



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas
Facultad 9

Plataforma VideoWeb. Rol Diseñador de Base de Datos.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor:

Yordany Fernández Padrón.

Tutor:

Ing. Yunier Albrecht Delgado.

Asesora:

Lic. Anisleidi Infante Rodríguez.

Ciudad de La Habana, mayo de 2009.

“Año del 50 aniversario del triunfo de la Revolución”



*Si quieres ser sabio, aprende a interrogar razonablemente, a escuchar con atención,
a responder serenamente y a callar cuando no tengas nada que decir.*

Johann Kaspar Lavater.



Declaración de Autoría.

Declaro que soy el único autor del trabajo titulado: "Plataforma VideoWeb. Rol Diseñador de Base de Datos" y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 27 días del mes de mayo del año 2009.

Yordany Fernández Padrón.

Autor

Ing. Yunier Albrecht Delgado.

Tutor



Opinión del Tutor.



Datos de Contacto.

Tutor: Ing. Yunier Albrecht Delgado.

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2007.
- Jefe del Polo de Video y Sonido Digital, Facultad 9, Universidad de las Ciencias Informáticas.
- Profesor del Departamento de Técnicas de Programación, Facultad 9, Universidad de las Ciencias Informáticas.

Correo electrónico: yalbrecht@uci.cu

Asesora: Lic. Anisleidis Infante Rodríguez.

- Licenciada en Lengua Inglesa con segunda Lengua Extranjera, Universidad de Oriente, 2007.
- Profesora del Departamento de Humanidades, Facultad 9, Universidad de las Ciencias Informáticas.

Correo electrónico: anisleidisi@uci.cu



Opiniones y Avales.





Agradecimientos.

A mis padres Ricardo y Nereida por haber confiado siempre en mí, por luchar conmigo todos estos años, por brindarme su apoyo incondicional en todo momento y hacer posible este sueño.

A Ramón y a Concepción por ayudarme y apoyarme durante estos 5 años, por ser mis segundos padres y tratarme como tal.

A toda mi familia por su apoyo y confianza, por estar siempre pendientes de mí.

A mis incondicionales amigos María, Yirobis y Lianet por estar siempre presentes, en los buenos y malos momentos.

A Jenny por transmitirme tanta seguridad, confianza y apoyo.

A mis compañeros de lucha de todos estos años Angel, Renier y Frank, por haber sido más que amigos.

A Yunier Albrecht Delgado por ser además de tutor amigo, por confiar en mí en todo momento y transmitirme tanta confianza.

A Carmen Leisa por auxiliarme tanto y estar ahí siempre dispuesta a ayudarme.

A todos los miembros del proyecto Plataforma VideoWeb por ser como una familia. A la antigua tropa de UCITeVe por compartir juntos tanto tiempo.

A todos mis compañeros de grupo, especialmente a los que se han mantenido conmigo durante estos cinco largos años.

A todos los profesores que contribuyeron a mi formación como profesional.

A la Revolución y su creador Fidel Castro Ruz por hacer posible que formase parte de este gran proyecto.



Dedicatoria.

Dedico este trabajo de diploma a todas las personas que de una forma u otra han aportado su granito de arena; brindándome las enseñanzas, el valor y la confianza que necesitaba para formarme como profesional. Especialmente a mis padres por ser mi razón de ser y sin los cuales no hubiera llegado hasta aquí, a mi familia por estar siempre a mi lado brindándome apoyo, a la memoria de los que ya no están, especialmente a mi abuela Lourdes por brindarme siempre su cariño.

A todos ustedes...



Resumen.

El presente trabajo consiste en la realización de un informe de investigación práctico del desarrollo del rol Diseñador de Base de Datos dentro del proyecto Plataforma VideoWeb del polo productivo Video y Sonido Digital de la Facultad 9 de la Universidad de las Ciencias Informáticas (UCI). Dicho proyecto tiene como objetivo desarrollar una plataforma web que permita la publicación y administración de archivos multimedia a través de la red de datos, basándose principalmente en la tecnología de streaming y en la distribución de aplicaciones cliente/servidor, empleando para el desarrollo de este producto solamente herramientas de software libre, que posibilite su uso y comercialización, sin restricciones por concepto de licencias de software propietario.

En el presente trabajo se realiza el diseño de la estructura de almacenamiento de datos persistentes que se utiliza en la Plataforma VideoWeb. Se define el diseño detallado de la base de datos, incluyendo tablas, índices, vistas, restricciones, desencadenantes, procedimientos almacenados y otras construcciones específicas de la base de datos necesarios para almacenar, recuperar y suprimir objetos persistentes almacenados, garantizando que los datos persistentes se almacenen de forma coherente y eficaz. Se determina el comportamiento que se debe implementar en la base de datos. Se analizan las principales clasificaciones, modelos, tendencias, gestores y herramientas de base de datos en el ámbito internacional, nacional y en la Universidad de las Ciencias Informáticas. Se desarrollan los artefactos correspondientes al rol Diseñador de Base de Datos y se realiza un análisis del trabajo realizado elaborando un resumen de buenas prácticas en el desempeño de este rol. Se efectúa la descripción y valoración de la solución propuesta para la realización del trabajo, así como las conclusiones y recomendaciones del mismo.

Palabras Claves.

Informática; VideoWeb; Material Audiovisual; Medias; Sistemas de Gestión de Base de Datos; Bases de Datos; Control de Acceso; Transferencia.



Abstract.

This work consists on the realization of a practical research report of the development of the Database Designer role within the VideoWeb platform project belonging to the Digital Video and Sound productive pole of the Faculty 9 of the University of Informatics Sciences (UCI). This project aims to develop a web platform that enables the publication and administration of multimedia files across the data network, based on streaming technology and the distribution of client / server applications, using for the development of this product only free software tools, enabling its use and marketing without restriction for proprietary software licenses.

