



Diseño de una Base de Datos para el Sistema de Captura y Catalogación de Medias

Autora

Lianet Camejo Dominguez

Tutor

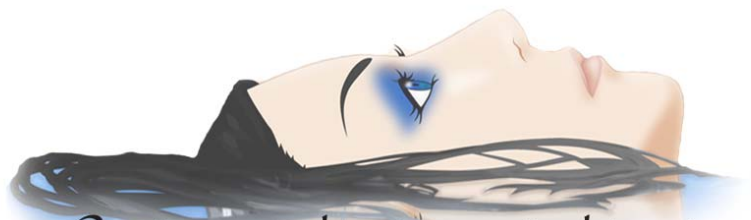
Ing. Yuniór Montaner Hernández

Co-Tutor

Ing. Yoandri Quintana Rondon

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMATICAS

Ciudad de La Habana | Junio 2011 "Año 53 de la Revolución"




*...Querido me es el sueño, y más el de piedra;
Mientras que perdure el daño y la vergüenza:
No ver, no sentir, es mi gran ventura;
Pero no me despierte, ¡oh! hable suavemente...*

Ergo Proxy



Dedicatoria


***A las personas de mi vida,
Especialmente
A mis abuelos Pedro y Francisca***



Agradecimientos

Quiero agradecer...

A Yunior: ¿Y esa luz? ¿Es tu sombra?

Por dar, por estar, por ser...

Por tu ayuda,

Por tus clases.



A Nuria, mi mamá y a Raidel, mi hermanito:



Porque son parte de mi,

Porque son esenciales,

Porque los quiero.

A Rosa, mi suegra:

Por enseñarme y ayudarme,

Porque haz sido extraordinaria,

Por tu apoyo,

Y porque Yunior es lo mejor que me ha pasado

A los amigos y las personas de mi vida, de mi historia:

Por creer...

Por ser.



A todos los que vivieron junto a mí en ese mundo de Undercity, Silvermoon, Orgrimmar y Thunderbluff cuando yo dejaba de ser Lianet y me convertía en Alouqua, porque fui muy feliz.

For the Horde!!!

A cada detalle que hizo posible que llegase hasta aquí...

Gracias!



Resumen

El Proyecto de Captura y Catalogación de Medias que se lleva a cabo en el Centro de Desarrollo de Software de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI) de La Habana, desarrolla el Sistema de Captura y Catalogación de Medias (SCCM), que tiene como objetivo principal automatizar los procesos que se realizan durante el manejo y almacenamiento de materiales audiovisuales en empresas que cuenten con grandes volúmenes de este tipo de archivos, tal es el caso de televisoras como el Instituto Cubano de Radio y Televisión (ICRT) en Cuba y Venezolana de Televisión (VTV) en Venezuela. Algunos de estos procesos son: gestión de fallas en las medias, catalogación de medias, gestión de solicitudes y préstamos de materiales, entre otros.

Para cumplir los objetivos que persigue SCCM de manera eficaz, éste requiere ser provisto de una estructura que permita organizar, centralizar y almacenar sus datos de manera segura. Precisamente en esta investigación se describen las herramientas y técnicas empleadas para el diseño e implementación de una base de datos como solución a este requerimiento, así también el modelo implementado y el proceso de validación de la base de datos obtenida.

Palabras Claves: Base de Datos, SCCM, *PostgreSQL*, Modelo Relacional, Tipologías Dinámicas.



Índice de Contenido

Introducción	1
Capítulo 1 “Fundamentación Teórica”	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Sistema de Captura y Catalogación de Medias	5
1.2.2 Base de datos	6
1.2.3 Tipos de bases de datos	7
1.2.4 Modelos de bases de datos	8
1.2.5 Diseño de base de datos	9
1.3 Análisis de otras soluciones existentes	11
1.4 Conclusiones parciales	13
Capítulo 2 “Herramientas y Tecnologías a Usar”	14
2.1 Introducción	14
2.2 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución	14
2.3 Herramienta de Modelado Visual	16
2.4 Sistema Gestor de Base de Datos	17
2.5 Conclusiones parciales	19
Capítulo 3 “Descripción y Análisis de la Solución”	20
3.1 Introducción	20
3.2 Descripción del producto SCCM	20
3.2.1 Volumen de la información a gestionar	20
3.2.2 Características de la información que se maneja en el sistema	20
3.2.3 Nivel de demanda de servicios concurrentes	21
3.2.4 Mecanismos de aseguramiento de la integridad de la información	21
3.2.5 Requisitos funcionales y no funcionales de SCCM	21
3.3 Clasificación de la base de datos propuesta	22
3.4 Diagrama de clases persistentes	22
3.4.1 Descripción de las clases persistentes	25
3.5 Diseño de la base de datos	30
3.5.1 Modelo lógico de datos	31
3.5.2 Descripción de las tablas	36
3.6 Conclusiones parciales	50



Capítulo 4 “Validación de la Solución Propuesta”	51
4.1 Introducción	51
4.2 Validación teórica del diseño realizado	51
4.2.1 Integridad	51
4.2.2 Normalización de la base de datos	52
4.2.3 Análisis de la redundancia de la información	54
4.2.4 Análisis de la seguridad de la base de datos	55
4.3 Validación funcional del diseño realizado	55
4.3.1 Pruebas de volumen	56
4.3.2 Consultas y tiempo de respuesta	56
4.3.2.1 Mejora del tiempo en búsquedas sobre texto.	58
4.4 Conclusiones parciales	61
Conclusiones Generales	62
Recomendaciones	63
Bibliografía	64
1. Referencias Bibliográficas	64
2. Bibliografía Consultada	65
Anexo I: Especificación de Requisitos del Proyecto SCCM	66
1. Requisitos Funcionales	66
2. Requisitos No Funcionales	79
Anexo II: Descripción de las tablas resultantes en las relaciones del Modelo Entidad-Relación	81
Anexo III: Otras Consultas a la Base de Datos	83



Índice de Figuras

Figura 1. Esquema para el Diseño de una Base de Datos	10
Figura 2. Tabla Media de la base de datos de la primera versión de SCCM.	12
Figura 4. Diagrama de Clases Persistentes - parte 1.	23
Figura 3. Diagrama de Clases Persistentes - parte 1.	24
Figura 5. Modelo lógico de datos. Parte 1.....	32
Figura 6. Modelo lógico de datos. Parte 2.....	33
Figura 7. Representación de la tipología: Película.....	34
Figura 8. Representación de una media virtual.....	34
Figura 9. Modelo lógico de datos. Parte 3.....	35
Figura 10. Mecanismo para controlar la restricción de la información.....	36
Figura 11 Estructura para agilizar las búsquedas de fichas a través de palabras claves.	60



Índice de Tablas

Tabla 1. Descripción de la Clase: Ficha, del Diagrama de Clases Persistentes.	25
Tabla 2. Descripción de la Clase: Media, del Diagrama de Clases Persistentes.	25
Tabla 3. Descripción de la Clase: Solicitud, del Diagrama de Clases Persistentes.	26
Tabla 4. Descripción de la Clase: Tipología, del Diagrama de Clases Persistentes.	26
Tabla 5. Descripción de la Clase: Campo, del Diagrama de Clases Persistentes.	26
Tabla 6. Descripción de la Clase: Componente, del Diagrama de Clases Persistentes.	27
Tabla 7. Descripción de la Clase: Tipo_Datos, del Diagrama de Clases Persistentes.....	27
Tabla 8. Descripción de la Clase: Propiedad_Campo, del Diagrama de Clases Persistentes.	27
Tabla 9. Descripción de la Clase: Atributos_Componente, del Diagrama de Clases Persistentes.	28
Tabla 10. Descripción de la Clase: Proyecto, del Diagrama de Clases Persistentes.	28
Tabla 11. Descripción de la Clase: Falla, del Diagrama de Clases Persistentes.	28
Tabla 12. Descripción de la Clase: Usuario, del Diagrama de Clases Persistentes.....	29
Tabla 13. Descripción de la Clase: Rol, del Diagrama de Clases Persistentes.....	29
Tabla 14. Descripción de la Clase: Permiso, del Diagrama de Clases Persistentes.....	29
Tabla 15. Descripción de la Clase: Autenticacion, del Diagrama de Clases Persistentes.	30
Tabla 16. Descripción de la Clase: Planificacion, del Diagrama de Clases Persistentes.....	30
Tabla 17. Descripción de la tabla: DUsuario.	37
Tabla 18. Descripción de la tabla: DRol.	37
Tabla 19. Descripción de la tabla: DPermiso.	37
Tabla 20. Descripción de la tabla: NTipoPermiso.	37
Tabla 21. Descripción de la tabla: NÁrea.	38
Tabla 22. Descripción de la tabla: DAutenticacion.....	38
Tabla 23. Descripción de la tabla: DPlanificacion.	39
Tabla 24. Descripción de la tabla: NTipoTarea.	39
Tabla 25. Descripción de la tabla: DProyecto.	40
Tabla 26. Descripción de la tabla: NFalla.....	40
Tabla 27. Descripción de la tabla: NTipoFallas.....	40
Tabla 28. Descripción de la tabla: NIntensidad.....	41
Tabla 29. Descripción de la tabla: NDuracion.	41
Tabla 30. Descripción de la tabla: NRestriccion.....	41
Tabla 31. Descripción de la tabla: NFalla_DProyecto.....	42
Tabla 32. Descripción de la tabla: DSolicitud.....	42
Tabla 33. Descripción de la tabla: DCliente.	43
Tabla 34. Descripción de la tabla: NEntidad.	43
Tabla 35. Descripción de la tabla: NPersonaAutoriza.....	43



Tabla 36. Descripción de la tabla: NPersonaAutoriza_DSolicitud.	44
Tabla 37. Descripción de la tabla: DMedia.....	45
Tabla 38. Descripción de la tabla: NTipoSoporte.....	45
Tabla 39. Descripción de la tabla: DSoporte.....	45
Tabla 40. Descripción de la tabla: DFicha.....	46
Tabla 41. Descripción de la tabla: DTipologia.....	46
Tabla 42. Descripción de la tabla: DPalabraClave.....	46
Tabla 43. Descripción de la tabla: DCampo.....	47
Tabla 44. Descripción de la tabla: NTipoDatos.....	47
Tabla 45. Descripción de la tabla: DPropiedCampo.....	47
Tabla 46. Descripción de la tabla: DCampo_DPropiedadCampo.....	48
Tabla 47. Descripción de la tabla: DPropiedCampoValor.....	48
Tabla 48. Descripción de la tabla: DTipologia_DCampo.....	48
Tabla 49. Descripción de la tabla: DTipologia_DCampo_DFicha.....	49
Tabla 50. Descripción de la tabla: DComponente.....	49
Tabla 51. Descripción de la tabla: DPropiedad.....	49
Tabla 52. Descripción de la tabla: DComponente_DTipologia_DCampo.....	50
Tabla 53. Descripción de la tabla: DPropiedadValor.....	50
Tabla 54. Descripción de la tabla: DRol_DPermiso, del Modelo lógico de datos. Anexos.....	81
Tabla 55. Descripción de la tabla: DSolicitud_DFicha, del Modelo lógico de datos. Anexos.....	81
Tabla 56. Descripción de la tabla: DSolicitud_DCliente, del Modelo lógico de datos. Anexos.....	81
Tabla 57. Descripción de la tabla: DSolicitud_DUsuario, del Modelo lógico de datos. Anexos. ..	82
Tabla 58. Descripción de la tabla: DFicha_DFicha, del Modelo lógico de datos. Anexos.....	82
Tabla 59. Descripción de la tabla: DTipologia_DTipologia, del Modelo lógico de datos. Anexos.	82
Tabla 60. Descripción de la tabla: DFicha_DPalabraClave, del Modelo lógico de datos. Anexos.	82



Introducción

Con el desarrollo de las distintas tecnologías asociadas a la informática como ciencia, se concede gran importancia a la gestión de la información a través de los medios electrónicos, pues perfecciona la fluidez de los procesos, la toma de decisiones y la calidad del trabajo en diferentes organizaciones.

Uno de los aportes que ha traído el desarrollo de los campos de la informática y la electrónica ha sido el concepto de Base de Datos (BD) con formatos electrónicos, por cuanto aporta a la planificación y organización en una empresa. La idea es agrupar todos los datos relevantes en un único lugar; la expresión: "datos relevantes" hace referencia a aquella información que por su significado será necesario almacenar y conservar, ya que sustentará las actividades del negocio y la toma de decisiones correspondientes en la empresa y el parámetro que guiará la elección de dichos datos estará dado por los objetivos que se persigan. El hecho de que la información se encuentre en un "único lugar" implica que los datos se encuentran lógicamente unificados e interrelacionados, y constituyen un todo, que debe diseñarse, administrarse y usarse desde un punto de vista global, pero no debe interpretarse desde el punto de vista de la ubicación física.

Actualmente hay empresas, como es el caso de las televisoras que poseen gran cantidad de archivos con materiales audiovisuales que se encuentran en su mayoría en formatos analógicos, almacenados en soportes magnéticos. Para las cuales el acceso a estos materiales es necesario, casi diario, pero resulta poco práctico, dado que la información asociada a ellos se encuentra recogida en fichas técnicas no digitales, lo que convierte su control, búsqueda y acceso en una tarea manual que dificulta la gestión de dichos materiales, incluso puede ocurrir que se cometan errores o simplemente no se encuentre el material solicitado aunque exista.

Para facilitar esta gestión, el Centro de Desarrollo de Software de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI) crea el proyecto Sistema de Captura y Catalogación de Medias (SCCM), con la intención de desarrollar un software capaz de dar soporte a procesos tales como: catalogación de medias, planificación de tareas de edición y catalogación, gestión de fallas de audio y video de un material durante su digitalización, la gestión de los préstamos de materiales, entre otros.



Para la ejecución de estos procesos de manera automática es necesario un mecanismo que permita el almacenamiento de datos de manera unificada e interrelacionada, que sea capaz de preservar y mantener la información relevante y consistente para que esté disponible en tiempo, forma y lugar deseados.

Por estas razones, el propósito de esta investigación radica en solucionar el siguiente **problema**: ¿Cómo proveer al Sistema de Captura y Catalogación de Medias, de una estructura que permita organizar, centralizar y almacenar sus datos de manera segura? Para ello se plantean como **objeto de estudio** el diseño de bases de datos y los procesos asociados al Sistema de Captura y Catalogación de Medias, para aplicar éstos al diseño de la base de datos de dicho sistema, lo cual constituye el **campo de acción**, con **el objetivo general** de desarrollar el diseño e implementación de una base de datos para el Sistema de Captura y Catalogación de Medias.

Se tiene como **idea a defender** que la implementación de la base de datos para el Sistema de Captura y Catalogación de Medias permite la centralización, seguridad y manejo eficaz de los datos.

A continuación se listan las **tareas científicas** para dar cumplimiento al objetivo planteado:

1. Establecimiento de los fundamentos teóricos y metodológicos de la investigación.
2. Descripción de los requisitos, la arquitectura y el funcionamiento del Sistema de Captura y Catalogación de Medias.
3. Descripción de las tecnologías a usar.
4. Desarrollo del diseño lógico de la base de datos.
5. Validación teórica del diseño de la base de datos.
6. Modelación física de la base de datos.
7. Validación funcional del diseño de la base de datos.



Con el propósito de dar solución al objetivo trazado durante la investigación, se utilizaron los **métodos científicos teóricos** siguientes:

1. Analítico-Sintético: Utilizado para lograr extraer lo esencial de la bibliografía consultada durante la descripción de los conceptos asociados al dominio del problema.

2. Modelación: Utilizado para lograr una abstracción y modelar los objetos del mundo real, primero mediante el modelo de clases persistentes, después mediante el diseño lógico de la base de datos, y finalmente cuando se crea el modelo físico.

Además se utilizó el **análisis documental** como método empírico para consultar bibliografía relacionada con el objeto de estudio, así como la **entrevista** en su función de técnica o instrumento para la obtención de información relacionada con el proyecto durante las diferentes etapas de la investigación.



Estructura del contenido del presente trabajo de diploma:

Este trabajo se ha estructurado del siguiente modo: introducción; cuatro capítulos, cada uno de ellos con sus conclusiones parciales; conclusiones generales, recomendaciones, bibliografía, anexos y glosario de términos.

De seguida se describe brevemente el contenido de los cuatros capítulos:

El **capítulo 1** aborda los elementos conceptuales que fundamentan la investigación, incluye temas de diseño de bases de datos, así como un primer acercamiento al funcionamiento del Sistema de Captura y Catalogación de Medias. También describe otras soluciones existentes con experiencias relacionadas y resultados similares.

El **capítulo 2** centra su atención en la descripción de la metodología que guiará el desarrollo de la solución y las herramientas usadas.

El **capítulo 3** describe la solución propuesta, las tablas finales que tendrá la base de datos y sus relaciones. Se muestran los diagramas Entidad-Relación por funcionalidades.

El **capítulo 4** contiene toda la información relativa a las validaciones teórica y funcional de la solución.



Capítulo 1 “Fundamentación Teórica”

1.1 Introducción

En el presente capítulo se hace una descripción detallada de cada uno de los elementos de la teoría que serán utilizados en el desarrollo de la investigación, tal es el caso de los conceptos referentes al diseño de bases de datos y los procesos asociados al Sistema de Captura y Catalogación de Medias, formulaciones sobre las cuales ha de fundamentarse el conocimiento científico propuesto durante la descripción de la solución. Se describen además los antecedentes de la solución, qué tipos de estudios similares se han efectuado, con qué tipo de sujetos, en qué lugares se han llevado a cabo y qué diseños se han utilizado, proporcionando un marco de referencia para alcanzar los resultados esperados.

Esta información queda expuesta de la siguiente manera: Sistema de Captura y Catalogación de Medias, epígrafe que aborda el funcionamiento de los principales procesos de este producto, enfatizando en el flujo de datos y la vida de la información en el sistema; Bases de datos, que constituye una presentación de postulados según autores e investigadores citables en el tema y permiten obtener una visión completa sobre las bases de datos; Tipos de bases de datos; Modelos de datos y Diseño de bases de datos, que orientan sobre cómo habrá de llevarse a cabo la solución; Análisis de soluciones existentes como último epígrafe.

1.2 Conceptos asociados al dominio del problema

1.2.1 Sistema de Captura y Catalogación de Medias

El flujo de datos en SCCM comienza al instante en que el cliente digitaliza sus materiales audiovisuales usando cualquier software (*Adobe Premiere*¹, *Final Cut*², entre otros) o componente de hardware (tarjetas digitalizadoras) habilitado para esto y se registra la información que genera esta actividad además del archivo de la media. Cada uno de estos archivos recién digitalizados pasa por un proceso de calidad en el cual se identifican y registran las posibles fallas de audio o video que puedan presentar para

¹ Adobe Premiere Pro es una aplicación destinada a la edición de video en tiempo real.

² Final Cut es un paquete de programas para la edición profesional de video.



Capítulo 1: “Fundamentación Teórica”

favorecer que sean mitigadas posteriormente durante el proceso de edición. Después de esta fase las medias quedan listas para ser catalogadas y almacenadas con toda la información relativa a ellas de forma organizada en un servidor de medias. Luego las tareas como búsquedas y solicitudes de materiales audiovisuales serán llevadas a cabo de modo rápido y eficaz.

Los procesos en SCCM que requieren almacenar algún tipo de información son los que se describen a continuación:

- 📌 **Administración.** Se encarga de la gestión y el control de usuarios y roles, accesos y tareas; así como la configuración del software.

- 📌 **Control de ingesta.** Permite incorporar al sistema los materiales digitalizados. Controla los procesos de calidad de las medias. Termina con la extracción de fotogramas claves y copias en baja resolución así como publicación en servidores de *streaming*.

- 📌 **Catalogación de medias.** Brinda funcionalidades asociadas a la inserción de los metadatos de los archivos almacenados que hayan sido procesados y publicados en el Servidor de *Streaming* según la tipología correspondiente al fichero y sus relaciones.

- 📌 **Gestión de préstamos.** Controla los préstamos del contenido audiovisual a los clientes. Brinda por tanto funcionalidades para agilizar la búsqueda y recuperación de archivos multimedia de acuerdo a filtros seleccionados por el usuario, en correspondencia con los datos insertados en su catalogación.

