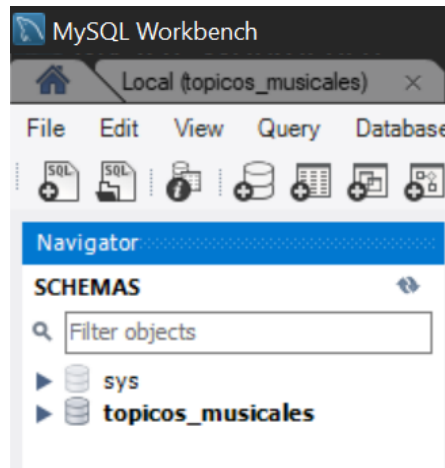


Ilustración 66 - Instalación paso 3.6

4. Tener instalado IDE NetBeans para desplegar la aplicación (la instalación y configuración de esta herramienta no está en el ámbito de este documento).
 - 4.1. Una vez abierto NetBeans, seleccione Archivo y después Abrir proyecto. Abra la carpeta obtenida al descomprimir el fichero zip.

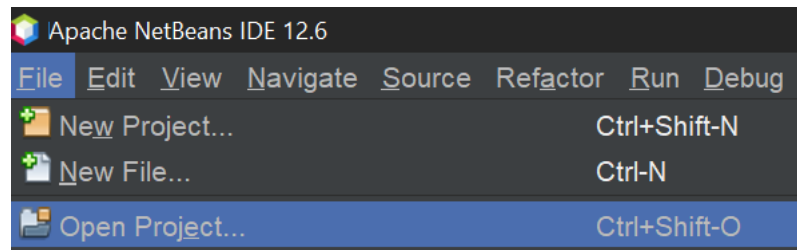


Ilustración 67 - Instalación paso 4.1

- 4.2. Pulsar sobre Limpieza y Generar para poder compilar el código fuente y resolver las dependencias del proyecto. Este paso deberá ser realizado cada vez que se realicen cambios en el código fuente de la aplicación para que todos los ficheros compilados estén actualizados.

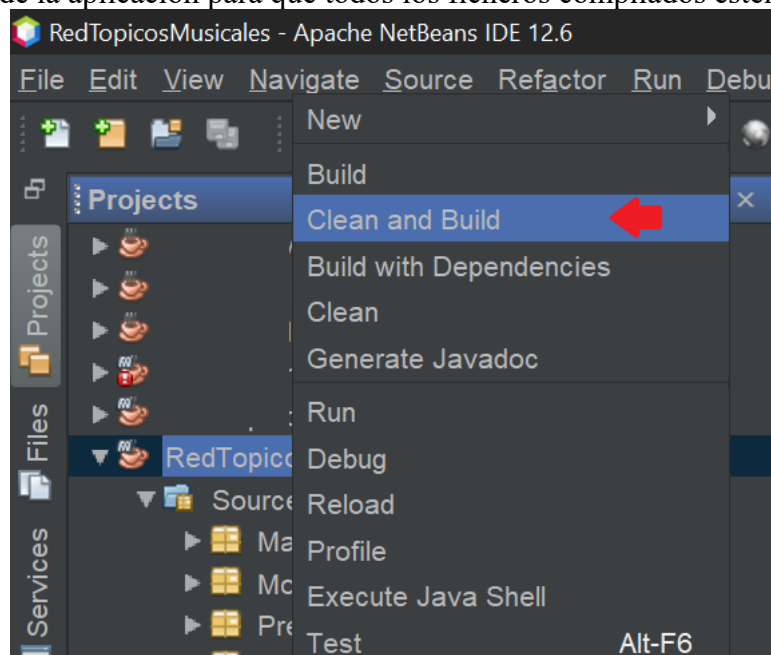


Ilustración 68 - Instalación paso 4.2

4.3. Para lanzar la aplicación, una vez realizados los anteriores pasos, seleccione el proyecto y pulse sobre Ejecutar. Con ello ya habrá lanzado la aplicación desde Netbeans.

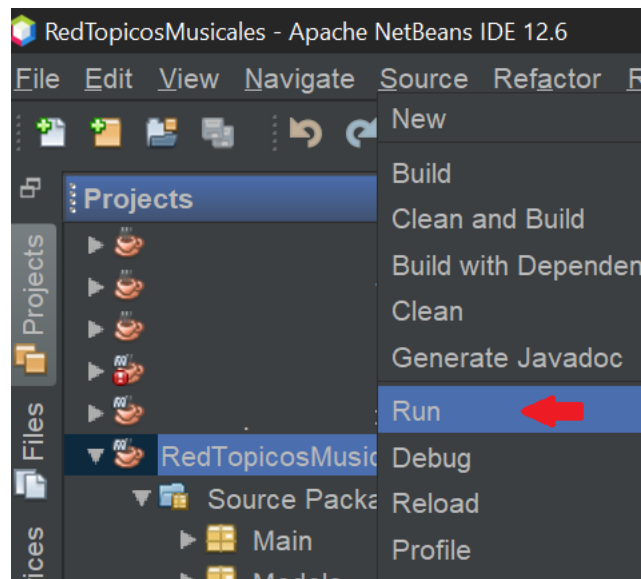


Ilustración 69 - Instalación paso 4.3