- 📌 **Planificación de tareas.** Permite asignar y controlar las tareas de edición y catalogación a usuarios autorizados para ello.

1.2.2 Base de datos

El concepto de base de datos ha sido definido desde el punto de vista de diferentes autores. A continuación se ofrecen algunas definiciones:

“Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que puede considerarse una colección de datos variables en el tiempo”. **(Mato, 2005)**.



Capítulo 1: “Fundamentación Teórica”

“Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada”. (Date, 2003).

“Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requisitos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción”. (Marquéz, 2001).

“Una base de datos es una colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición –estructura de la base de datos– única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos”. (Castaño, 1999)

Básicamente, una base de datos computarizada es una colección o depósito de datos almacenados para su posterior uso, de manera integrada, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real.

1.2.3 Tipos de bases de datos

Las bases de datos pueden clasificarse según la **variabilidad** de los datos almacenados y según el **contenido** que esta posea.

La clasificación según la variabilidad de los datos depende de cómo perduran éstos dentro de la base de datos: si los datos cambian, o si se mantienen estáticos sin sufrir modificaciones. Conceptualmente se catalogan como:

“**Bases de datos estáticas:** Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.” (Barajas, 2009).



Capítulo 1: “Fundamentación Teórica”

“**Bases de datos dinámicas:** Son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.” (Barajas, 2009).

Según el contenido las bases de datos pueden ser catalogadas como:

Directorios: Los diccionarios y guías telefónicas en formato electrónico.

Bases de datos bibliográficas: Sólo contienen un representante de la fuente primaria, por ejemplo: una colección determinada de análisis o autores de alguna materia.

Bases de datos de texto completo: Almacenan las fuentes primarias, como por ejemplo: todo el contenido de todas las ediciones de una colección de revistas científicas.

1.2.4 Modelos de bases de datos

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que permiten describir el mundo real, así como las operaciones que permiten manipular toda esa información. Como modelo, es una abstracción que permite la implementación de sistemas eficientes; por lo general se refieren a algoritmos, y conceptos matemáticos.

Existen varios modelos que con frecuencia son utilizados para el diseño de las bases de datos, alguno de ellos son:

Base de datos jerárquica: Almacena la información en una estructura de datos en forma similar a un árbol, donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padre es llamado raíz, y los que no tienen hijos se los conoce como hojas. Estas bases de datos son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Base de datos de red: Conformada por una colección de registros, conectados entre sí por medio de enlaces en una red. La estructura red abarca más que la estructura de árbol, porque un nodo hijo puede tener más de un nodo padre; es decir, la restricción de que en un árbol jerárquico cada hijo puede tener sólo un padre, se hace menos severa.



Capítulo 1: “Fundamentación Teórica”

Estas bases de datos ofrecen una solución eficiente al problema de redundancia de datos si se compara con el modelo jerárquico; pero, aún así, la dificultad que significa administrar la información trae como consecuencia que sea utilizado en su mayoría por programadores y no por usuarios finales.

Base de datos relacional: Se basa en la teoría matemática de las "relaciones". La idea esencial es el uso de entidades (tablas), compuestas por registros (las filas de una tabla), que representan los campos. En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red), lo cual tiene la ventaja de ser fácil de entender y utilizar por un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas en bases de datos relacionales es SQL (Structured Query Language), un estándar empleado por los principales motores o sistemas de gestión de bases de datos relacionales por explotar la flexibilidad y potencia de los mismos posibilitando la gran variedad de operaciones en ellos.

Base de datos orientada a objeto: Incorpora los conceptos del paradigma orientado a objetos en la programación: encapsulación (propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos), herencia (propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases) y polimorfismo (propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos). En las bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos.

Base de datos multidimensional: Son estructuradas como un cubo, con un eje por dimensión, de estructura basada en dimensiones orientadas a consultas complejas y alto rendimiento. Las bases de datos con este modelo proporciona el desarrollo de los cálculos y análisis multidimensionales que requiere la inteligencia empresarial.

1.2.5 Diseño de base de datos

El proceso de diseñar una base de datos se divide en varios subprocesos que comienzan una vez concluida la fase de recopilar y analizar los requisitos de la base de



Capítulo 1: “Fundamentación Teórica”

datos (ver figura 1). Estos consisten en desarrollar los diseños conceptual, lógico y físico de la base de datos, y cada uno se realiza con el uso de técnicas y métodos específicos.

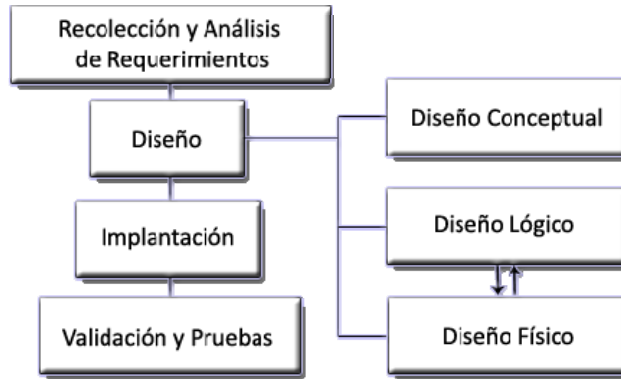


Figura 1. Esquema para el Diseño de una Base de Datos

En el **diseño conceptual** se construye un esquema de la información del entorno o el sistema donde se implantará la base de datos, independientemente de cualquier consideración física. A éste se le denomina **esquema conceptual** y tiene como objetivo lograr la comprensión de la estructura, semántica, relaciones y restricciones de la base de datos, realizar una descripción estable del contenido de la base de datos, lograr la comunicación entre usuarios, analistas y diseñadores, para dar paso al diseño lógico de la base de datos. Se construye utilizando la información que se encuentra en la especificación de los requisitos de usuarios.

El **diseño lógico** es el proceso de construir un esquema de información pero basándose en un modelo de datos específico, independiente del Sistema Gestor de Base de Datos (SGBD) concreto que se utilizará o de cualquier otra consideración física. En esta etapa, se transforma el esquema conceptual en un **esquema lógico** con las estructuras de datos del modelo de base de datos que se vaya a utilizar. Los resultados que se obtienen en esta fase son el conjunto de estructuras propias del modelo abstracto de datos, como son el conjunto de tablas en las bases de datos relacionales.

Durante el **diseño físico** se obtiene un esquema físico, a partir del esquema lógico. Es en el proceso en el cual se realizan la implementación de la base de datos, las estructuras de almacenamiento y los métodos para acceder a la información. Para dar



Capítulo 1: “Fundamentación Teórica”

comienzo a esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, ya que el esquema físico se adapta a él.

Entre el diseño físico y el diseño lógico hay una retroalimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

Al concluir estas fases ya existe prácticamente el sistema de base de datos, entonces se pasa al momento de la evaluación del sistema y posteriormente a una última fase de instalación y mantenimiento del mismo. Casi siempre es necesario modificar el diseño de la base de datos tras su puesta en funcionamiento, por lo que se incluye explícitamente esta fase en el proceso de diseño de bases de datos.

1.3 Análisis de otras soluciones existentes

En el actual mercado de aplicaciones para la gestión de archivos multimedia de grandes, medianas y pequeñas cadenas televisivas imperan las soluciones hechas a la medida para dar solución a requisitos específicos de cada cliente.

La empresa española TEDIAL es líder en el sector gracias a su producto Tarsys, una solución de gestión de archivos multimedia y se compone de diversos módulos que pueden funcionar como productos independientes. Estos módulos son:

- **Indexer:** Software para la catalogación de vídeos que crea un storyboard³ con los fotogramas más característicos, convierte el audio en texto escrito, reconoce patrones, e incorpora un tesoro.
- **Browser:** Visualizador y editor de documentos audiovisuales.
- **AST:** Gestor de librerías robotizadas.
- **IBP Edit:** Editor de vídeo y audio para la postproducción.

Este sistema funciona sobre una base de datos relacional en Oracle que almacena metadatos técnicos, administrativos y de contenido del material audiovisual. Al ser

³ Un storyboard o guión gráfico es un conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender una historia.



Capítulo 1: “Fundamentación Teórica”

Tarsys un software de carácter privativo y de código cerrado, no se tiene acceso a más información de su estructura y funcionamiento.

Para contrarrestar las altas licencias de TEDIA surge el proyecto Sistema de Captura y Catalogación de Medias (SCCM) en el año 2008, encaminado a brindar una solución libre para la gestión multimedia. Este proyecto generó una primera versión del producto con prestaciones muy limitadas que nunca llegó a desplegarse. El Sistema de Captura y Catalogación de Medias (SCCM), para el cual se realiza el diseño de la base de datos en esta investigación, constituye una evolución lógica de esta primera versión e implementa algunas funcionalidades comunes, pero mejoradas, y otras que se identifican y son necesarias para optimizar las soluciones a las que se aspiran. Para ello, se aprovecha y adaptan parte de los conocimientos y experiencias obtenidas anteriormente.

Una de las novedades que introduce esta versión mejorada, es el concepto de tipologías dinámicas⁴.

Como se muestra en la figura 2, la tabla Media es la única estructura en la base de datos de la primera versión de SCCM para guardar la información asociada a cada material gestionado.

Media		
#operacion_id	varchar(255)	Nullable = false
#servidor_media_id	int4	Nullable = false
+id	varchar(255)	Nullable = false
origen	varchar(255)	Nullable = false
fecha_obtencion	date	Nullable = false
duracion	float4	Nullable = false
tipo	varchar(255)	Nullable = false
tiempo_inicio	float4	Nullable = false
nombre	varchar(255)	Nullable = true
título	varchar(255)	Nullable = true
materia	varchar(255)	Nullable = true
catalogado	date	Nullable = true
descripcion	varchar(255)	Nullable = true
calidad	varchar(255)	Nullable = true

Figura 2. Tabla Media de la base de datos de la primera versión de SCCM.

Tener una única estructura para almacenar los datos de todas las distintas categorías que pueden tener los archivos, y sobre todo cuando se trata de caracterizar

⁴ Conjunto de plantillas para guardar datos específicos de cada clasificación de los materiales que se crean de manera dinámica en tiempo de ejecución.



Capítulo 1: “Fundamentación Teórica”

materiales audiovisuales, trae consigo la posibilidad de que la base de datos sea redundante, inconsistente y que posea campos nulos. Además, con este diseño, una vez desplegado el producto éste no será flexible ni adaptable a nuevas situaciones del entorno, lo que hace al software muy específico para no ser construido a la medida.

Éste es sólo un ejemplo, pero el modelo de datos de la primera versión de SCCM en general presenta un diseño estático desde el punto de vista de la manera de guardar sus datos, luego su reutilización es casi imposible.

La nueva versión del producto Sistema de Captura y Catalogación de Medias, identifica nuevos requisitos que requieren incorporar nuevas funcionalidades para lo cual es imprescindible un diseño diferente de la base de datos, que sea tan novedoso como para resolver las insuficiencias detectadas en el diseño anterior.

1.4 Conclusiones parciales

Después de un primer acercamiento a los procesos del Sistema de Captura y Catalogación de Medias, y analizar la base de datos de la primera versión de este producto, así como realizar un estudio de los conceptos asociados a esta estructura, se ha reafirmado la necesidad de proveer al sistema de una base de datos capaz de superar a su predecesora en cuanto a flexibilidad y escalabilidad.

Al concluir este capítulo queda establecido el marco conceptual sobre el cual se dará solución al problema que se pretende resolver con esta investigación.



Capítulo 2 “Herramientas y Tecnologías a Usar”

2.1 Introducción

El presente capítulo aborda la metodología seleccionada por el proyecto Sistema de Captura y Catalogación de Medias que guía el ciclo de desarrollo de la solución, así como las herramientas que se usan durante su modelación, diseño, construcción y despliegue.

2.2 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución

La metodología que guía el desarrollo de la solución es RUP.

“El Proceso Racional Unificado (*Rational Unified Process* habitualmente resumido como RUP), es un proceso de ingeniería de software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requisitos de los usuarios dentro de una planificación y presupuesto establecido. Cubre el ciclo de vida y desarrollo de software”.

(Díaz-Antón, 2004)

“RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

- Desarrollo de software en forma iterativa.
- Manejo de requisitos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente (modela con UML).
- Verifica la calidad del software.
- Controla los cambios.” **(Jacobson, 2000)**

El Proceso Unificado es un proceso de desarrollo de software cuyo ciclo de vida se caracteriza por ser:

- Dirigido por casos de uso.



Capítulo 2: “Tendencias y Tecnologías a Usar”

- 📍 Centrado en arquitectura.
- 📍 Iterativo e incremental.

“El Proceso Unificado de Racional consta de cuatro fases o etapas:

- 📍 **Comienzo o Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- 📍 **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requisitos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- 📍 **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios *release*⁵ del producto que han pasado las pruebas y se ponen a consideración de un subconjunto de usuarios.
- 📍 **Transición:** El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.” (Kruchten, 2004).

Además de las fases, el ciclo de vida de esta metodología contiene flujos de trabajos, los cuales se dividen en Flujos de trabajo de desarrollo y Flujos de trabajo de soporte.

Flujos de trabajo de desarrollo:

- 📍 **Modelado del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- 📍 **Requisitos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

⁵ Un *release* de un producto es una versión candidata a versión final o candidata para el lanzamiento. Comprende un producto final, preparado para publicarse como versión definitiva a menos que aparezcan errores que lo impidan.



Capítulo 2: “Tendencias y Tecnologías a Usar”

- 🔗 **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requisitos), por lo que indica con precisión lo que se debe programar.
- 🔗 **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- 🔗 **Prueba:** Busca los defectos a lo largo del ciclo de vida.

Según esta metodología, el rol del diseñador de base de datos está guiado por el desarrollo de un conjunto de artefactos y actividades para garantizar el cumplimiento satisfactorio de su labor, entre las que se encuentran:

- 🔗 Identificar las clases persistentes.
- 🔗 Elaborar el diagrama de clases persistentes.
- 🔗 Crear clave primaria y restricciones de integridad.
- 🔗 Definir las reglas de integridad referencial y de la información.
- 🔗 Normalizar el diseño de la BD para su optimización.

El artefacto que se obtiene como resultado de las actividades desarrolladas por el diseñador de base de datos es el Modelo de Datos. Éste describe las representaciones lógicas y físicas de los datos persistentes utilizados por la aplicación. Puede ser creado a partir de un conjunto de clases del diseño persistentes, del modelo de diseño, a partir de un Modelo Entidad-Relación creado por el diseñador de BD o mediante ingeniería inversa a partir de una BD existente.

2.3 Herramienta de Modelado Visual

Las herramientas CASE⁶, permiten modelar un software en su totalidad antes de codificarlo, mediante diagramas que permiten modelar cada etapa del ciclo de vida del software, y que logran que se utilice un lenguaje común en el equipo de trabajo.

⁶ CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora).



Capítulo 2: “Tendencias y Tecnologías a Usar”

En el proyecto Sistema de Captura y Catalogación de Medias, se usó como herramienta de modelado: *Visual Paradigm*, una plataforma diseñada para arquitectos, desarrolladores y diseñadores con el objetivo de acelerar el proceso de análisis y diseño de aplicaciones empresariales complejas a través de UML (Lenguaje de Modelado Unificado), regulado por OMG (*Object Management Group*⁷) que permite construir y documentar los artefactos del sistema reduciendo el ciclo de vida de construcción del software y aumentando su calidad.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Agiliza la construcción de aplicaciones con calidad y a un menor coste. Posibilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, así como ingeniería inversa de bases de datos. (Padron, 2009).

Otras ventajas que justifican esta elección vienen dadas por el hecho de que esta herramienta es fácil de instalar y actualizar, es de software libre y cuenta con una comunidad que le brinda soporte técnico.

2.4 Sistema Gestor de Base de Datos

Un sistema gestor de bases de datos (SGBD o DBMS), es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. (Date, 2003). Un sistema gestor de bases de datos debe:

- Dar la capacidad a los usuarios de almacenar datos en la base de datos, acceder a ellos y actualizarlos.
- Proporcionar mecanismos para asegurar que la base de datos se actualice correctamente cuando varios usuarios la actualizan concurrentemente.

⁷ **OMG** (de sus siglas en inglés Grupo de Gestión de Objetos) es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA. Es una organización que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para las mismas.



Capítulo 2: “Tendencias y Tecnologías a Usar”

- ❖ Proporcionar mecanismos para recuperar la base de datos en caso de que ocurra algún suceso que la dañe llevándola a un estado consistente.
- ❖ Garantizar que sólo los usuarios autorizados pueden acceder a la base de datos.
- ❖ Suministrar los medios necesarios para garantizar la integridad de la base de datos.
- ❖ Facilitar un conjunto de herramientas que permitan administrar la base de datos de modo efectivo.

Características como rapidez y efectividad en los procesos y los flujos de información son imprescindibles para optimizar servicios. Ante esta notable demanda han surgido multitud de gestores de bases de datos. Estos programas permiten manejar la información de modo sencillo y prestar servicios para el desarrollo y el manejo de bases de datos. **(Date, 2003)**. Al seleccionar el SGBD, que se usará en un proyecto determinado, los criterios suelen ser variables, ya que dichos sistemas pueden tener comportamientos similares, lo que requiere que se haga un análisis, previo a la elección, de las opciones a partir de las necesidades del problema. La base de datos del Sistema de Captura y Catalogación de Medias, se implementó con *PostgreSQL*, uno de los gestores más estables y robustos que existen actualmente, en su versión 8.4.

Seguidamente se enumeran elementos que justifican la elección de *PostgreSQL*:

- ❖ Diseño para ambientes de alto volumen de información.
- ❖ Instalación ilimitada y flexibilidad a los cambios.
- ❖ Sus considerables ahorros en operación, su licencia BSD ⁸(*Berkeley Software Distribution*), rentabilidad en los modelos de negocio, y su extensibilidad son características muy atractivas.
- ❖ Es multiplataforma (compatible con Linux, Windows, y varias versiones de UNIX).

⁸ BSD es una licencia de software flexible respecto a la distribución, de modo que el software puede ser redistribuido como software libre o software propietario, siendo libre la licencia original del autor. Permite el uso del código fuente en software no libre.



Capítulo 2: “Tendencias y Tecnologías a Usar”

- Alta concurrencia, mediante un sistema denominado Acceso Concurrente Multiversión. Éste permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tablas o por filas común en otros gestores.
- Soporta diferentes tipos de datos.
- Posee estrategias preventivas para desastres: posee una estructura adelantada de registros que evita pérdida de datos en caso de falla eléctrica, de sistema operativo o de hardware.
- Incorpora productos de software para el trabajo con el mismo: *PgAdmin*, *PgAcces*, *Psql*, *PhpPgAdmin*, *PgCluster*, entre otros.
- Utiliza el *PgCluster*, el *Slony-I* y *PgPool* para la replicación de datos, el primero en réplicas multi-maestro y los demás en réplicas maestro-esclavo.
- Posee buen soporte brindado por la gran comunidad de usuarios que existe en el mundo que aportan experiencias y resultados obtenidos del uso del mismo.

A continuación se justifica la elección de la versión *PostgreSQL* 8.4 específicamente:

Cada versión de *PostgreSQL* presenta nuevas características debido a la creciente comunidad de contribuyentes. La versión 8.4 proporciona mejoras con nuevas implementaciones de funcionalidades que hacen de *PostgreSQL* más fácil de administrar. Se han añadido varias optimizaciones, tales como semi-joins y anti-une, que proveen una mejora notable en el tiempo de ejecución de algunas de las consultas más exigentes.

2.5 Conclusiones parciales

Al concluir este capítulo se establece la metodología RUP, como base para el desarrollo de la solución y las herramientas que se usarán en el desarrollo de la solución. Tal es el caso de *Visual Paradigm*, como herramienta de modelado, aprovechando entre otras de sus características el hecho de que usa UML como lenguaje de modelado; y *PostgreSQL* 8.4 como SGBD.



Capítulo 3 “Descripción y Análisis de la Solución”

3.1 Introducción

Para construir el modelo de datos para la base de datos relacional de SCCM, se transitó por las tres fases generales en el desarrollo de un modelo de datos, según *Rational Unified Process*, que reflejan los diferentes niveles de detalle en el diseño del almacenamiento de datos persistentes y los mecanismos de recuperación de la aplicación: conceptual, lógico y físico.

En este capítulo se describe la solución propuesta después de enfatizar en los requisitos funcionales y no funcionales del SCCM y obtener el diagrama de clases persistentes, como parte del diseño conceptual del modelo. Para mostrar el diseño de la base de datos obtenida se describe con detalle el modelo lógico de datos obtenido, artefacto que figura las representaciones lógicas y físicas de los datos persistentes utilizados por el sistema, así como cada una de sus tablas.

3.2 Descripción del producto SCCM

A continuación se describen los requisitos del producto Sistema de Captura y Catalogación de Medias en cuanto a la distribución de los servicios, características de la información a manejar, volumen de la información a gestionar, funcionamiento, aseguramiento de la integridad de la información y concurrencia de los servicios.

3.2.1 Volumen de la información a gestionar

El sistema debe ser capaz de manejar toda la información audiovisual que se digitalice y los datos asociados por concepto de procesamiento y catalogación de los mismos. Además la información relacionada con las solicitudes de materiales por parte de clientes a la empresa y al proceso de administración de SCCM.

3.2.2 Características de la información que se maneja en el sistema

La información que se maneja con la aplicación se clasifica según dos criterios: Los **ficheros audiovisuales** que se han digitalizado con una copia en baja resolución y los fotogramas clave; Los **datos de catalogación** asociados a audiovisuales y los datos de



Capítulo 3: “Descripción y Análisis de la Solución”

usuarios que interactúan con el software, todos almacenados en la base de datos del sistema.

La información contenida en el sistema sólo debe ser accedida por los usuarios que tengan roles asignados y permisos para su consulta y manipulación.

3.2.3 Nivel de demanda de servicios concurrentes

La aplicación tendrá en cuenta las peticiones concurrentes sobre los materiales audiovisuales digitalizados que se encuentren en el archivo. Se debe permitir que los documentalistas audiovisuales realicen la consulta y descripción de los mismos.

El resto de los usuarios podrá realizar consultas de la información digitalizada disponible mediante los flujos de *streaming* publicados, es posible la conexión concurrente de 100 usuarios o más en condiciones óptimas de conexión.

3.2.4 Mecanismos de aseguramiento de la integridad de la información

La aplicación contará con un procedimiento de autenticación de usuario y control de acceso para todas las funcionalidades según los roles que se definan con el cliente para cada proceso a desarrollar, que permitirá la integridad de la información y seguridad.

3.2.5 Requisitos funcionales y no funcionales de SCCM

Para que un sistema de software funcione adecuadamente se debe lograr una comunicación efectiva entre el equipo de proyecto y los usuarios finales, con el objetivo de llegar a un entendimiento de lo que se quiere lograr, lo que constituye la clave del éxito en la producción de un software.

Los requisitos se dividen en dos grupos: Requisitos Funcionales y No Funcionales. Donde los primeros son capacidades o condiciones que el sistema debe tener, manteniéndose estos invariables, sin importar con qué propiedades o cualidades se relacionen y los segundos son propiedades o cualidades que el producto debe cumplir, entiéndase estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. **(Pressman, 2005)**



Capítulo 3: “Descripción y Análisis de la Solución”

El Informe Técnico de Especificación de Requisitos del proyecto Sistema de Captura y Catalogación de Medias, se encuentra en el **Anexo I** de este documento.

3.3 Clasificación de la base de datos propuesta

La información almacenada en el Sistema de Captura y Catalogación de Medias, necesitará ser actualizada, editada y modificada constantemente, independientemente de que parte de ella sólo será almacenada para consulta, por tal motivo será clasificada de acuerdo a la variabilidad de sus datos como **dinámica**.

En cuanto a la clasificación del modelo que se utilizará en la solución propuesta será una **base de datos relacional**, pues las otras como la orientada a objetos imponen restricciones en la estructura básica (los objetos) y la relación entre estos (sólo por herencia). Además, modelos como el de base de datos jerárquica o de red son de muy complejo entendimiento para los usuarios y sus mecanismos para minimizar la redundancia de la información son insuficientes, **(Padron, 2009)**. El modelo relacional en cambio, presenta ventajas al poseer mecanismos de disminución de redundancia e inconsistencias de datos; garantizar la integridad referencial, es decir, al eliminar un registro se eliminan sus relaciones y dependencias; ser de fácil comprensión al ignorar el almacenamiento físico de los datos y centrarse en el modelo lógico de la base de datos. **(Mato, 2005)**.

3.4 Diagrama de clases persistentes

La persistencia de una clase está definida por la propiedad de los objetos para trascender su estado en el tiempo y el espacio: una clase persistente existirá durante la ejecución de un programa, y deberá sobrevivir incluso a la eliminación o colapso del mismo. Lo contrario a las clases persistentes son las clases temporales que dejan de existir cuando termina el programa. **(Un método para el diseño de la Base de Datos a partir del Modelo Orientado a Objetos, 2004)**.

En el Diagrama de clases persistentes además de representar las clases que se definen, se representan los atributos que contendrá cada una, las relaciones de asociación o composición entre ellas. **(Pressman, 2005)**. El Diagrama de clases persistentes de SCCM, se muestra a continuación dividido en dos partes.



Capítulo 3: “Descripción y Análisis de la Solución”

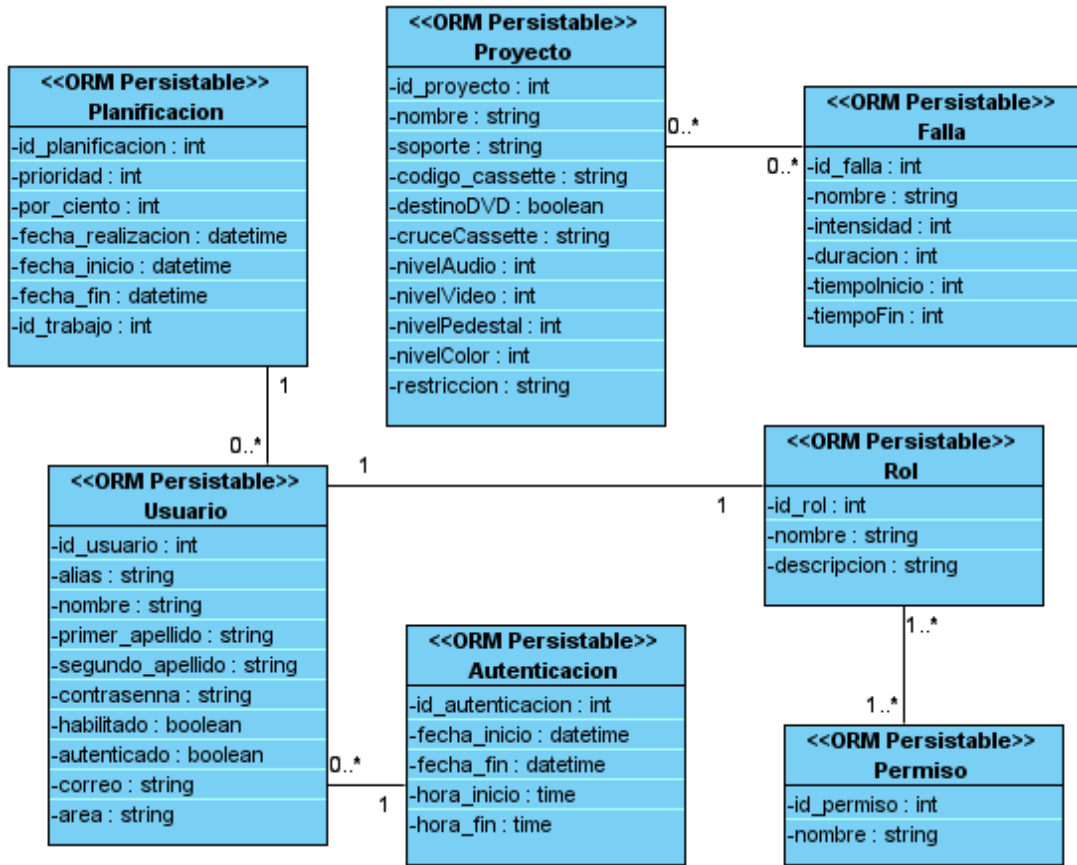


Figura 3. Diagrama de Clases Persistentes - parte 1.



Capítulo 3: “Descripción y Análisis de la Solución”

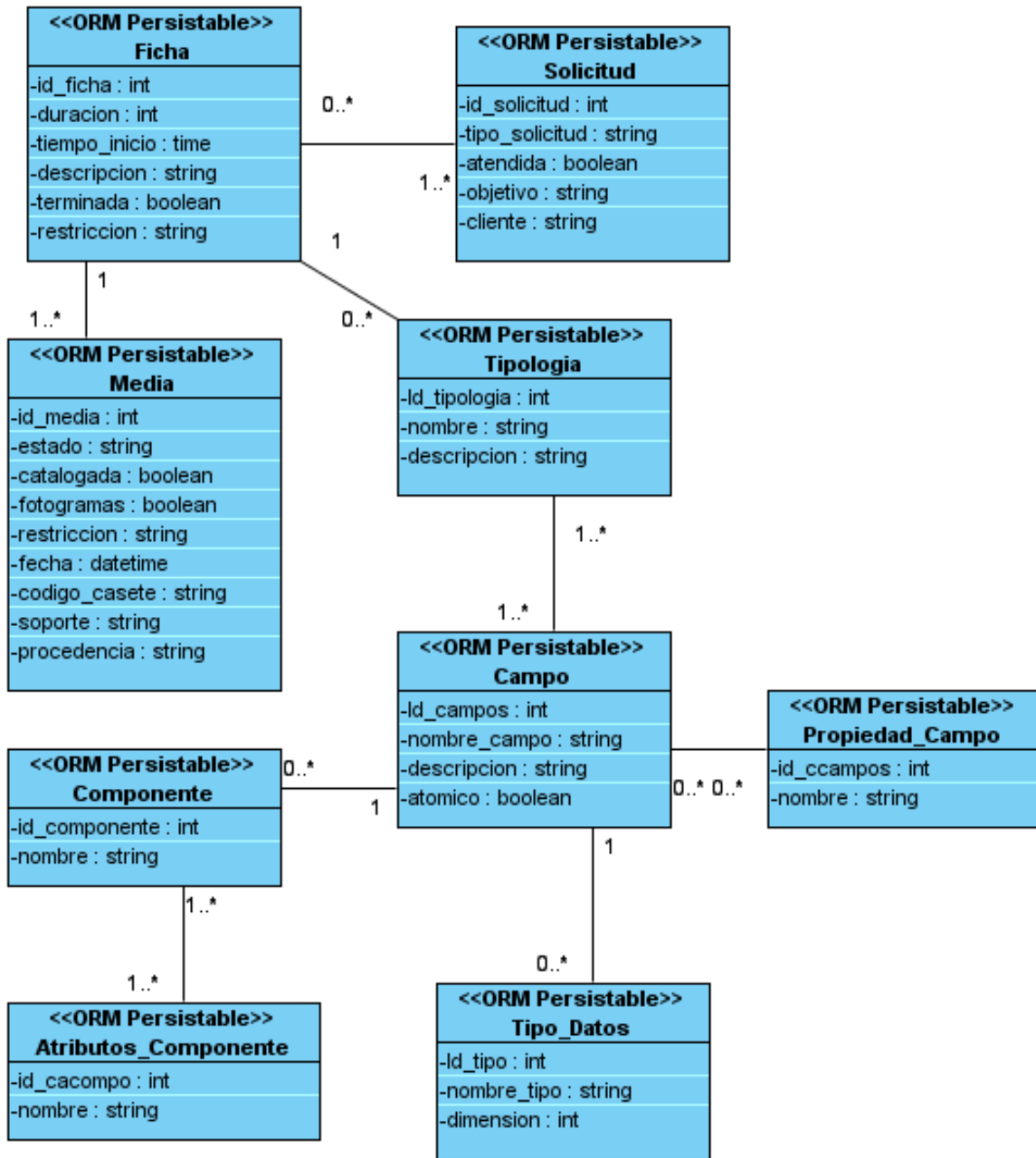


Figura 4. Diagrama de Clases Persistentes - parte 1.



Capítulo 3: “Descripción y Análisis de la Solución”

3.4.1 Descripción de las clases persistentes

Se describen a continuación las clases y sus atributos:

Nombre de la Clase: Ficha	
Descripción de la Clase: Representa virtualmente una media física o un fragmento de media. Para los usuarios del sistema la ficha es la media.	
Atributo	Tipo de dato
id_ficha	int
duracion	int
tiempo_inicio	time
descripcion	string
terminada	boolean
restriccion	string

Tabla 1. Descripción de la Clase: Ficha, del Diagrama de Clases Persistentes.

Nombre de la Clase: Media	
Descripción de la Clase: Representa el archivo físico que contiene el material audiovisual almacenado en el sistema.	
Atributo	Tipo de dato
id_media	int
estado	string
catalogada	boolean
fotogramas	boolean
restricción	string
fecha	datetime
codigo_cassette	string
soporte	string
procedencia	string

Tabla 2. Descripción de la Clase: Media, del Diagrama de Clases Persistentes.



Capítulo 3: “Descripción y Análisis de la Solución”

Nombre de la Clase: Solicitud	
Descripción de la Clase: Representa la solicitud de materiales que recibe la entidad por parte de clientes.	
Atributo	Tipo de dato
id_solicitud	int
tipo_solicitud	string
atendida	boolean
objetivo	string
cliente	string

Tabla 3. Descripción de la Clase: Solicitud, del Diagrama de Clases Persistentes.

Nombre de la Clase: Tipología	
Descripción de la Clase: Representa el prototipo o la tipología que se creará para agrupar campos, que almacenarán información particular de un material o de un conjunto de materiales específico.	
Atributo	Tipo de dato
id_tipologia	int
nombre	string
descripcion	string

Tabla 4. Descripción de la Clase: Tipología, del Diagrama de Clases Persistentes.

Nombre de la Clase: Campo	
Descripción de la Clase: Representa el nombre que se le dará a un atributo de un material determinado y se agrupará en una tipología.	
Atributo	Tipo de dato
id_campo	int
nombre_campo	string
descripcion	string
atomico	boolean

Tabla 5. Descripción de la Clase: Campo, del Diagrama de Clases Persistentes.



Capítulo 3: “Descripción y Análisis de la Solución”

Nombre de la Clase: Componente	
Descripción de la Clase: Representa el componente visual con el cual se creará la interfaz del formulario que contendrá los campos que corresponden a una tipología determinada.	
Atributo	Tipo de dato
id_componente	int
nombre	string

Tabla 6. Descripción de la Clase: Componente, del Diagrama de Clases Persistentes.

Nombre de la Clase: Tipo_Datos	
Descripción de la Clase: Representa el tipo de datos que tendrá el valor contenido en un campo determinado.	
Atributo	Tipo de dato
Id_tipo	int
nombre_tipo	string
dimension	int

Tabla 7. Descripción de la Clase: Tipo_Datos, del Diagrama de Clases Persistentes.

Nombre de la Clase: Propiedad_Campo	
Descripción de la Clase: Representa los campos que contiene un campo compuesto. Por ejemplo: Al campo locutor lo componen otros campos como nombre, apellidos, edad, entre otros.	
Atributo	Tipo de dato
Id_ccampos	int
nombre	string

Tabla 8. Descripción de la Clase: Propiedad_Campo, del Diagrama de Clases Persistentes.

Nombre de la Clase: Atributos_Componente	
Descripción de la Clase: Representa el atributo que se necesita especificar para que un componente pueda construirse. Por ejemplo: Un componente de tipo editor de texto, debe especificar atributos como tamaño de la letra, fuente, entre otros.	
Atributo	Tipo de dato



Capítulo 3: “Descripción y Análisis de la Solución”

Id_cacompo	int
nombre	string

Tabla 9. Descripción de la Clase: Atributos_Componente, del Diagrama de Clases Persistentes.

Nombre de la Clase: Proyecto	
Descripción de la Clase: El proyecto se crea cuando se digitaliza la media. Un proyecto contiene las fallas que posee un material, sin editar.	
Atributo	Tipo de dato
id_proyecto	int
nombre	string
soporte	string
codigo_casette	string
destinoDVD	boolean
cruceCasette	string
nivelAudio	int
nivelVideo	int
nivelPedestal	int
nivelColor	int
restriccion	string

Tabla 10. Descripción de la Clase: Proyecto, del Diagrama de Clases Persistentes.

Nombre de la Clase: Falla	
Descripción de la Clase: Representa las fallas, que puede contener un material, y que será necesario reparar antes de pasar al proceso de catalogación.	
Atributo	Tipo de dato
id_falla	int
nombre	string
intensidad	int
duracion	int
tiempoInicio	int
tiempoFin	int

Tabla 11. Descripción de la Clase: Falla, del Diagrama de Clases Persistentes.



Capítulo 3: “Descripción y Análisis de la Solución”

Nombre de la Clase: Usuario	
Descripción de la Clase: Representa al usuario que tendrá el sistema.	
Atributo	Tipo de dato
id_usuario	int
alias	string
contrasenna	string
nombre	string
primer_apellido	string
segundo_apellido	string
habilitado	boolean
autenticado	boolean
correo	string
area	string

Tabla 12. Descripción de la Clase: Usuario, del Diagrama de Clases Persistentes.

Nombre de la Clase: Rol	
Descripción de la Clase: Representa el rol que juega cada usuario dentro del sistema. Cada rol posee diferentes permisos o accesos.	
Atributo	Tipo de datos
id_rol	Int
nombre	string
descripción	string

Tabla 13. Descripción de la Clase: Rol, del Diagrama de Clases Persistentes.

Nombre de la Clase: Permiso	
Descripción de la Clase: Representa los permisos o accesos que puede tener un usuario del sistema a diferentes funcionalidades	
Atributo	Tipo de dato
id_permiso	int
nombre	string

Tabla 14. Descripción de la Clase: Permiso, del Diagrama de Clases Persistentes.



Capítulo 3: “Descripción y Análisis de la Solución”

Nombre de la Clase: Autenticacion	
Descripción de la Clase: Representa las entradas y los accesos de cada usuario al sistema.	
Atributo	Tipo de dato
id_autenticacion	int
fecha_inicio	datetime
fecha_fin	datetime
hora_inicio	time
hora_fin	time

Tabla 15. Descripción de la Clase: Autenticacion, del Diagrama de Clases Persistentes.

Nombre de la Clase: Planificacion	
Descripción de la Clase: Representa la programación o planificación que se hace en el sistema para realizar cada una de las tareas. Las tareas pueden ser: editar, catalogar, entre otras.	
Atributo	Tipo de dato
id_planificacion	int
prioridad	int
por_ciento	int
fecha_realizacion	datetime
fecha_inicio	datetime
fecha_fin	datetime
Id_trabajo	int

Tabla 16. Descripción de la Clase: Planificacion, del Diagrama de Clases Persistentes.

3.5 Diseño de la base de datos

La actividad de diseñar la base de datos, tiene como propósito fundamental, asegurar que los datos persistentes sean almacenados consistentemente de manera eficiente y definir el comportamiento que debe ser implementado en la base de datos. Para lograr este resultado se realiza primeramente, el mapeo del Modelo de clases persistentes al Modelo de datos. (Jacobson, et al., 2000)



Capítulo 3: “Descripción y Análisis de la Solución”

3.5.1 Modelo lógico de datos

Para una mejor comprensión del diseño de la base de datos propuesto y basado en la teoría que se plantea en la Guía de *Rational Unified Process V 7.0.1*⁹, se describirá en este epígrafe el modelo lógico de los datos en las figuras 5, 6, 9 y 10. En él se ilustran las entidades lógicas clave y sus relaciones que son necesarias para satisfacer los requisitos del sistema para datos persistentes que sean coherentes con la arquitectura global de la aplicación. Antes de desarrollar los elementos del modelo lógico de datos para crear el diseño físico de la base de datos este modelo se ha perfeccionado aplicando las reglas estándar de normalización que se detallan en el capítulo 4 de este trabajo.

La figura 5, modela las tablas en la que se almacenan los datos correspondientes a los procesos de gestión de usuarios, gestión de solicitudes de materiales y planificación de tareas de edición y catalogación de medias. Éstos se explican a continuación:

- Gestión de usuarios: Cada usuario pertenece a un área de la empresa y juega un rol determinado dentro del sistema. Cada rol tiene asociado permisos de diferentes tipos. Para custodiar la seguridad del sistema se controlan las entradas de los usuarios al mismo a través del mecanismo de autenticación.
- Gestión de solicitudes de materiales: Un material es representado por su ficha correspondiente. Las solicitudes de materiales pueden ser internas, es decir, realizadas por usuarios del sistema o externas, realizadas por clientes de otras empresas. Para atender la solicitud de un material, antes debe ser autorizada por una persona facultada para ello.
- Planificación de tareas de edición y catalogación: El proceso de planificación consiste en asignar tareas de edición o catalogación a determinados usuarios, para que sean realizadas en un margen de tiempo determinado.

⁹ Guía para el estudio de *Rational Unified Process* Versión 7.0.1 de la Corporación IBM 1987, 2005.



Capítulo 3: “Descripción y Análisis de la Solución”

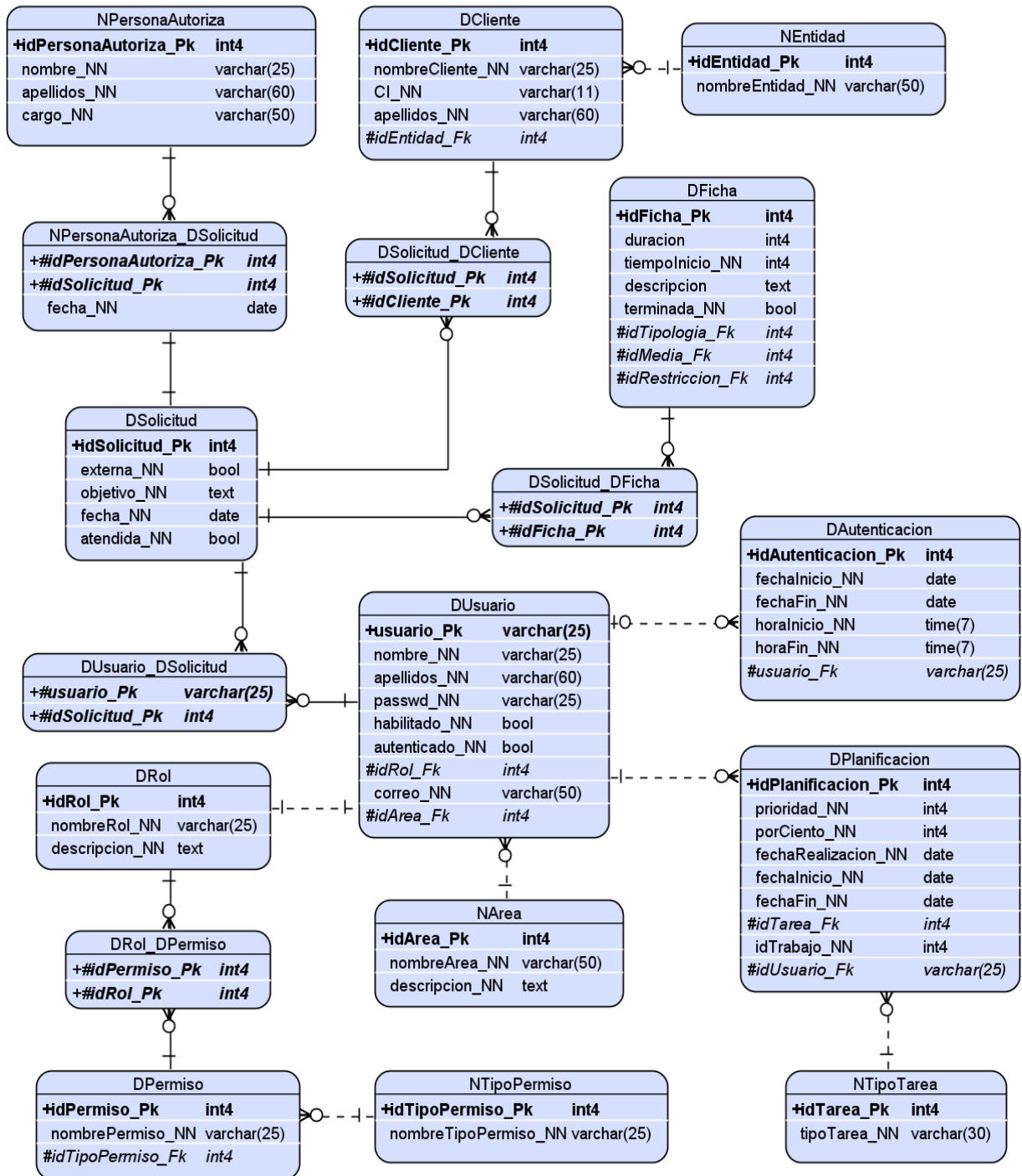


Figura 5. Modelo lógico de datos. Parte 1.



Capítulo 3: “Descripción y Análisis de la Solución”

En la figura 6, se modelan las tablas que almacenan los datos de la gestión de de los proyectos. Un proyecto se crea después de que la media es digitalizada y registra las fallas que ésta posee para facilitar el posterior proceso de edición.

Las fallas pueden ser de tipo audio o video y tienen cierta duración e intensidad, atributos que son calificados numéricamente.

Las medias provienen de diferentes tipos de soporte, de los cuales se almacena su código. Si la media es confidencial, o posee algún nivel de restricción, el proyecto que la maneja es también restringido.

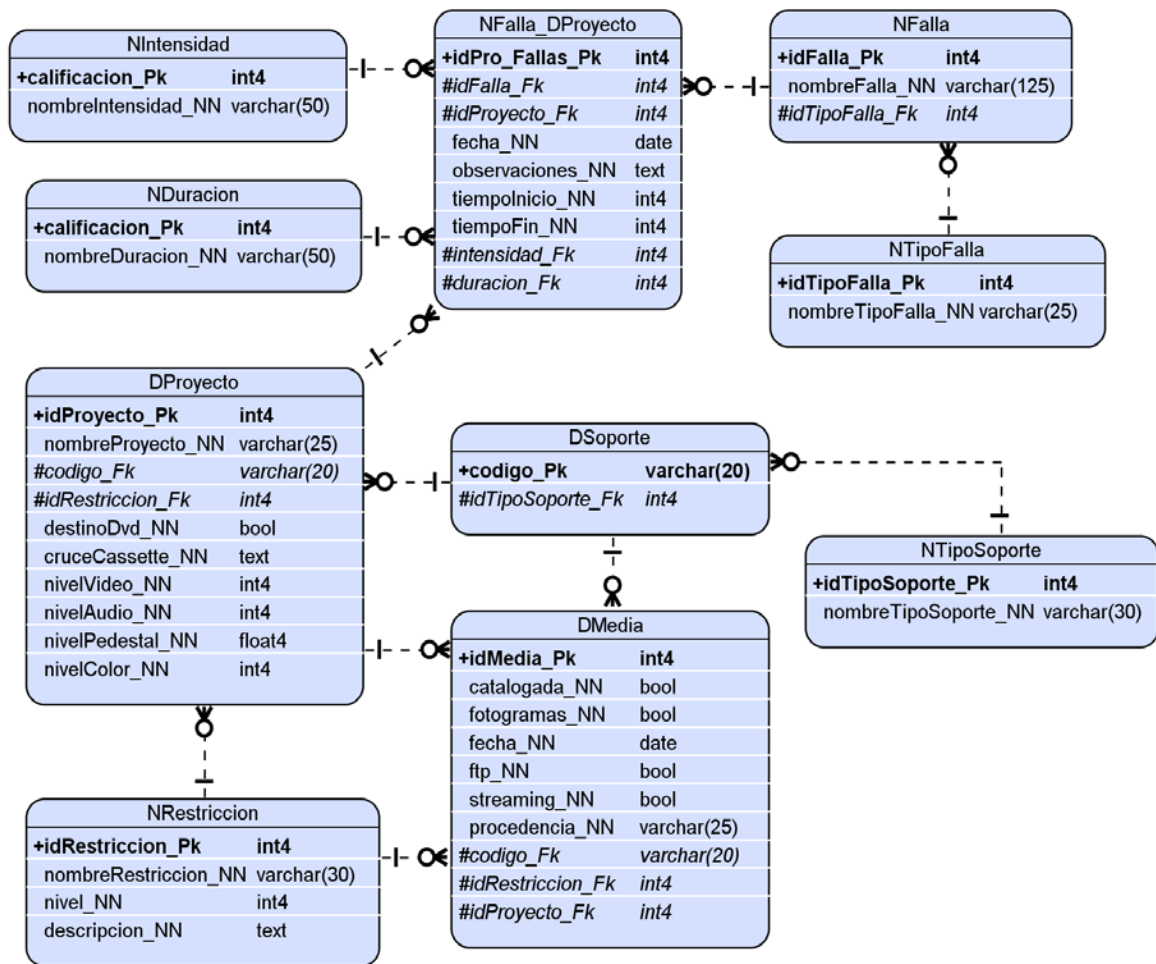


Figura 6. Modelo lógico de datos. Parte 2.

Una tipología, es el prototipo de catalogación que se creará para agrupar campos, que van a almacenar información particular de un material o de un conjunto de materiales específico. Por ejemplo, en la figura 7, Película es una tipología, que contiene



Capítulo 3: “Descripción y Análisis de la Solución”

campos como Título, Género, Sinopsis y Año. Sus datos son específicos de los materiales representados con fichas 1 y 2.

Película				
Ficha	Título	Género	Sinopsis	Año
1	K-PAX	Drama	2001
2	The Last Unicorn	Fantasía	1982

Figura 7. Representación de la tipología: Película

Cuando un usuario va a catalogar una media, busca la tipología que corresponda con la información que quiera almacenar sobre la misma. En caso que no encuentre ninguna, podrá crear la nueva tipología de manera dinámica, y podrá asociarle campos que ya están definidos para otras tipologías o campos nuevos.

Una tipología, es también un formulario que el usuario debe llenar, por tanto se necesita la manera de asociar a cada campo un componente con propiedades definidas, para que pueda construirse en tiempo de ejecución, con ello se logra que tanto tipologías como sus interfaces sean totalmente dinámicas, configurables y adaptables a cualquier entorno.

La estructura que almacena los datos del proceso de catalogación de medias es la representada en el diagrama de la figura 9.

Otro suceso que se logra con este diseño de base de datos es que la ficha de una media no necesariamente tiene que representar el archivo físico de la media, porque fragmentos de un material pueden ser tratados y catalogados como medias independientes, propiciando un menor consumo de espacio en el servidor de almacenamiento. (Ver figura 8)



Figura 8. Representación de una media virtual



Capítulo 3: “Descripción y Análisis de la Solución”

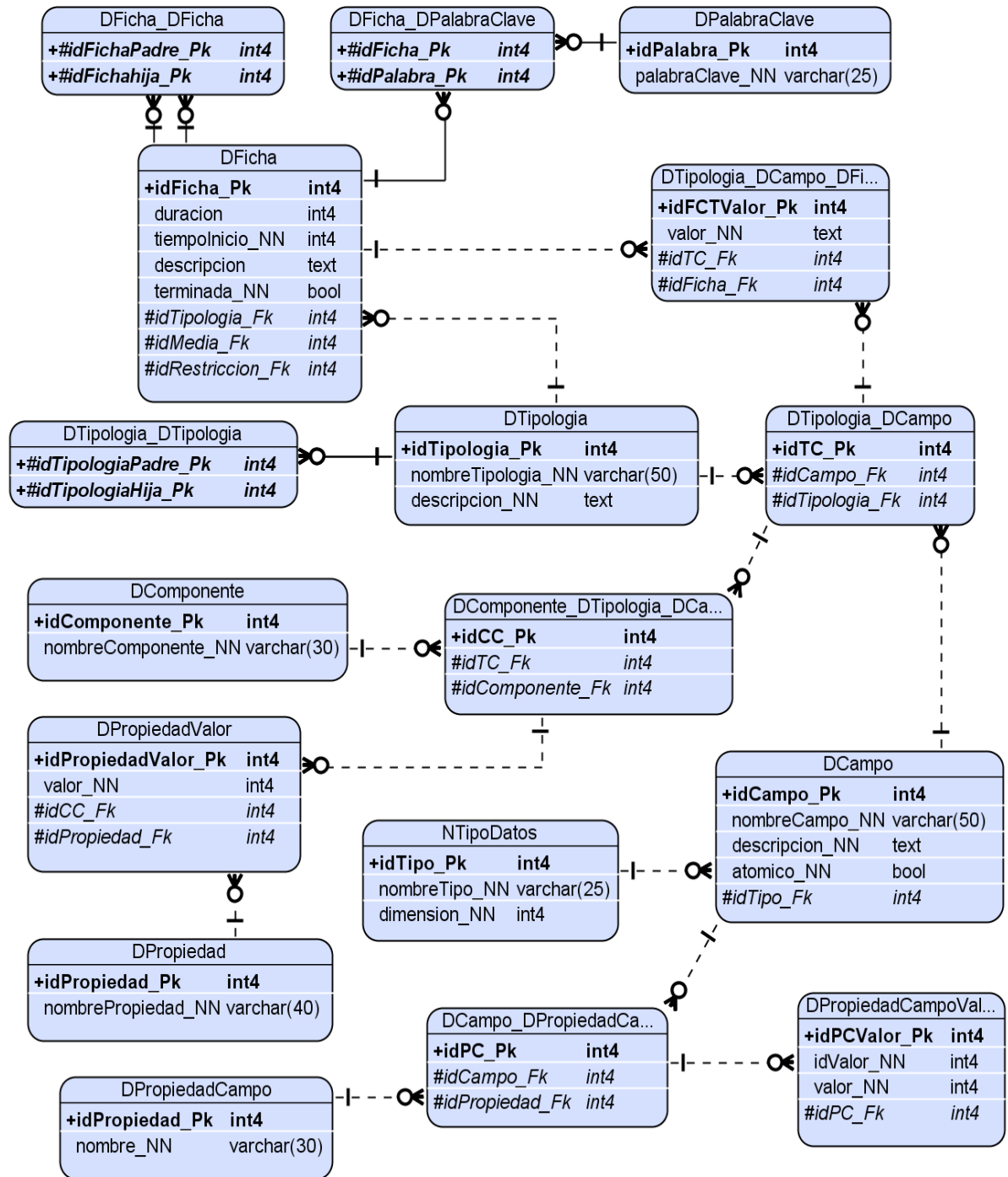


Figura 9. Modelo lógico de datos. Parte 3.

El sistema manipula información sensible en muchos casos, por lo que cada proyecto, media o ficha tiene un nivel de restricción para controlar los accesos a los



Capítulo 3: “Descripción y Análisis de la Solución”

materiales durante la ejecución de los diferentes procesos. En la figura 10 se muestra la estructura que posibilita este mecanismo.

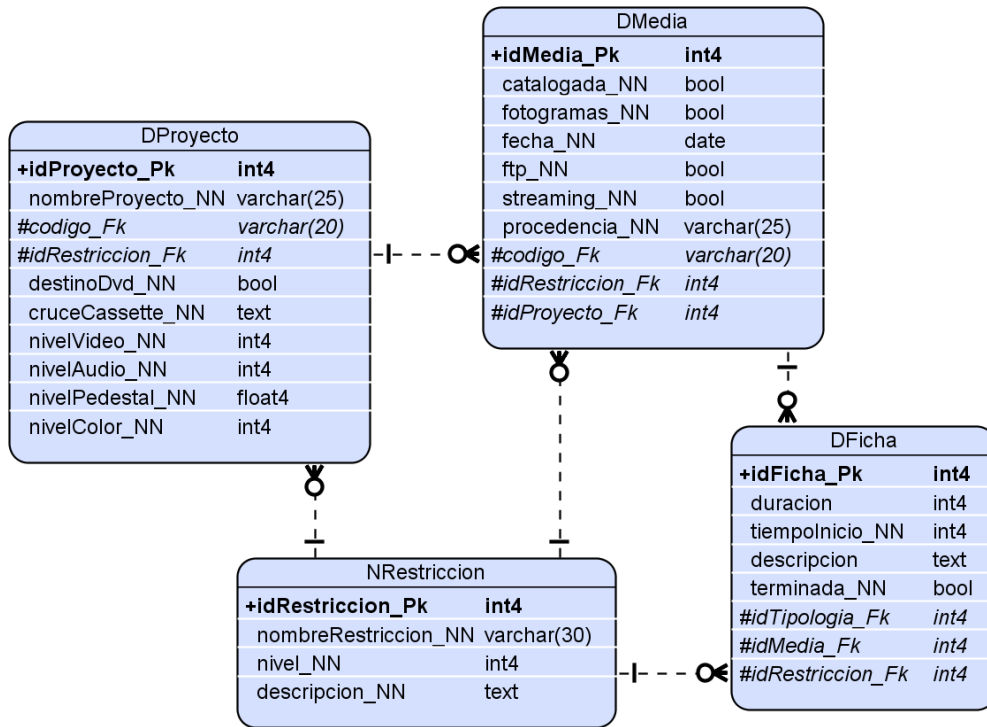


Figura 10. Mecanismo para controlar la restricción de la información.

3.5.2 Descripción de las tablas

En este epígrafe se describen las tablas del Modelo de datos. Las tablas que se crean como resultado de las relaciones se describen en los anexos.

Nombre de la Tabla: DUsuario		
Descripción de la Tabla: Almacena la información correspondiente a los usuarios del sistema.		
Atributo	Tipo de dato	Descripción
usuario_Pk	varchar(25)	Alias del usuario. Campo llave de la tabla.
nombre_NN	varchar(25)	Nombre del usuario.
apellidos_NN	varchar(60)	Apellidos del usuario.
passwd_NN	varchar(25)	Contraseña del usuario.
habilitado_NN	bool	Campo que indica si el usuario está



Capítulo 3: “Descripción y Análisis de la Solución”

		habilitado para acceder al sistema o no.
correo_NN	varchar(50)	Dirección de correo electrónico del usuario.
autenticado_NN	bool	Campo que indica si el usuario está actualmente conectado al sistema o no.
idArea_Fk	int4	Área a la que pertenece el usuario dentro de la empresa.
id_rol2_Fk	int4	Identificador del rol del usuario.

Tabla 17. Descripción de la tabla: DUsuario.

Nombre de la Tabla: DRol		
Descripción de la Tabla: Almacena los roles que tendrán los usuarios del sistema.		
Atributo	Tipo de dato	Descripción
idRol_Pk	int4	Campo llave de la tabla.
nombreRol_NN	varchar(25)	Nombre del rol.
descripcion_NN	text	Descripción del rol.

Tabla 18. Descripción de la tabla: DRol.

Nombre de la Tabla: DPermiso		
Descripción de la Tabla: Almacena los permisos de las acciones que se pueden realizar dentro del sistema. Por ejemplo: Permiso para reproducir un video.		
Atributo	Tipo de dato	Descripción
idPermiso_Pk	int4	Campo llave de la tabla.
nombrePermiso_NN	varchar(25)	Nombre del permiso.
idTipoPermiso_Fk	int4	Identificador del tipo de permiso.

Tabla 19. Descripción de la tabla: DPermiso.

Nombre de la Tabla: NTipoPermiso		
Descripción de la Tabla: Almacena los posibles tipos de permisos del sistema. Por ejemplo: Permiso de administración.		
Atributo	Tipo de dato	Descripción
idTipoPermiso_Pk	int4	Campo llave de la tabla.
nombreTipoPermiso_NN	varchar(25)	Nombre del tipo de permiso.

Tabla 20. Descripción de la tabla: NTipoPermiso.



Capítulo 3: “Descripción y Análisis de la Solución”

Nombre de la Tabla: NArea		
Descripción de la Tabla: Almacena los posibles tipos de permisos del sistema. Por ejemplo: Permiso de administración.		
Atributo	Tipo de dato	Descripción
idArea_Pk	int4	Campo llave de la tabla.
nombreArea_NN	varchar(25)	Nombre del área de la empresa.
descripcion_NN	text	Descripción del área de la empresa.

Tabla 21. Descripción de la tabla: NÁrea.

Nombre de la Tabla: DAutenticacion		
Descripción de la Tabla: Almacena todos los accesos al sistema por parte de los usuarios.		
Atributo	Tipo de dato	Descripción
id_Autenticacion_Pk	int4	Campo llave de la tabla.
fechaInicio_NN	date	Fecha en que el usuario inició la sección.
fechaFin_NN	date	Fecha en que el usuario finalizó la sección.
horaInicio_NN	time(7)	Hora en que el usuario inició la sección.
horaFin_NN	time(7)	Hora en que el usuario finalizó la sección.
usuario_Fk	int4	Identificador del usuario que se autentica.

Tabla 22. Descripción de la tabla: DAutenticacion.

Nombre de la Tabla: DPlanificacion		
Descripción de la Tabla: Almacena las planificaciones de tareas que se harán en el sistema.		
Atributo	Tipo de dato	Descripción
idPlanificacion_Pk	int4	Campo llave de la tabla.
prioridad_NN	int4	Determina la prioridad con que debe realizarse la tarea.
porCiento_NN	int4	Por ciento de cumplimiento de una tarea.
fechaInicio_NN	date	Fecha de inicio del intervalo de tiempo para realizar una tarea.



Capítulo 3: “Descripción y Análisis de la Solución”

fechaFin_NN	date	Fecha de fin del intervalo de tiempo para realizar una tarea.
fechaRealizacion_NN	date	Fecha de realización de la tarea.
idUsuario_Fk	varchar(25)	Identificador del usuario responsable de una tarea.
idTarea_Fk	int4	Identificador del tipo de tarea asignada.
idTrabajo_NN	int4	Campo que contiene el identificador de un proyecto si la tarea es de tipo edición. En caso de ser de tipo catalogación se corresponde con el identificador de la media a catalogar.

Tabla 23. Descripción de la tabla: DPlanificacion.

Nombre de la Tabla: NTipoTarea		
Descripción de la Tabla: Almacena los tipos de tarea a realizar dentro del sistema.		
Atributo	Tipo de dato	Descripción
idTarea_Pk	int4	Campo llave de la tabla.
tipoTarea_NN	varchar(30)	Nombre del tipo de tarea.

Tabla 24. Descripción de la tabla: NTipoTarea.

Nombre de la Tabla: DProyecto		
Descripción de la Tabla: Almacena los datos de los proyectos registrados en el sistema.		
Atributo	Tipo de dato	Descripción
id_proyecto_Pk	int4	Campo llave de la tabla.
nombreProyecto_NN	varchar(30)	Nombre del proyecto.
codigo_Fk	int4	Identificador del soporte en que se encuentra la media antes de ser digitalizada.
idRestriccion_Fk	int4	Identificador de la restricción asociada al proyecto.
destinoDVD_NN	bool	Dato específico necesario para realizar el



Capítulo 3: “Descripción y Análisis de la Solución”

		proyecto.
cruceCassette_NN	text	Dato específico necesario para realizar el proyecto.
nivelVideo_NN	int4	Dato específico necesario para realizar el proyecto.
nivelAudio_NN	int4	Dato específico necesario para realizar el proyecto.
nivelPedestal_NN	float4	Dato específico necesario para realizar el proyecto.
nivelColor_NN	int4	Dato específico necesario para realizar el proyecto.

Tabla 25. Descripción de la tabla: DProyecto.

Nombre de la Tabla: NFalla		
Descripción de la Tabla: Almacena las posibles fallas de los archivos detectadas en los proyectos. Por ejemplo, si la falla es de tipo video puede tener la falla nombrada efecto fantasma.		
Atributo	Tipo de dato	Descripción
idFalla_Pk	int4	Campo llave de la tabla.
nombreFalla_NN	varchar(125)	Nombre de la falla.
idTipoFalla_Fk	int4	Identificador del tipo de la falla.

Tabla 26. Descripción de la tabla: NFalla.

Nombre de la Tabla: NTipoFalla		
Descripción de la Tabla: Almacena el tipo de fallas que puede tener una media. Por ejemplo: falla de video, falla de audio.		
Atributo	Tipo de dato	Descripción
idTipoFalla_Pk	int4	Campo llave de la tabla.
nombreTipoFalla_NN	varchar(25)	El tipo de la falla.

Tabla 27. Descripción de la tabla: NTipoFallas.

Nombre de la Tabla: NIntensidad		
Descripción de la Tabla: Especifica el modo de intensidad de la falla (imperceptible,		



Capítulo 3: “Descripción y Análisis de la Solución”

claramente molesto, entre otros modos) y le da una calificación.

Atributo	Tipo de dato	Descripción
calificacion_Pk	int4	Campo llave de la tabla, representa la calificación del modo de intensidad.
nombreIntensidad_NN	varchar(50)	Nombre de la intensidad. Éste suele ser descriptivo.

Tabla 28. Descripción de la tabla: NIntensidad.

Nombre de la Tabla: NDuracion

Descripción de la Tabla: Especifica la duración de la falla en el material (Uno o varios instantes, durante varios períodos prolongados, entre otros modos) y le da una calificación.

Atributo	Tipo de dato	Descripción
calificacion_Pk	int4	Campo llave de la tabla, representa la calificación de la duración.
nombreDuracion_NN	varchar(50)	Nombre de la duración. Este nombre describe cuanto dura la falla.

Tabla 29. Descripción de la tabla: NDuracion.

Nombre de la Tabla: NRestriccion

Descripción de la Tabla: Almacena los datos de las restricciones de acceso que tienen los materiales.

Atributo	Tipo de dato	Descripción
idRestriccion_Pk	int4	Campo llave de la tabla.
nombreRestriccion_NN	varchar(30)	Nombre de la restricción.
nivel_NN	int4	Nivel de la restricción.
descripcion_NN	text	Descripción de la restricción.

Tabla 30. Descripción de la tabla: NRestriccion.

Nombre de la Tabla: NFalla_DProyecto

Descripción de la Tabla: Esta tabla es el resultado de la relación de muchos a muchos entre las tablas NFallas y DProyecto. Almacena los datos de una falla específica en un proyecto específico.



Capítulo 3: “Descripción y Análisis de la Solución”

Atributo	Tipo de dato	Descripción
idProFalla_Pk	int4	Campo llave de la tabla.
idFalla_Fk	int4	Identificador de la falla.
idProyecto_Fk	int4	Identificador del proyecto que contendrá la falla.
fecha_NN	date	Fecha en la que se detectó la falla en el proyecto.
obsevaciones_NN	text	Cualquier observación que quiera hacer el usuario sobre la falla.
tiempolnicio_NN	int4	Tiempo de la media en la que aparece la falla.
tiempoFin_NN	int4	Tiempo en la que desaparece la falla en la media.
intensidad_Fk	int4	Calificación del modo de intensidad de la falla.
duracion_Fk	int4	Calificación de la duración de la falla.

Tabla 31. Descripción de la tabla: NFalla_DProyecto.

Nombre de la Tabla: DSolicitud		
Descripción de la Tabla: Registra las solicitudes de materiales que se le hacen a la entidad.		
Atributo	Tipo de dato	Descripción
idSolicitud_Pk	int4	Campo llave de la tabla.
externa_NN	bool	Campo que determina si la solicitud es interna o externa, (false si es interna y true si es externa).
atendida_NN	bool	Campo que determina si la solicitud fue atendida o no, (true si fue atendida y false si no lo fue).
objetivo_NN	text	Campo para describir el objetivo que tiene la solicitud.

Tabla 32. Descripción de la tabla: DSolicitud.

Nombre de la Tabla: DCliente		
Descripción de la Tabla: Almacena los datos de los clientes que han hecho las		



Capítulo 3: “Descripción y Análisis de la Solución”

solicitudes de materiales.		
Atributo	Tipo de dato	Descripción
idCliente_Pk	int4	Campo llave de la tabla.
nombreCliente_NN	varchar(25)	Nombre del cliente.
CI_NN	varchar(11)	Número de identidad del cliente.
apellidos_NN	varchar(60)	Apellidos del cliente.
idEntidad_Fk	int4	Identificador de la entidad a la que pertenece el cliente.

Tabla 33. Descripción de la tabla: DCliente.

Nombre de la Tabla: NEntidad		
Descripción de la Tabla: Tabla que contiene las entidades a la que pertenecen los clientes que han realizado solicitudes de material.		
Atributo	Tipo de dato	Descripción
idEntidad_Pk	int4	Campo llave de la tabla.
nombreEntidad_NN	varchar(50)	Nombre de la entidad.

Tabla 34. Descripción de la tabla: NEntidad.

Nombre de la Tabla: NPersonaAutoriza		
Descripción de la Tabla: Tabla que registra las personas que autorizan solicitudes de material.		
Atributo	Tipo de dato	Descripción
idPersonaAutoriza_Pk	int4	Campo llave de la tabla.
nombre_NN	varchar(25)	Nombre de la persona.
apellidos_NN	varchar(60)	Apellidos de la persona que autoriza.
cargo_NN	varchar(50)	Cargo que ocupa la persona en la entidad.

Tabla 35. Descripción de la tabla: NPersonaAutoriza.

Nombre de la Tabla: NPersonaAutoriza_DSolicitud		
Descripción de la Tabla: Tabla que registra las solicitudes que han sido atendidas.		



Capítulo 3: “Descripción y Análisis de la Solución”

Atributo	Tipo de dato	Descripción
idPersonaAutoriza_Pk	int4	Identificador de la persona que autoriza a atender la solicitud.
idSolicitud_Pk	int4	Identificador de la solicitud atendida.
fecha_NN	date	Fecha en que se atiende.

Tabla 36. Descripción de la tabla: NPersonaAutoriza_DSolicitud.

Nombre de la Tabla: DMedia		
Descripción de la Tabla: Tabla para almacenar los datos que corresponden a las medias que se encuentran físicamente y que gestionará el sistema con los procesos de edición y catalogación.		
Atributo	Tipo de dato	Descripción
idMedia_Pk	int4	Campo llave de la tabla.
catalogada_NN	bool	Campo que determina si la media fue catalogada, (true si fue catalogada, false si no).
fotogramas_NN	bool	Campo que determina si a la media se le extrajeron sus fotogramas, (true si fueron extraídos sus fotogramas, false si no).
fecha_NN	date	Fecha de inserción de la media en el servidor.
ftp_NN	bool	Campo que determina si la media está disponible en el servidor FTP (true si está, false si no).
streaming_NN	bool	Campo que determina si la media está lista para <i>streaming</i> (true si está, false si no).
codigo_Fk	int4	Identificador del soporte en que se encuentra la media antes de ser digitalizada.
procedencia_NN	varchar(25)	Procedencia de la media.
idRestriccion_Fk	int4	Identificador de la Restricción que puede tener la media.
idProyecto_Fk	int4	Identificador del proyecto al que esta



Capítulo 3: “Descripción y Análisis de la Solución”

		asociado la media.
--	--	--------------------

Tabla 37. Descripción de la tabla: DMedia.

Nombre de la Tabla: NTipoSoporte		
Descripción de la Tabla: Tabla que contiene los tipos de soporte en los que se pueden encontrar las medias.		
Atributo	Tipo de dato	Descripción
idTipoSoporte_Pk	int4	Campo llave de la tabla.
nombreTipoSoporte_NN	varchar(30)	Nombre de la soporte, puede ser cinta, cassette, disco duro, entre otros.

Tabla 38. Descripción de la tabla: NTipoSoporte.

Nombre de la Tabla: DSoporte		
Descripción de la Tabla: Tabla que contiene los soportes en los que se pueden encontrar las medias.		
Atributo	Tipo de dato	Descripción
codigo_Pk	int4	Código del soporte, Es el número de serie del soporte. Campo llave de la tabla.
idTipoSoporte_Fk	varchar(30)	Identificador del tipo de soporte.

Tabla 39. Descripción de la tabla: DSoporte.

Nombre de la Tabla: DFicha		
Descripción de la Tabla: Tabla para almacenar las fichas que caracterizan un material.		
Atributo	Tipo de dato	Descripción
idFicha_Pk	int4	Campo llave de la tabla.
duracion_NN	int4	Duración de la media.
tiempoInicio_NN	int4	Campo que almacena el tiempo de inicio de la secuencia virtual que se quiere definir como una media independiente o puede ser el tiempo inicial del material original.
descripcion_NN	text	Descripción del material.



Capítulo 3: “Descripción y Análisis de la Solución”

terminada_NN	bool	Campo que determina si se terminó completamente la catalogación de la media, (True si se terminó, false si no)
idTipologia_Fk	int4	Identificador de la tipología a la que está asociada la ficha.
idRestriccion_Fk	int4	Identificador de la restricción que tiene la ficha.
idMedia_Fk	int4	Identificador de la media a la que está asociada la ficha.

Tabla 40. Descripción de la tabla: DFicha.

Nombre de la Tabla: DTipologia		
Descripción de la Tabla: Almacena las tipologías creadas en el sistema.		
Atributo	Tipo de dato	Descripción
idTipologia_Pk	int4	Campo llave de la tabla.
nombreTipologia_NN	varchar(50)	Nombre de la tipología.
descripcion_NN	text	Descripción de la tipología.

Tabla 41. Descripción de la tabla: DTipologia.

Nombre de la Tabla: DPalabraClave		
Descripción de la Tabla: Almacena las palabras claves que pertenecen a las fichas que servirán para realizar búsquedas posteriormente.		
Atributo	Tipo de dato	Descripción
idPalabra_PK	int4	Campo llave de la tabla.
palabraClave_NN	varchar(25)	Palabra clave.

Tabla 42. Descripción de la tabla: DPalabraClave.

Nombre de la Tabla: DCampo		
Descripción de la Tabla: Almacena los campos que pertenecen a las distintas tipologías del sistema.		
Atributo	Tipo de dato	Descripción
idCampo_Pk	int4	Campo llave de la tabla.
nombreCampo_NN	varchar(50)	Nombre del campo.



Capítulo 3: “Descripción y Análisis de la Solución”

descripcion_NN	text	Descripción del campo.
atomico_NN	bool	Campo que determina si el campo puede ser multivaluado o no.
idTipo_Fk	int4	Identificador del tipo de datos del campo.

Tabla 43. Descripción de la tabla: DCampo.

Nombre de la Tabla: NTipoDatos		
Descripción de la Tabla: Almacena los tipos de datos de los campos de las tipologías.		
Atributo	Tipo de dato	Descripción
idTipo_Pk	int4	Campo llave de la tabla.
nombreTipo_NN	varchar(25)	Nombre del tipo de dato.
dimension_NN	int4	Representa la dimensión que puede alcanzar el valor del campo. Por ejemplo, si el nombre del tipo es string, la dimensión puede ser 25.

Tabla 44. Descripción de la tabla: NTipoDatos.

Nombre de la Tabla: DPropiedadCampo		
Descripción de la Tabla: Almacena las propiedades o atributos de los campos que son compuestos.		
Atributo	Tipo de dato	Descripción
idPropiedad_Pk	int4	Campo llave de la tabla.
nombre_NN	varchar(30)	Nombre de la propiedad o atributo.

Tabla 45. Descripción de la tabla: DPropiedCampo.

Nombre de la Tabla: DCampo_DPropiedadCampo		
Descripción de la Tabla: Se genera a partir de la relación muchos a muchos que existe entre las tablas DCampo y DPropiedCampo, representando las propiedades que pertenecen a los campos compuestos.		
Atributo	Tipo de dato	Descripción
idPC_Pk	int4	Campo llave de la tabla. Identifica cada relación DCampo y DPropiedCampo simplificando el entendimiento y la duplicación de información en



Capítulo 3: “Descripción y Análisis de la Solución”

		relaciones posteriores.
idPropiedad_Fk	int4	Identificador de la propiedad o el atributo del campo.
idCampo_Fk	int4	Identificador del campo.

Tabla 46. Descripción de la tabla: DCampo_DPropiedadCampo.

Nombre de la Tabla: DPropiedadCampoValor		
Descripción de la Tabla: Almacena los valores de las propiedades o atributos de los campos		
Atributo	Tipo de dato	Descripción
idPCValor_Pk	int4	Identificador de la tabla.
idvalor	int4	Identificador de un valor específico de una propiedad de un campo, este id se usa para encontrar el valor desde la tabla DTipologia_DCampo_Dficha.
idPC_Fk	int4	Identificador de la propiedad de un campo específico.
valor_NN	text	Valor de la propiedad.

Tabla 47. Descripción de la tabla: DPropiedCampoValor.

Nombre de la Tabla: DTipologia_DCampo		
Descripción de la Tabla: Se crea de la relación muchos a muchos entre las tablas tipologías y campos. Representa los campos que pertenecen a una tipología.		
Atributo	Tipo de dato	Descripción
idTC_Pk	int4	Campo llave de la tabla. Identifica cada relación DTipologia_DCampo simplificando el entendimiento y la duplicación de información en relaciones posteriores.
idCampo_Fk	int4	Identificador del campos
idTipologia_Fk	int4	Identificador de la tipología.

Tabla 48. Descripción de la tabla: DTipologia_DCampo.

Nombre de la Tabla: DTipologia_DCampo_DFicha		
Descripción de la Tabla: Almacena los valores de cada campo de una tipología con		



Capítulo 3: “Descripción y Análisis de la Solución”

la que se cataloga una ficha.

Atributo	Tipo de dato	Descripción
idFCTValor_Pk	int4	Campo llave de la tabla.
idTC_Fk	int4	Identificador del campo específico de una tipología.
idFicha_Fk	int4	Identificador de la Ficha.
valor_NN	text	Campo que almacena el valor del campo de una tipología para una ficha específica. En caso de que el campo sea compuesto almacena la referencia a sus valores en la tabla DPropiedadCampoValor.

Tabla 49. Descripción de la tabla: DTipologia_DCampo_DFicha.

Nombre de la Tabla: DComponente

Descripción de la Tabla: Almacena los componentes de las interfaces visuales de los formularios de cada una de las tipologías.

Atributo	Tipo de dato	Descripción
idComponente_Pk	int4	Campo llave de la tabla.
nombreComponente_NN	varchar(30)	Nombre del componente.

Tabla 50. Descripción de la tabla: DComponente.

Nombre de la Tabla: DPropiedad

Descripción de la Tabla: Almacena las propiedades de los componentes.

Atributo	Tipo de dato	Descripción
idPropiedad_Pk	int4	Campo llave de la tabla.
nombrePropiedad_NN	varchar(40)	Propiedad del componente.

Tabla 51. Descripción de la tabla: DPropiedad.

Nombre de la Tabla: DComponente_DTipologia_DCampo

Descripción de la Tabla: Almacena la información que vincula los campos de una tipología con los componentes con que serán mostrados.

Atributo	Tipo de dato	Descripción
idCC_Pk	int4	Campo llave de la tabla.
idTC_Fk	int4	Identificador de un campo de una tipología.



Capítulo 3: “Descripción y Análisis de la Solución”

idComponente_Fk	int4	identificador del componente
-----------------	------	------------------------------

Tabla 52. Descripción de la tabla: DComponente_DTipologia_DCampo.

Nombre de la Tabla: DPropiedadValor		
Descripción de la Tabla: Almacena los valores de las propiedades de cada uno de los componentes pertenecientes a cada uno de los campos de las tipologías del sistema.		
Atributo	Tipo de dato	Descripción
idPropiedadValor_Pk	int4	Campo llave de la tabla.
id_CC_Fk	int4	Identificador del componente perteneciente a un campo de una tipología específica.
idPropiedad_Fk	int4	Identificador de una propiedad.
valor_NN	text	Valor de la propiedad.

Tabla 53. Descripción de la tabla: DPropiedadValor.

Otras tablas resultantes de las relaciones en la base de datos, se encuentran descritas en el **Anexo II** de este documento.

3.6 Conclusiones parciales

El diseño propuesto para la base de datos satisface los requisitos funcionales del Sistema de Captura y Catalogación de Medias, resuelve las insuficiencias de versiones anteriores. A lo largo del capítulo se explica su estructura y ventajas que lo hacen adaptable y escalable a diferentes entornos de ejecución.

En este punto se tiene una descripción clara y precisa del funcionamiento y de cada una de las tablas de la base de datos, lo que servirá de documentación y contribuirá en la toma de decisiones de líderes y desarrolladores del sistema.



Capítulo 4 “Validación de la Solución Propuesta”

4.1 Introducción

Para lograr un diseño eficaz de una base de datos, hay que garantizar la consistencia, integridad y seguridad de sus datos.

En este capítulo se describe el proceso de validación del diseño de la base de datos desarrollada a través de una validación teórica, mediante la cual se normaliza la base de datos y una validación funcional donde su funcionamiento se pone a prueba.

4.2 Validación teórica del diseño realizado

Para realizar la validación teórica del diseño de una base de datos se incluye fundamentalmente un análisis detallado de la integridad de la información, característica que asegura la calidad de almacenamiento y disponibilidad de los datos. Con su tratamiento se evita errores de entrada introducidos por los usuarios descuidados o cualquier otra circunstancia de intento de violar la información existente en la base de datos.

4.2.1 Integridad

“La integridad de los datos consiste en garantizar la no contradicción entre los datos almacenados en la base de datos de modo que, en cualquier momento del tiempo, los datos almacenados sean correctos, es decir, que no se detecte inconsistencia entre los ellos. Está relacionada con la minimización de redundancia, ya que es más fácil garantizar la integridad si se elimina la redundancia”. **(Mato, 2005)**.

La integridad es uno de los factores más importantes a la hora de realizar el diseño de una base de datos. Se divide en varios aspectos como son:

- ❖ **“Integridad referencial:** Garantiza interrelaciones válidas entre entidades. Implica que los datos sean correctos, sin duplicaciones, pérdida de datos o relaciones mal resueltas. Todas las bases de datos relacionales incluyen esta propiedad pues el software gestor es responsable de su cumplimiento”.
- ❖ **“Integridad de Dominio:** La integridad de dominio viene dada por la validez de las entradas de los datos para una columna determinada. Se puede exigir



Capítulo 4: “Validación de la Solución Propuesta”

la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones *CHECK*, o el intervalo de valores posibles mediante restricciones *FOREIGN KEY*, definiciones *DEFAULT*, definiciones *NOT NULL*.” (Monge).

- ❗ **“Integridad de la Entidad:** define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones *UNIQUE*, o restricciones *PRIMARY KEY*”. (Mora).

En el caso de la base de datos desarrollada, la integridad referencial es verificada por el Sistema Gestor de Base de Datos con excepción del caso de las planificaciones. En la tabla DPlanificación existe el campo “idTrabajo_NN” que almacena los identificadores del contenido de trabajo de una tarea específica, si el campo “idTarea_Fk” indica que la tarea es de edición, “idTrabajo_NN” coincidirá con algún “idProyecto_Pk” en la tabla DProyecto. Ocurre lo mismo con la tabla DMedia en caso de que la tarea sea de catalogación. Se hace evidente que se necesita un proceso de validación en tiempo de ejecución para velar por la integridad referencial para la tabla Planificación. Los dominios de los datos han sido definidos para que cada campo en la base de datos tome el valor que le corresponde salvo la excepción de los valores de los campos de las tipologías, los cuales se almacenan en la tabla DTipologia_DCampo_DFicha pero debido que cada campo definido en la tabla DCampo tiene un tipo de datos asociado se podrá conocer su dominio cuando el sistema este en funcionamiento. En cada una de las tablas se define una llave primaria (incluso en tablas creadas a partir de relaciones) por lo que cada fila tiene un índice único, luego cada fila es una entidad independiente.

4.2.2 Normalización de la base de datos

La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes que eviten las anomalías de actualización.

“La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información”. (Mato, 2005).



Capítulo 4: “Validación de la Solución Propuesta”

Ventajas:

- Evita anomalías en la actualización.
- Mejora la independencia de los datos, permitiendo realizar extensiones de la base de datos, afectando muy poco, o nada, a los programas de aplicación existentes que accedan la base de datos.

“La normalización involucra varias fases que se realizan en orden. La realización de la 2da fase supone que se ha concluido la 1ra y así sucesivamente. Tras completar cada fase se dice que la relación está en:

💡 **Primera Forma Normal (1FN):** Una relación está en 1ra FN si cumple con la propiedad de que sus dominios no contengan elementos que a su vez sean conjuntos.

- La relación no incluye elementos repetitivos.
- Toda relación normalizada, o sea, con valores atómicos de los atributos.

💡 **Segunda Forma Normal (2FN):** Está basada en el concepto de dependencia funcional total, es decir que los atributos no primos dependen totalmente de los atributos llaves y debe de cumplir con lo establecido en la 1ra FN. Se aplica solamente a los esquemas de relación que tienen claves primarias compuestas por dos o más atributos. Si un esquema de relación está en 1FN y su clave primaria es simple entonces está en segunda forma normal.

- Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.
- Para la transformación de una relación de 1FN a 2FN hay que eliminar las dependencias parciales de su clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se descomponen en una nueva relación con una copia de los atributos de la clave primaria de los que dependen.

💡 **Tercera Forma Normal (3FN):** Está basada en el concepto de dependencias transitivas. Se considera que está en 3FN si:



Capítulo 4: “Validación de la Solución Propuesta”

- Si y sólo si, la relación está en 2da FN y los atributos no llaves son independientes de cualquier otro atributo no llave primaria.

Esto es lo mismo que decir que se deben eliminar las dependencias transitivas de atributos no llaves respecto a la llave primaria, con la relación en 2FN.

- Forma Normal de Boyce-Codd (FNBC): Esta forma normal es una refinación de la 3ra FN un poco más estricta (por lo que para estar en la FNBC hay que estar en la 3ra FN). Una relación R está en FNBC si y sólo si cada determinante de las dependencias funcionales es una llave candidata.
- Existen, además, 4FN y 5FN.

Las relaciones en 1FN son un subconjunto del universo de todas las relaciones posibles. Las relaciones en 2FN son un subconjunto de las que están en 1FN y así sucesivamente”. (Mato, 2005).

La base de datos del Sistema de Captura y Catalogación de Medias actualmente se encuentra en 3era FN, porque cumple con las especificaciones de las tres formas expuestas anteriormente, es decir en ella no existen atributos multivaluados, dependencias transitivas entre las relaciones y redundancia de la información. La 3era FN es la más usada casi en la totalidad de los productos que utilizan bases de datos, puesto que éstas garantizan poca o una redundancia casi nula de información, además logra eficacia en las consultas, factor fundamental en el diseño y la implementación de una base de datos.

4.2.3 Análisis de la redundancia de la información

La redundancia de datos es aquella información duplicada o almacenada varias veces en la misma base de datos. Esto dificulta la tarea de modificación de datos y es el motivo más frecuente de inconsistencia de los mismos. Además requiere un mayor espacio de almacenamiento, que influye en un mayor costo y tiempo de acceso a los datos. Los procesos de normalización mejoran en buena medida el comportamiento de este parámetro.



Capítulo 4: “Validación de la Solución Propuesta”

Después de llevar la base de datos del Sistema de Captura y Catalogación de Medias a 3FN, es decir, al normalizarla, queda libre de redundancias, ya que se elimina completamente la presencia de datos repetidos innecesariamente.

4.2.4 Análisis de la seguridad de la base de datos

La seguridad es un punto esencial en las bases de datos para evitar ataques e impedir cualquier acceso no autorizado, con la intención de modificar, usar y/o difundir información almacenada en las bases de datos, ya sea por error del usuario o un ataque intencional.

Los Sistema Gestores de Base de Datos actuales, garantizan en cierta medida que el control de acceso por parte de los usuarios a la base de datos se realice de acuerdo a los privilegios que tengan asignados y los lugares donde ellos se encuentren ubicados, y además los datos manejados deben estar encriptados, garantizando así su seguridad.

En el caso de *PostgreSQL* permite realizar configuraciones para controlar los permisos de los usuarios. Tal es el caso del archivo *pg_hba.conf* que permite configurar los permisos de conexión para los host y los usuarios de la base de datos, definiéndose qué direcciones de IP tendrán acceso, y en qué modo se conectarán (conexión sin contraseña, o con validación del usuario y la contraseña para conectarse); del archivo *pg_ident.conf*, que permite configurar qué usuarios tendrán acceso a determinada base de datos; así como configurar los accesos a las tablas, mediante los comandos *GRANT* y *REVOKE* para permitir o denegar los permisos, respectivamente.

Para seguridad del Sistema de Captura y Catalogación de Medias, se definió dar control total de todas las tablas de la base de datos solamente a los usuarios que respondieran al Rol de Administrador, y estos usuarios serán los que podrán insertar, modificar o eliminar los nuevos usuarios.

4.3 Validación funcional del diseño realizado

Para comprobar que la base de datos para el Sistema de Captura y Catalogación de Medias cumple con los requisitos funcionales definidos en el proyecto, antes de dar por terminado su proceso de diseño y creación se le realizan pruebas en las que se simula una carga de producción real y se ejecutan consultas complejas, lo que permite observar cómo se comporta la base de datos.



Capítulo 4: “Validación de la Solución Propuesta”

4.3.1 Pruebas de volumen

Las pruebas de volumen se realizan con el objetivo de analizar el comportamiento de la base de datos, con volúmenes de datos almacenados lo más similar posible a los esperados en la ejecución real del sistema.

Se realizó el llenado voluminoso de las tablas de la base de datos de manera dinámica con *EMS Data Generator for PostgreSQL*, herramienta que permite generar datos para una o varias tablas a la vez, definir el rango de valores admisibles para cada campo, así como posibilitar generar los datos a través de consultas SQL o de escogerlos por medio de listas de valores. Esta herramienta, respeta la integridad referencial, ya que genera los datos que provienen de otras tablas para evitar errores.

4.3.2 Consultas y tiempo de respuesta

Otra de las pruebas que se le realizan a las bases de datos, son las que chequean su comportamiento frente a las consultas más frecuentes y complejas que se realizarán a la misma, para verificar que su comportamiento sea eficaz. Para realizar las consultas y ver el tiempo de respuesta se usó la herramienta *EMS Query for PostgreSQL*.

La base de datos se pobló con:

- 20 tuplas en DTipologia.
- 8 tuplas en NTipoDatos.
- 4 tuplas en NRestriccion.
- 140 tuplas en DCampo.
- 300 relaciones entre DCampo y DTipologia.
- 50 tuplas en DComponente.
- 360 tuplas en DPropiedades (propiedades de los componentes).
- 5000 tuplas en DPropiedadesValor (valores de los componentes asociados a cada campo de una tipología).
- 100 tuplas en Dpropiedadcampo (propiedades de los campos).



Capítulo 4: “Validación de la Solución Propuesta”

- 🔗 3000 tuplas en Dpropiedadcampovalor
- 🔗 100000 tuplas en DFicha.
- 🔗 500000 valores asociados a campos de fichas, en la tabla DTipologia_DCampo_DFicha.

Las consultas que se realizaron en este tipo de pruebas a la base de datos para el Sistema de Captura y Catalogación de Medias se muestran a continuación:

La primera consulta realizada extrae los datos de la ficha de una media que ya ha sido catalogada.

```
SELECT dtipologia.nombretipologia_nn AS tipologia,
ntipodatos.nombretipo_nn AS tipo,
dcampo.nombrecampo_nn AS nombrecampo,
dcampo.atómico_nn AS atómico,
dtipologia_dcampo_dficha.valor_nn AS valor
FROM (((((dficha JOIN dtipologia_dcampo_dficha USING (idficha_fk)) JOIN
dtipologia_dcampo USING (idtc_fk)) JOIN dcampo USING (idcampo_fk)) JOIN
dtipologia USING (idtipologia_pk)) JOIN ntipodatos USING (idtipo_pk))
WHERE (dficha.idficha_fk = 4);
```

Para el volumen de datos expuesto el tiempo de respuesta para saber los datos de catalogación de una ficha dada, fue de 282 ms y la búsqueda mostró 5 resultados.

La siguiente consulta extrae los datos de los campos que son compuestos una vez que se encuentra su id de valor, también se obtienen los nombres de las propiedades asociadas a cada campo.

```
SELECT dpropiedadcampo.nombre_nn AS nombre,
dpropiedadcampovalor.valor_nn AS valor FROM dpropiedadcampo,
dpropiedadcampovalor, dcampo_dpropiedadcampo
WHERE((dpropiedadcampo.idpropiedad_pk=dcampo_dpropiedadcampo.idpropiedad
_fk) AND (dcampo_dpropiedadcampo.idpc_pk = propiedadcampovalor.idpc_fk))
AND (dpropiedadcampovalor.idvalor_nn = 217445898));
```



Capítulo 4: “Validación de la Solución Propuesta”

Para el volumen de datos expuesto el tiempo de respuesta para saber los datos de un campo compuesto sabiendo su ubicación es de 47 ms y generó 8 resultados.

Una vez obtenidos los datos de la ficha y sus posibles campos compuestos sólo queda obtener los datos de los componentes de cada campo de una tipología específica.

```

SELECT tc.idtc_fk AS idcampo,
c.nombrecomponente_nn AS componente,
p.nombrepropiedad_nn AS propiedad,
pv.valor_nn AS valor

FROM dtipologia_dcampo tc, dcomponente c, dpropiedad p,
dpropiedadvalor pv, dcomponente_dtipologia_dcampo ctc

WHERE
((((p.idpropiedad_pk = pv.idpropiedad_fk) AND
(pv.idcc_fk = ctc.idcc_pk)) AND
(ctc.idcomponente_fk = c.idcomponente_pk)) AND
(ctc.idtc_fk = tc.idtc_fk)) AND
(tc.idtc_fk = ANY (ARRAY [23, 24, 25, 26, 27, 28, 29])))
ORDER BY tc.idtc_fk;

```

Para el volumen de datos el tiempo empleado para recuperar los datos de los componentes fue de 78 ms y se obtuvieron 116 tuplas.

4.3.2.1 Mejora del tiempo en búsquedas sobre texto.

Un aspecto clave en el Sistema de Captura y Catalogación de Medias son las consultas para localizar u obtener información de los materiales ya catalogados que estará guardada en la base de datos. Las descripciones de cada una de las fichas que representan las medias, en no pocas ocasiones contienen la información clave para estas búsquedas.

En un volumen de datos como el usado para las pruebas, la siguiente consulta tarda 1.5 segundos en buscar las fichas que contienen en su descripción las palabras “historia” y “cubana”.



Capítulo 4: “Validación de la Solución Propuesta”

```
SELECT public.dficha.idficha_fk
FROM public.dficha
WHERE public.dficha.descripcion LIKE '%historia%' AND
public.dficha.descripcion LIKE '%cubana%';
```

Como se puede apreciar el tiempo que demora esta búsqueda es elevado, teniendo en cuenta que el sistema procesará cientos de miles de fichas en un momento determinado. Según el sitio oficial de PostgreSQL¹⁰, El operador LIKE al igual que ~, ~* e ILIKE no se auxilia de índices por lo que deben procesar todas las fichas en cada búsqueda realizada en el sistema; Estos operadores tampoco poseen apoyo lingüístico y las expresiones regulares no son suficientes ya que no pueden manejar las palabras derivadas o con una semántica igual; por ejemplo, si se usan las palabras “cumple” y “satisface” en la búsqueda, al usar estos operadores se perderían los documentos que contengan “satisface” si se usa como filtro “cumple” y viceversa, aunque es probable que le gustaría encontrar en los resultados de la búsqueda los documentos que contienen la palabra “satisface”. Es posible utilizar el operador OR para buscar múltiples formas derivadas, pero esto es tedioso y propenso a errores (algunas palabras pueden tener varios miles de derivados).

En el sitio oficial de PostgreSQL se explica además que dicho gestor en su versión 8.4 agrega un mecanismo para realizar búsquedas de texto denominado “Búsqueda de Texto Completo” (*Full Text Search*, FTS por sus siglas en inglés) que incorpora una serie de procedimientos como son: documentos pre-procesados, que almacena en un tipo de datos *ts_vector* y una función para la búsqueda sobre esta estructura específica llamada *ts_query* que representa consultas pre-procesadas. La solución para reducir el tiempo de búsqueda sobre las descripciones de las fichas de las medias será: procesar mediante este mecanismo el campo descripción de la ficha haciendo una copia en un nuevo campo que se agrega a la tabla con el formato *ts_vector*. Este paso brinda ventajas: al pre-procesar el texto con FTS se obtiene apoyo lingüístico dado que el *ts_vector* guardado omite las llamadas “*stop word*”¹¹; posee un mapa de sinónimos para cada

¹⁰ Sitio Oficial de PostgreSQL: <http://www.postgresql.org>. Consultado en junio del 2011.

¹¹ Un *stop word* o palabra irrelevante puede ser una palabra con significado en un idioma determinado o un token sin significado lingüístico.



Capítulo 4: “Validación de la Solución Propuesta”

palabra simple; contiene un diccionario de sinónimos que permite mapear frases a una palabra simple y variaciones de una palabra. Una vez satisfecho los requisitos de búsquedas sobre texto solo quedaría asegurar la rapidez en el tiempo de respuesta, para ello PostgreSQL usa los índices inversos generalizados (*Generalized Inverted Index*, GIN por sus siglas en inglés) para campos de tipo *ts_vector*. Para aplicar esta operación habrá que agregar la siguiente sentencia al DDL ¹²de la base de datos.

```
CREATE INDEX IndiceFicha ON ficha USING GIN (descripcionTSV);
```

El tiempo de respuesta fue reducido a 35ms con la consulta resultante al combinar estas herramientas:

```
SELECT * FROM Ficha
WHERE descripcionTSV @@ to_tsquery ('Historia & Cubana');
```

Como se puede apreciar esta sentencia es más pequeña, legible y entendible a los programadores, además que FTS posee toda una gama de funciones adicionales que hacen mas extensibles y configurables todas sus potencialidades.

Otro punto clave en SCCM es la búsqueda por palabras claves la cual será demorada cuando exista un gran volumen de datos, sin embargo, al guardar en una tabla todas las palabras claves de cada una de las fichas con una estructura como la mostrada en la figura 11.

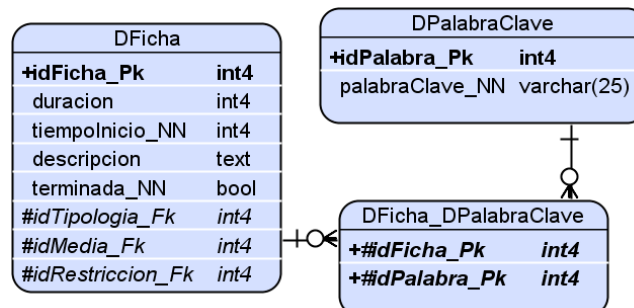


Figura 11 Estructura para agilizar las búsquedas de fichas a través de palabras claves.

¹² DDL (*Data Definition Language*, por sus siglas en inglés, Lenguaje de definición de datos).



Capítulo 4: “Validación de la Solución Propuesta”

Si se establece un índice para los identificadores que intervienen en la relación de muchos a muchos, al aplicar la siguiente sentencia de SQL se sabrá qué fichas pertenecen a la descripción de una o varias palabras claves en un tiempo promedio de 47 ms:

```
SELECT f.idficha_fk AS idFicha
FROM public.dficha f JOIN public.dficha_dpalabraclave fp USING
(idficha_fk) JOIN public.dpalabraclave p USING (idpalabra_pk)
WHERE p.palabraclave_nn = 'historia'
AND p.palabraclave_nn = 'cubana';
```

Con estos mecanismos el Sistema de Captura y Catalogación de Medias podrá disminuir el tiempo de respuesta en búsquedas dentro del campo descripción de las fichas y con palabras claves exponencialmente.

Estas potencialidades de SQL podrán ser aplicadas en otras sentencias complejas lo que posibilita un sistema robusto, escalable y adaptable a cualquier entorno.

4.4 Conclusiones parciales

Después de realizar las pruebas de volumen en la base de datos propuesta, no surgieron problemas, ya sean de límites de volumen de datos, desbordamientos de columnas, atributos o errores con los tipos de datos. Esto garantiza que el diseño de las estructuras de la base de datos y el gestor utilizado para el desarrollo de la misma soporta el almacenamiento de los niveles de información requeridos para el comienzo del funcionamiento de la base de datos en el Sistema de Captura y Catalogación de Medias. Además frente a las consultas más frecuentes y las que más información deben procesar se observó que el tiempo de respuesta estuvo por debajo de los 300 milisegundos y en algunos casos no demoró la decima parte de segundo. Estos resultados aseguran que la base de datos puede gestionar estos volúmenes de información de una manera eficaz.



Conclusiones Generales

El objetivo general de esta investigación se cumplió, pues se desarrolló el diseño e implementación de una base de datos que permite al Sistema de Captura y Catalogación de Medias alcanzar la centralización, seguridad y manejo eficaz de sus datos. Conjuntamente:

- Se logró que la base de datos creada sea adaptable y escalable, elementos imprescindibles, dado que no existe un cliente específico para el producto.
- Se garantizó la integridad normalizando su estructura hasta la 3ra Forma Normal, procedimiento que evade la redundancia e incoherencia en los datos.
- Se demostró mediante pruebas que la base de datos diseñada está en condiciones de ser puesta en práctica, pues soporta los volúmenes de datos que serán almacenados a lo largo de la vida útil del sistema y responde de manera estable y rápida frente a las consultas más complejas y frecuentes.

Este trabajo generó documentación que puede servir como guía a desarrolladores para el uso de la base de datos durante la implementación del sistema, y a otros especialistas que diseñen bases de datos relacionales.



Recomendaciones

Para perfeccionar el resultado obtenido se recomienda:

- ❖ Considerar el uso de otros mecanismos para agilizar los procesos de búsquedas de información dentro de campos que contienen texto.
- ❖ Dinamizar la gestión de reportes en el Sistema de Captura y Catalogación de Medias utilizando el procesamiento analítico en línea (en inglés, On-Line Analytical Processing u OLAP) con bases de datos multidimensionales.



Bibliografía

1. Referencias Bibliográficas

- 📌 Barajas, Matilde Moreno. 2009. **EL SERVICIO DE CONSULTA DINÁMICA. GestioPolis.** [En línea] 23 de diciembre de 2010.
www.gestiopolis.com/administracion-estrategia/servicio-de-consulta-olap-bases-de-datos.htm...
- 📌 Castaño, Miguel. 1999. **Fundamentos y Modelos de Base de Datos.** España : RA-MA, 1999.
- 📌 Date, C. J. 2003. **Introducción a los Sistemas de Bases de Datos. Primera Parte.** La Habana : Felix Varela, 2003.
- 📌 Date, C. J. 2003. **Introducción a los Sistemas de Bases de Datos. Segunda Parte.** La Habana : Felix Varela, 2003.
- 📌 Date, C. J. 2003. 2003. **Introducción a los Sistemas de Bases de Datos. Tercera Parte.** La Habana : Felix Varela, 2003.
- 📌 Marquéz, Andrés. 2001. **Apuntes de Ficheros y Bases de Datos.** España : s.n., 2001.
- 📌 Mato, Rosa María. 2005. **Sistemas de Bases de Datos.** La Habana : Pueblo y Educación, 2005.
- 📌 Mora, Oscar Pérez. **Diseño de Base de Datos en Postgres.**
- 📌 Pressman, Roger S. 2005. **Ingeniería del Software, un enfoque práctico. Parte 1.** La Habana : Felix Varela, 2005.
- 📌 **Sitio Oficial de PostgreSQL.** [En línea] junio de 2011. PostgreSQL Global Development Group, 2011. <http://www.postgresql.org>.
- 📌 **Rational Unified Process. 2005 Vol.Version** [En línea] junio de 2011. PostgreSQL Global Development Group, 2011. <http://www.postgresql.org>.
- 📌 2005. **Rational Unified Process. 2005. Vol. Versión 7.0.1.**
- 📌 Padron, Yusniel Yanes. 2009. **Diseño de la Base de Datos para el Sistema de Captura y Catalogacion de Medias.** La Habana : s.n., 2009.
- 📌 **Un método para el diseño de la Base de Datos a partir del Modelo Orientado a Objetos.** González, Anaisa Hernández. 2004. 004, México : Red de Revistas Científicas de America Latina y el Caribe, España y Portugal, 2004, Vol. 7.



- 📍 2011. **Sitio Oficial de PostgreSQL.** [Online] PostgreSQL Global Development Group, 2011. <http://www.postgresql.org>.

2. Bibliografía Consultada

- 📍 Castaño, Miguel. 1999. **Fundamentos y Modelos de Base de Datos.** España : RA-MA, 1999.
- 📍 Date, C. J. 2003. **Introducción a los Sistemas de Bases de Datos. Primera Parte.** La Habana : Felix Varela, 2003.
- . 2003. **Introducción a los Sistemas de Bases de Datos. Segunda Parte.** La Habana : Felix Varela, 2003.
- . 2003. **Introducción a los Sistemas de Bases de Datos. Tercera Parte.** La Habana : Felix Varela, 2003.
- 📍 González, Maikel Hugo Álvarez. **DISEÑO DE LA BASE DE DATOS DEL SISTEMA DE INFORMACIÓN DE PERFORACIÓN DE POZOS.**
- 📍 Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. **El proceso Unificado del Desarrollo de Software.** Madrid : s.n., 2000.
- 📍 Marquéz, Andrés. 2001. **Apuntes de Ficheros y Bases de Datos.** España : s.n., 2001.
- 📍 Mato, Rosa María. 2005. **Sistemas de Bases de Datos.** La Habana : Pueblo y Educación, 2005.
- 📍 Meléndrez, Dra Edelsys Hernández. 2006. **Cómo escribir una tesis.** La Habana : Ciencias Médicas, 2006.
- 📍 Díez, Carmen Hernández. 2006. **Modelos Conceptuales de Datos.** 2006.
- 📍 Monge, Raúl. **Base de Datos Distribuidas: Replicación.**
- 📍 Mora, Oscar Pérez. **Diseño de Base de Datos en Postgres.**
- 📍 Pressman, Roger S. 2005. **Ingeniería del Software, un enfoque práctico. Parte 1.** La Habana : Felix Varela, 2005.
- . 2005. **Ingeniería del Software, un enfoque práctico. Parte 2.** La Habana : Felix Varela, 2005.
- 📍 Rosa, Karen Zerguera. 2009. PRIMICIA, Plataforma de Televisión **Informativa. Rol de Diseñador de Bases de Datos.** La Habana : s.n., 2009.



Anexo I: Especificación de Requisitos del Proyecto SCCM

La información que se expone a continuación, fue extraída del documento oficial de Especificación de Requisitos de Software v2.0, del proyecto Captura y Catalogación de Medias de la UCI. Dicho documento fue elaborado por los ingenieros Yoandri Quintana Rondón, Jean Michael Suárez Pérez y Zorilin Alonso Guerrero y está dirigido a los analistas del proyecto y a los posibles clientes del producto.

1. Requisitos Funcionales

1.1 Autenticar Usuario

Descripción: El sistema permitirá la autenticación de usuarios introduciendo el nombre de usuario y contraseña, accediendo de esta forma a las funcionalidades de la aplicación para las que se tienen permiso.

Entradas:

Usuario: Campo de texto que indica el nombre bajo el cual se registra una persona en el sistema. No se admite el campo vacío.

Contraseña: Campo de texto que indica la contraseña para entrar al sistema. No se admite el campo vacío.

Salidas: Se muestra la interfaz correspondiente de acuerdo a los permisos que posee el usuario autenticado.

1.2 Gestionar Rol

Insertar Rol

Descripción: El sistema debe permitir insertar nuevos roles con sus descripciones y permisos. Los roles definen el nivel de interacción de los usuarios con el sistema a partir de la asignación en el mismo de los permisos o las funcionalidades a las que se puede acceder.



Entradas:

Nombre: Campo de texto que define nombre descriptivo del rol. Este nombre debe relacionarse intuitivamente con las funcionalidades que puede realizar el usuario que posea el rol. No se admite el campo vacío.

Permisos: Acciones que podrá realizar el usuario que posea el rol dentro del sistema. Se debe seleccionar al menos un permiso.

Salidas: El sistema muestra el mensaje de notificación de la creación de rol. En caso de no introducirse datos válidos se notifica el error correspondiente.

Modificar Rol

Descripción: El sistema debe permitir modificar los datos de los roles insertados.

Entradas: Rol: Campo seleccionable para indicar el rol que se desea modificar (se asume como rol el nombre del mismo). Se debe seleccionar un rol.

Salidas: El sistema muestra el mensaje de notificación de la modificación del rol. En caso de no introducirse datos válidos se notifica el error correspondiente.

Eliminar Rol

Descripción: El sistema debe permitir eliminar los roles existentes.

Entradas: Rol: Campo seleccionable para indicar el rol que se desea eliminar (se asume como rol el nombre del mismo). Se debe seleccionar un rol.

Salidas: El sistema muestra el mensaje de notificación de la eliminación del rol.

1.3 Gestionar área de la institución

Insertar área

Descripción: El sistema debe permitir insertar las áreas en las que está estructurada la organización.

Entradas: Nombre: Campo de texto que define nombre del área.

Salidas: El sistema muestra el mensaje de notificación de la creación del área. En caso de no introducirse datos válidos se notifica el error correspondiente.



Modificar área

Descripción: El sistema debe permitir modificar los datos de áreas insertadas.

Entradas: Área: Campo seleccionable para indicar el área que se desea modificar. Se debe seleccionar un área.

Salidas: El sistema muestra el mensaje de notificación de la modificación del área. En caso de no introducirse datos válidos se notifica el error correspondiente.

Eliminar área

Descripción: El sistema debe permitir eliminar las áreas existentes.

Entradas: Área: Campo seleccionable para indicar el área que se desea eliminar. Se debe seleccionar un área.

Salidas: El sistema muestra el mensaje de notificación de la eliminación del área.

1.4 Gestionar Usuario

Buscar usuario en el servidor de directorio

Descripción: El sistema debe permitir localizar un usuario registrado en el servidor de directorio que se encuentra configurado en la aplicación.

Entradas:

Usuario: Campo de texto que especifica el usuario a buscar.

Nombre: Campo de texto que especifica el nombre del usuario a buscar.

Apellidos: Campo de texto que especifica el apellido del usuario a buscar.

Salida: Listado de usuarios que coinciden con los criterios de búsqueda especificados.

Crear Usuario

Descripción: El sistema debe permitir crear los usuarios de la aplicación.

Entradas: Se deben insertar los datos correspondientes para la creación de un nuevo usuario.



Usuario: Campo que especifica el usuario para el acceso a la aplicación, debe coincidir con un usuario existente en el servidor de directorio especificado. No admite el campo vacío.

Rol que desempeña: Campo seleccionable para indicar el rol del usuario en el sistema. Debe seleccionar un rol.

Identificación: Campo de texto que indica la identificación del usuario (Carné de Identidad, Identificación). No admite el campo vacío.

Habilitado: Campo de selección que indica si el usuario puede realizar acciones o no en el sistema.

Salidas: El sistema muestra un mensaje de notificación. En caso de no cumplirse algún requerimiento de los datos de entrada se notifica el error correspondiente.

Modificar Usuario

Descripción: El sistema debe permitir modificar los datos de usuarios que han sido insertados en la aplicación.

Entradas: Se debe seleccionar el usuario al cual se le realizarán las modificaciones. Se debe seleccionar un usuario.

Salidas: El sistema muestra un mensaje de notificación. En caso de no cumplirse algún requerimiento de los datos de entrada se notifica el error correspondiente.

Eliminar Usuario

Descripción: El sistema debe permitir eliminar usuarios existentes en la aplicación.

Entradas: Se debe seleccionar el usuario que se desea eliminar.

Salidas: El sistema muestra un mensaje de notificación. En caso de no cumplirse algún requerimiento de los datos de entrada se notifica el error correspondiente.

1.5 Realizar búsquedas de material audiovisual

Realizar búsqueda por clasificación



Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenadas en el servidor. Se permitirá establecer filtros referentes a la clasificación del material y a los atributos específicos de la clasificación.

Entradas:

Clasificación: Campo seleccionable para establecer la clasificación por la que se desea buscar materiales (Noticiero, Documental, Cine...). No admite campo vacío.

Atributos: Podrán insertarse los valores correspondientes a los atributos de la clasificación seleccionada. Puede especificarse ninguno, algunos o todos los valores de los atributos.

Salidas: El sistema muestra la lista de los materiales existentes en el servidor de acuerdo a las especificaciones realizadas.

Realizar búsqueda avanzada

Descripción: El sistema permitirá realizar búsquedas definiendo filtros avanzados relativos a la fecha o a la descripción del material, lo que posibilitará la búsqueda más específica de materiales atendiendo a las fechas almacenadas y a las descripciones asociadas a los mismos.

Entradas:

Búsqueda avanzada por fecha. Se pueden establecer los siguientes filtros de fecha:

- 📍 Filtro por intervalo de fechas: Busca los materiales cuya fecha esté comprendida en un rango definido.
- 📍 Fecha: Fecha sobre la que se quiere realizar la búsqueda.
- 📍 Fecha inicial: Fecha menor que se desea establecer en el rango de búsqueda.
- 📍 Fecha final: Fecha mayor que se desea establecer en el rango de búsqueda.

Filtro omitiendo algún parámetro de la fecha establecida (día, mes, año): Busca los materiales dada una fecha, permitiendo omitir algún atributo de la misma, ya sea el día, el mes o el año, o varios de ellos a la vez.



- 📍 Atributos a omitir en la búsqueda: Campo seleccionable para especificar lo(s) atributos(s) de la fecha que se desea omitir en la búsqueda: día, mes, año.

En el caso de los filtros avanzados de datos generales se podrán utilizar otros criterios, los cuales se realizarán tomando el campo de descripción de los materiales:

- 📍 Texto: Campo de texto para definir la frase que se desea buscar. En caso de no especificar texto no se toman en cuenta los criterios que se detallan debajo en caso de que se seleccionen.
- 📍 Ámbito: Campo seleccionable para definir la manera en que se va a buscar el texto:
- 📍 Con alguna de las palabras: los resultados deben incluir al menos uno de los términos especificados.
- 📍 Con todas las palabras: todos los términos que se incluyan deberán encontrarse en el resultado.
- 📍 Sin las palabras: los resultados no incluyen ninguno de los términos especificados en el texto.
- 📍 Con la frase exacta: deben aparecer todos los términos en el mismo orden en los resultados mostrados.
- 📍 Precisión: Campo seleccionable para definir el nivel de similitud que se desea en los resultados con respecto al texto que se ha introducido:
- 📍 Exacto: el resultado debe coincidir exactamente con el texto especificado, se tendrán en cuenta las mayúsculas y minúsculas.
- 📍 Contiene: en los resultados debe coincidir el texto especificado, pero no son tenidos en cuenta ni los acentos ni las letras mayúsculas.
- 📍 Subcadena: el término especificado debe ser parte de los términos de los resultados.

Salidas: El sistema muestra la lista de los materiales existentes en el servidor de acuerdo a las especificaciones realizadas.



Realizar búsqueda básica

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor. Se permitirá establecer filtros básicos en correspondencia con los datos que han sido asociados en la catalogación de materiales audiovisuales digitalizados.

Entradas: Se deben introducir los filtros que se requiera para la búsqueda, no necesariamente se tienen que especificar datos para todos los filtros.

Título: Campo de texto asociado al título del material que se desea buscar. Puede o no especificarse.

Palabras claves: Campo de texto en el que se deben introducir las palabras claves que se desea buscar, en caso de ser más de una separadas por comas. Los resultados a mostrar corresponderán con los materiales que contengan al menos una de las palabras claves especificadas.

Contenido: Campo seleccionable que responde a la naturaleza del material en la entidad: Recreativo, Informativo, Deportivo, entre otros que sean definidos.

Descripción: Campo de texto asociado a la descripción o tema del material que se desea buscar. Los resultados a mostrar corresponderán con los materiales que contengan dentro de su descripción el texto exacto introducido por el usuario.

Fecha inicial: Fecha menor que se desea establecer en el rango de búsqueda, corresponde con la fecha de inserción del material.

Fecha final: Fecha mayor que se desea establecer en el rango de búsqueda, corresponde con la fecha de inserción del material.

Salidas: El sistema muestra la lista de los materiales existentes en el servidor de acuerdo a las especificaciones realizadas.

Realizar búsqueda de materiales a catalogar

Descripción: El sistema permitirá la realización de búsqueda de videos que están almacenados en el servidor que no hayan sido catalogados.

Entradas:



Contenido: Campo seleccionable que responde a la naturaleza del material en la entidad: Recreativo, Informativo, Deportivo, entre otros que se definan.

Fecha inicial: Fecha menor que se desea establecer en el rango de búsqueda, corresponde con la fecha de inserción del material.

Fecha final: Fecha mayor que se desea establecer en el rango de búsqueda, corresponde con la fecha de inserción del material.

Salidas: El sistema muestra la lista de los materiales existentes en el servidor de acuerdo a las especificaciones realizadas y que no hayan sido catalogados.

1.6 Gestionar Solicitud

Insertar Solicitud

Descripción: El sistema debe permitir realizar la solicitud de material audiovisual indicando sus datos de identidad y los datos del material que desea.

Entradas:

Nombre y apellidos: Campo de texto que indica el nombre y los apellidos de la persona que desea realizar la solicitud. No admite campo vacío.

Identificación: Campo numérico para indicar la identificación de la persona que realiza la solicitud. No admite campo vacío.

Entidad perteneciente: Campo de texto para indicar la entidad a la que pertenece la persona que solicita el material. No admite campo vacío.

En el caso de los usuarios internos este campo representa el área de la institución

Autoriza: Campo de texto que indica la persona que autoriza la solicitud. No admite el campo vacío.

Materiales: Lista de materiales audiovisuales que se solicitan.

Perfil: Campo seleccionable que permite establecer el perfil de codificación para el o los materiales que se solicitan.



Salidas: El sistema muestra un mensaje de notificación que contiene el número de la solicitud realizada.

Modificar Solicitud

Descripción: El sistema debe permitir a los usuarios modificar los datos de solicitudes de material audiovisual que han realizado.

Entradas: Listado de las solicitudes realizadas por el usuario y que no han sido atendidas. Se debe seleccionar una solicitud.

Salidas: El sistema muestra un mensaje de notificación y el número que identifica la solicitud realizada.

Eliminar Solicitud

Descripción: El sistema debe permitir a los usuarios externos o internos del canal eliminar los datos de la solicitud de material audiovisual que ha realizado.

Entradas: Listado de las solicitudes realizadas por el usuario y que no han sido atendidas. Se debe seleccionar una solicitud.

Salidas: El sistema muestra un mensaje de notificación.

1.7 Atender Solicitud

Buscar Solicitud

Descripción: El sistema debe permitir acceder a los datos de las solicitudes realizadas por los usuarios para darle atención a las mismas.

Entradas:

Identificación: Campo para insertar la identificación de la persona que realizó la solicitud. No admite el campo vacío.

Número: Campo numérico para insertar el número de la solicitud y la fecha en que fue realizada. Puede o no especificarse.

Fecha Inicial: Campo para indicar la fecha inicial que se desea establecer en el rango de búsqueda de realización de la solicitud. No admite el campo vacío.



Fecha Final: Campo para indicar la fecha final que se desea establecer en el rango de búsqueda de realización de la solicitud. No admite el campo vacío.

Salidas: El sistema muestra el listado de las solicitudes que cumplen con los parámetros insertados.

Responder Solicitud

Descripción: El sistema debe permitir responder a las solicitudes que han sido realizadas por los usuarios.

Entrada: Se debe seleccionar una de las solicitudes que han sido mostradas en los resultados de la búsqueda.

Autoriza: Campo de texto que indica la persona que autoriza la solicitud. No admite campo vacío.

Salidas: El sistema muestra una interfaz donde el personal de atención al público podrá descargar los materiales solicitados en caso de estar listos para su descarga.

1.8 Catalogar Media

Insertar datos de catalogación

Descripción: El sistema debe permitir describir los videos digitales almacenados de acuerdo a la catalogación que se realice en la entidad.

Entradas: Se selecciona el material a catalogar a partir de alguna búsqueda realizada sobre los materiales existentes en el servidor.

En caso de seleccionarse un material ya catalogado se realizará la modificación de la ficha de catalogación existente editando los datos de la misma.

En caso de seleccionarse un material sin catalogar se insertarán los datos de catalogación.

Salida: El sistema almacena los datos de catalogación del video conformando el catálogo digital.

Crear Subclip



Descripción: El sistema debe permitir crear una secuencia virtual a partir del material que se está catalogando. Se debe haber establecido la marca de inicio y/o fin de la secuencia virtual que se quiere crear. En caso de sólo haberse establecido la marca de inicio en un tiempo del material, se asume como marca de fin el tiempo final del material original. En caso de sólo haberse establecido la marca de fin en un tiempo del material, se asume como marca de inicio el tiempo inicial del material original.

Entrada: Crear Secuencia: Función que permite crear una secuencia virtual a partir del material que se está reproduciendo.

Salida: Se crea la secuencia virtual correspondiente a la selección del usuario, será posible realizar sobre éste las mismas acciones de catalogación que sobre el resto de los materiales.

1.9 Emitir reportes de usuarios por rol.

Descripción: El sistema debe permitir emitir reportes sobre los datos de usuarios almacenados por rol.

Entradas: Rol: Campo de selección que permite especificar el rol de usuario por el que se desea filtrar. Debe seleccionarse un rol.

Salidas: El sistema muestra los resultados en un reporte de acuerdo al filtro establecido. El resultado se muestra agrupado por Nombre y Apellidos de los usuarios, el rol y el atributo habilitado.

1.10 Emitir reportes de actividad de usuario en el sistema.

Descripción: El sistema debe permitir emitir reportes sobre las acciones de interés realizadas por los usuarios del sistema.

1.11 Emitir reportes de catalogación por usuario.

Descripción: El sistema debe permitir emitir reportes sobre los materiales catalogados por un documentalista audiovisual.

Permitir especificar el documentalista

Entradas: Documentalista: Campo seleccionable que permite especificar el documentalista por el que se desea filtrar el resultado. Se puede seleccionar un solo documentalista. No admite el campo vacío.



Permitir especificar rango de fechas de la catalogación.

Fecha inicial: Campo de fecha que permite especificar la fecha de inicio del rango de búsqueda. No admite el campo vacío.

Fecha final: Campo de fecha que permite especificar la fecha de fin del rango de búsqueda. No admite el campo vacío.

Salidas: El sistema muestra los resultados en un reporte de acuerdo a los filtros establecidos. El resultado se muestra agrupado por Nombre y apellidos del documentalista, Fecha de catalogación, Identificador del material, Título de material y Clasificación del material.

1.12 Emitir reportes sobre los datos de los materiales no catalogados almacenados

Descripción: El sistema debe permitir emitir reportes sobre los datos de los materiales almacenados que no han sido catalogados.

Permitir filtrar por rango de fechas en el que el material fue insertado.

Entradas:

Contenido: Campo de selección que permite especificar a qué categoría pertenece el material dentro de la entidad. Se puede o no seleccionar.

Fecha inicial: Campo de fecha que permite especificar la fecha de inserción de inicio del rango de búsqueda. No admite el campo vacío.

Fecha final: Campo de fecha que permite especificar la fecha de inserción fin del rango de búsqueda. No admite el campo vacío.

Salidas: El sistema muestra los resultados en un reporte de acuerdo a los filtros establecidos. El resultado se muestra agrupado por ID del material, Código del casete en el que se encuentra el material original, Contenido, Fecha de inserción en el sistema.

1.13 Emitir reportes sobre los datos de los materiales catalogados almacenados

Descripción: El sistema debe permitir emitir reportes sobre los datos de los materiales almacenados que han sido catalogados.



Permitir especificar rango de fechas en el que el material fue catalogado.

Entradas:

Contenido: Campo de selección permite especificar a qué tipo de contenido pertenece el material dentro de la entidad. Se puede o no seleccionar.

Fecha inicial de inserción: Campo de fecha que permite especificar la fecha de inserción de inicio del rango de búsqueda. No admite el campo vacío.

Fecha final de inserción: Campo de fecha que permite especificar la fecha de inserción fin del rango de búsqueda. No admite el campo vacío.

Clasificación: Campo de selección que permite especificar la clasificación de materiales por la que se desea filtrar. Puede seleccionar ninguna, una o más de una clasificación.

Fecha inicial de catalogación: Campo de fecha que permite especificar la fecha de catalogación de inicio del rango de búsqueda. No admite el campo vacío.

Fecha final de catalogación: Campo de fecha que permite especificar la fecha de catalogación fin del rango de búsqueda. No admite el campo vacío.

Salidas: El sistema muestra los resultados en un reporte de acuerdo a los filtros establecidos. El resultado se muestra agrupado por Código de casete en el que se encuentra el material original, Contenido, Fecha de inserción en el sistema, Título, Clasificación, Fecha de catalogado.

1.14 Emitir reportes por datos de solicitudes realizadas en el sistema

Descripción: El sistema debe permitir emitir reportes sobre los datos de las solicitudes de material realizadas en el sistema.

Entradas:

Naturaleza de la Solicitud: Campo de selección que permite especificar si se desea filtrar por solicitudes internas, externas o ambas. Se puede o no especificar.



Resuelta: Campo de selección para indicar si la solicitud se quiere filtrar por solicitudes resueltas, no resueltas o ambas. Se puede o no especificar.

Fecha inicial: Campo de fecha que permite especificar la fecha de inicio del rango de búsqueda. No admite el campo vacío.

Fecha final: Campo de fecha que permite especificar la fecha de fin del rango de búsqueda. No admite el campo vacío.

Salidas: El sistema muestra los resultados en un reporte de acuerdo a los filtros establecidos. El resultado se muestra agrupado por Identificador de la solicitud, Nombre del usuario solicitante, Fecha de la solicitud, Naturaleza de la solicitud (interna o externa), Estado de la solicitud (resuelta o no), Nombre de quien autorizó la solicitud, Lista de los materiales solicitados.

2. Requisitos No Funcionales

2.1 Usabilidad

La aplicación debe ser concebida para ser utilizada por personas que tengan conocimientos básicos de informática y en el trabajo de gestión y catalogación de audiovisuales.

Tiempo de adiestramiento

Para que los usuarios puedan utilizar óptimamente el sistema, necesitarán un tiempo de adiestramiento y estudio de los conceptos principales asociados al dominio de la aplicación.

2.2 Fiabilidad

Disponibilidad: el sistema debe estar disponible todas las horas del día.

Tiempo medio de reparación: el sistema no debe estar fuera de servicio más de 2 horas a partir de un fallo.

2.3 Eficiencia

El sistema debe ser eficiente a las peticiones realizadas en cada momento, el flujo de trabajo que sigue la aplicación no permite el fallo de ninguna de las partes pues éste influye de manera drástica sobre el próximo paso.



Tiempo de respuesta

El tiempo de respuesta promedio de las transacciones u operaciones realizadas en el sistema no debe exceder un tiempo máximo de 5 segundos. Las opciones de reproducción de video (adelantar, retrasar, ir a un tiempo específico, entre otras) deben tener un tiempo de respuesta aceptable para la realización eficiente de la catalogación audiovisual.

2.4 Software

Sistema Gestor de Base de Datos: *PostgreSQL* 8.4 (*server* BD) Licencia Código Abierto. TPL. y *pgAdmin3*, Licencia Código Abierto. TPL.

2.5 Hardware

Servidor de Base de Datos: Arquitectura 64 bits (x64), Procesador 2 *QuadCore*, Memoria de 16 GB.

2.6 Soporte

El sistema permitirá modificar o añadirle módulos cuando sea necesario, asegurando su extensibilidad y mejores prestaciones. Las opciones de instalación y mantenimiento del sistema deben ser fáciles, además debe ser adaptable a cualquier plataforma y los instaladores deben incluir todas las librerías que necesite el sistema para funcionar.

2.7 Seguridad

Se debe garantizar la integridad, disponibilidad y confidencialidad de la información manipulada por el sistema. Es necesario proteger el sistema contra accesos no autorizados, impidiendo el acceso a la información y los recursos por usuarios que no tengan permisos. Para ello se deberán establecer mecanismos para llevar constancia de las acciones realizadas por los usuarios en el sistema que sean de interés controlar por la entidad.

Se deberá realizar la petición de confirmación ante acciones irreversibles en el sistema, dígase eliminación de cualquier información.



Anexo II: Descripción de las tablas resultantes en las relaciones del Modelo Entidad-Relación

Nombre de la Tabla: DRol_DPermiso		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos entre las tablas DRol y DPermisos.		
Atributo	Tipo de dato	Descripción
id Permiso_Pk	int4	Identificador del Permiso.
id Rol_Pk	int4	Identificador del Rol.

Tabla 54. Descripción de la tabla: DRol_DPermiso, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DSolicitud_DFicha		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos entre las tablas DSolicitud y DFicha		
Atributo	Tipo de dato	Descripción
idSolicitud_Pk	int4	Identificador de la solicitud.
idFicha_Pk	int4	Identificador de la ficha.

Tabla 55. Descripción de la tabla: DSolicitud_DFicha, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DSolicitud_DCliente		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos entre las tablas DSolicitud y DCliente		
Atributo	Tipo de dato	Descripción
idSolicitud_Pk	int4	Identificador de la solicitud.
idCliente_Pk	int4	Identificador del cliente.

Tabla 56. Descripción de la tabla: DSolicitud_DCliente, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DSolicitud_DUusuario		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos entre las tablas DSolicitud y DUusuario		
Atributo	Tipo de dato	Descripción
idSolicitud_Pk	int4	Identificador de la solicitud.



idUsuario_Pk	int4	Identificador del usuario que realiza una solicitud.
--------------	------	--

Tabla 57. Descripción de la tabla: DSolicitud_DUusuario, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DFicha_DFicha		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos de la tabla ficha consigo misma que representa una jerarquía en las fichas almacenadas.		
Atributo	Tipo de dato	Descripción
idFichaPadre_Pk	int4	Ficha padre.
idFichaHija_Pk	int4	Ficha hija.

Tabla 58. Descripción de la tabla: DFicha_DFicha, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DTipologia_DTipologia		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos de la tabla DTipologia consigo misma que representa una jerarquía en las Tipologías almacenadas.		
Atributo	Tipo de dato	Descripción
idTipologiaPadre_Pk	int4	Tipología padre.
idTipologiaHija_Pk	int4	Tipología Hija.

Tabla 59. Descripción de la tabla: DTipologia_DTipologia, del Modelo lógico de datos. Anexos.

Nombre de la Tabla: DFicha_DPalabraClave		
Descripción de la Tabla: Se crea a partir de la relación muchos a muchos entre las tablas DFicha y DPalabraClave		
Atributo	Tipo de dato	Descripción
idFicha_Pk	int4	Identificador de la ficha.
idPalabraClave_Pk	int4	Identificador de la palabra clave.

Tabla 60. Descripción de la tabla: DFicha_DPalabraClave, del Modelo lógico de datos. Anexos.



Anexo III: Otras Consultas a la Base de Datos

🔗 Saber los permisos de un determinado usuario del sistema.

SELECT

```
public.dpermiso.nombrepermiso_nn AS permiso,
public.ntipopermiso.nombretipopermiso_nn AS tipo_permiso
```

FROM

```
public.dpermiso,
public.drol,
public.drol_dpermiso,
public.dusuario,
public.ntipopermiso
```

WHERE

```
public.dpermiso.idtipopermiso_fk =
public.ntipopermiso.idtipopermiso_pk AND
public.drol_dpermiso.idpermiso_pk = public.dpermiso.idpermiso_pk AND
public.drol.idrol_pk = public.drol_dpermiso.idrol_pk AND
public.drol.idrol_pk = public.dusuario.idrol_fk AND
public.dusuario.usuario_pk = 'Thomson';
```

🔗 Conocer las fallas de un proyecto determinado.

SELECT

```
public.nfalla.nombrefalla_nn AS falla,
public.ntipofalla.nombretipofalla_nn AS tipo,
public.nfalla_dproyecto.tiempoinicio_nn AS tiempoinicio,
public.nfalla_dproyecto.tiempofin_nn AS tiempofin,
public.nduracion.nombreduracion_nn AS duracion_s,
public.nintensidad.nombreintensidad_nn AS intensidad_s,
public.nfalla_dproyecto.fecha_nn AS fecha
```



FROM

```
public.nfalla,  
public.ntipofalla,  
public.nfalla_dproyecto,  
public.nduracion,  
public.nintensidad
```

WHERE

```
public.ntipofalla.idtipofalla = public.nfalla.idtipofalla AND  
public.nfalla_dproyecto.idfalla = public.nfalla.idfalla AND  
public.nintensidad.intensidad = public.nfalla_dproyecto.intensidad AND  
public.nfalla_dproyecto.duracion = public.nduracion.duracion AND  
public.nfalla_dproyecto.idproyecto_pk = '471';
```

 **Conocer las fichas catalogadas por un usuario.****SELECT**

```
public.dficha.idficha_fk AS ficha
```

FROM

```
public.dficha,  
public.dplanificacion,  
public.dmedia
```

WHERE

```
public.dplanificacion.idusuario_fk LIKE 'Thomson' AND  
public.dplanificacion.idtarea_fk = 4 AND  
public.dplanificacion.porciento_nn = 100 AND  
public.dmedia.idmedia_pk = public.dplanificacion.idtrabajo_nn AND  
public.dmedia.idmedia_pk = public.dficha.idmedia_fk AND  
public.dficha.terminada_nn;
```



🔗 **Conocer qué archivos medias catalogó un usuario determinado.**

SELECT

```
public.dficha.idficha_fk AS idFicha
```

FROM

```
public.dficha,  
public.dplanificacion,  
public.dmedia
```

WHERE

```
public.dplanificacion.idusuario_fk LIKE '%AVRDWTZKGIZ%' AND  
public.dplanificacion.idtarea_fk = 4 AND  
public.dplanificacion.porciento_nn = 100 AND  
public.dmedia.idmedia_pk = public.dplanificacion.idtrabajo_nn AND  
public.dmedia.idmedia_pk = public.dficha.idmedia_fk AND  
public.dficha.terminada_nn
```

🔗 **Conocer el nombre, los apellidos y el cargo de una persona que autorizó una solicitud y la fecha en que ocurrió.**

SELECT

```
public.npersonaautoriza.nombre_nn AS nombre,  
public.npersonaautoriza.apellidos_nn AS apellidos,  
public.npersonaautoriza.cargo_nn AS cargo,  
public.npersonaautoriza_dsolicitud.fecha_nn AS fecha
```

FROM

```
public.npersonaautoriza_dsolicitud JOIN public.npersonaautoriza  
USING (idpersonaautoriza_pk)
```

WHERE

```
public.npersonaautoriza_dsolicitud.idsolicitud_pk = 2;
```



 **Conocer los permisos que tiene un rol determinado en el sistema.**

SELECT


```
public.dpermiso.nombrepermiso_nn AS permiso,  
public.ntipopermiso.nombretipopermiso_nn AS tipo
```

FROM

```
public.drol,  
public.drol_dpermiso,  
public.dpermiso,  
public.ntipopermiso
```

WHERE

```
public.drol.nombrerol_nn = 'pvjhnovrog' AND  
public.dpermiso.idpermiso_pk = public.drol_dpermiso.idpermiso_pk AND  
public.drol.idrol_pk = public.drol_dpermiso.idrol_pk AND  
public.ntipopermiso.idtipopermiso_pk =  
public.dpermiso.idtipopermiso_fk;
```

 **Conocer las medias que están sin catalogar.**

SELECT

```
public.dmedia.idmedia_pk AS media
```

FROM

```
public.dmedia
```

WHERE

```
public.dmedia.catalogada_nn = 'false';
```



Glosario de Términos

Atributo: Es la unidad menor de información sobre un objeto almacenado en la BD. Son propiedades que poseen las entidades, por ejemplo: el color, el nombre, identificador, o cualquier otro elemento que describa a la entidad que se necesite conocer. Se representan a través de elipses, o pequeños círculos, conectadas a la entidad mediante una línea recta.

Base de Datos (BD): Conjunto de datos con carácter persistente almacenados para su posterior uso en un sistema informático. Estos datos pueden estar localizados tanto en la misma máquina del sistema que los necesita (máquina cliente) como en una máquina remota (servidor de base de datos) conectados mediante una red local.

Base de Datos Relacional: Base de datos cuya estructura está constituida por tablas bidimensionales estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común. Los campos de las tablas tienen las mismas características: nombre, tipo de dato, longitud. En cada tabla existe un campo (o conjunto de éstos) que identifica a cada registro como único, este campo es llamado identificador o llave.

CASE: (*Computer-Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador). Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Consistencia: Una transacción transforma un estado consistente de la base de datos en otro igual sin necesidad de conservar la consistencia en todos los puntos intermedios.

Clase: Define los atributos y los métodos de un conjunto de objetos.

Cubo OLAP (Procesamiento analítico en línea): Se utiliza cuando nos referimos a las bases de datos multidimensionales, en las cuales el almacenamiento físico de los datos se realiza en vectores multidimensionales. Se pueden considerar como una ampliación de las dos dimensiones de una hoja de cálculo. Su objetivo principal es agilizar las consultas de grandes cantidades de datos.



DDL: Un lenguaje de definición de datos (*Data Definition Language*, DDL por sus siglas en inglés), es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

Escalabilidad: Es la capacidad que posee el sistema informático para adaptarse a nuevos cambios. Es su habilidad de adaptar su tamaño o configuración para ajustarse a las circunstancias cambiantes.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

Herramienta: Subprograma o módulo encargado de funciones específicas y afines entre sí para realizar una tarea. Una aplicación o programa puede contar con múltiples herramientas a su disposición. Por ejemplo, el corrector ortográfico puede ser una herramienta en una aplicación para redactar documentos, pero no es una aplicación en sí misma.

Licencia BSD: Pertenece al grupo de licencias de software Libre. Utilizada por primera vez en 1980 por *Berkeley Source Distribution* (BSD). Esta licencia permite el uso de código BSD sobre software no libre aunque éste luego pase a ser liberado.

Media: Contenido digital, entiéndase audio o video, para informar o entretener al usuario.

Metodología: Es un conjunto de procedimientos, técnicas, instrumentos y documentos que ayudan a los analistas y programadores en sus esfuerzos para obtener un nuevo sistema informativo.

Modelo: Representación abstracta de la realidad.

Multiplataforma: Término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en sistemas operativos o plataformas.



Normalización: La normalización es un proceso que permite obtener relaciones de la forma plana bidimensional, logrando que las tablas de datos tengan la forma adecuada, para evitar tanto la ocurrencia de anomalías de actualización como la falta de consistencia de los datos.

Proceso: Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados, utilizando recursos.

Producto: Resultado de cada etapa. Proyecto: Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos.

Release: Término que se utiliza para nombrar un entregable de producto que esta listo para salir al mercado.

SCCM: Sistema de Captura y Catalogación de Medias.

Seguridad: Seguridad es la cualidad de seguro, es decir, de estar libre y exentos de todo daño, peligro o riesgo.

SGBD: Sistema Gestor de Base de Datos.

Software: Equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Software libre: Software que puede ser distribuido, modificado, redistribuido, copiado y usado libremente. Que un software sea libre no quiere decir que sea gratuito, error que viene de la traducción Free Software, pues free en inglés significa lo mismo libre que gratuito.

SQL: (*Structured Query Language*). Es un lenguaje de acceso a las Bases de Datos, permite especificar todas las operaciones sobre la base de datos como por ejemplo: Inserción, Borrado, Actualización. Utiliza características de álgebra y cálculo relacional permitiendo de esta forma realizar consultas a la base de datos de forma sencilla.



Glosario de Términos

Tecnología: Abarca un conjunto de técnicas, conocimientos y procesos para el diseño y construcción de objetos para satisfacer necesidades humanas. La tecnología puede referirse a objetos que usa la humanidad (como máquinas, utensilios, hardware), pero también abarca sistemas, métodos de organización y técnicas, se basa en aportes científicos.

Tupla: Una ocurrencia de artículo o tupla consiste en un grupo de ocurrencias de campos relacionados, representando una acción entre ellos.

UCI: Universidad de las Ciencias Informáticas

UML: (*Unified Modeling Language*, Lenguaje Unificado de Modelado). Notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