In the present work, the design of the structure for the storage of persistent data that will be used in the VideoWeb Platform is made. The detailed design of the database is also made, including tables, indexes, views, constraints, triggers, stored procedures and other specific construction of the database required to store, retrieve and delete persistent objects stored, ensuring that persistent data is stored in a coherent and effective way. The behavior that must be implemented in a database is defined. The main classifications, models, tendencies, database managers and other database tools are analyzed in the international and national scope, as well as at the University of Informatics Sciences (UCI). The artifacts corresponding to the Database Designer role are developed and an analysis of the done work is made, making a summary of good practices in the performance of this role. Finally the description and valuation of the proposed solution for the realization of the work is made, as well as the conclusions and recommendations of it.

Keywords.

Informatics; VideoWeb; Audiovisual Materials; Medias; Management Systems Database; Databases; Access Control; Transfer.



Índice de Contenidos.

Introducción.	1
Capítulo 1: Fundamentación Teórica	5
1.1. Introducción.....	5
1.2. Plataforma Web.	5
1.3. Transmisión de vídeo y audio.	6
1.4. Plataforma VideoWeb.	6
1.5. Introducción a las Bases de Datos.....	6
1.5.1. Breve historia de los sistemas de almacenamiento de datos.	7
1.6. ¿Qué es una Base de Datos?	8
1.7. Sistema Gestor de Base de Datos (SGBD).....	10
1.8. Clasificaciones de las Bases de Datos.	11
1.8.1. Según la forma en que cambia la información almacenada.	11
1.8.1.1. Bases de datos estáticas.	11
1.8.1.2. Bases de datos dinámicas.	11
1.8.2. Según la información que almacenan.	12
1.8.2.1. Bases de datos bibliográficas.	12
1.8.2.2. Bases de datos de texto completo.	12
1.8.2.3. Directorios.	12
1.8.2.4. Banco.	12
1.9. Modelos de Bases de Datos.....	12
1.9.1. Bases de Datos Jerárquicas.....	13
1.9.2. Bases de Datos de Red.	14
1.9.3. Bases de Datos Relacional.	14
1.9.4. Bases de Datos Multidimensionales.	15
1.9.5. Bases de Datos Orientadas a Objetos.	16
1.9.6. Bases de Datos Objeto Relacionales.....	18
1.9.7. Bases de Datos Documentales.	19
1.9.8. Bases de Datos Deductivas.	19
1.9.9. Bases de Datos Distribuidas.....	19



1.10.	Fases del Diseño de Base de Datos.-----	19
1.10.1.	Diseño Conceptual.-----	19
1.10.2.	Diseño Lógico.-----	20
1.10.3.	Diseño Físico.-----	20
1.11.	Principales Tendencias de Bases de Datos.-----	20
1.12.	Principales Gestores de Bases de Datos.-----	21
1.12.1.	Sistema Gestor de Bases de Datos Oracle.-----	22
1.12.2.	Sistema Gestor de Bases de Datos Microsoft SQL Server 2000.-----	23
1.12.3.	Sistema Gestor de Bases de Datos MySQL.-----	25
1.12.4.	Sistema Gestor de Bases de Datos PostgreSQL.-----	26
1.13.	Herramientas CASE (Computer Aided Software Engineering).-----	28
1.13.1.	ERwin.-----	28
1.13.2.	EasyCASE.-----	29
1.13.3.	Oracle Designer.-----	29
1.13.4.	Case Studio.-----	29
1.13.5.	Visual Paradigm.-----	30
1.14.	Conclusiones.-----	30
Capítulo 2:	Descripción y análisis de la solución propuesta-----	32
2.1.	Introducción.-----	32
2.2.	Estrategia de integración de la solución con otros módulos o sistemas.-----	32
2.3.	Descripción de la arquitectura y fundamentación.-----	33
2.3.1.	Modelo Vista Controlador (MVC).-----	33
2.3.2.	Sistema Gestor de Base de Datos (SGBD).-----	34
2.3.3.	Modelo de Bases de Datos.-----	35
2.3.4.	Herramienta CASE (Computer Aided Software Engineering).-----	36
2.3.5.	Herramienta de administración y desarrollo para PostgreSQL.-----	37
2.4.	Análisis de optimización de consultas.-----	37
2.5.	Requisitos funcionales y no funcionales del sistema propuesto.-----	39
2.5.1.	Requisitos funcionales (RF).-----	40
2.5.2.	Requisitos no funcionales (RNF).-----	45



2.6.	Diagrama de clases persistentes.	49
2.6.1.	Descripción de las clases.	51
2.7.	Diseño de la Base de Datos.	60
2.7.1.	Diagrama Entidad Relación de la Base de Datos.	60
2.7.2.	Descripción de las Tablas.	62
2.8.	Conclusiones.	74
Capítulo 3:	Validación del diseño realizado	76
3.1.	Introducción.	76
3.2.	Validación teórica del diseño.	76
3.2.1.	Integridad.	76
3.2.2.	Normalización de la base de datos.	79
3.2.2.1.	Primera forma normal (1FN).	80
3.2.2.2.	Segunda forma normal (2FN).	81
3.2.2.3.	Tercera forma normal (3FN).	81
3.2.3.	Análisis de redundancia de información.	82
3.2.4.	Análisis de la seguridad de la base de datos.	82
3.2.5.	Trazabilidad de las acciones.	84
3.3.	Validación funcional.	85
3.3.1.	Llenado voluminoso e inteligente de la base de datos.	85
3.3.2.	Pruebas de carga intensiva.	86
3.4.	Conclusiones.	89
Conclusiones.		90
Recomendaciones.		92
Referencias Bibliográficas.		93
Bibliografía.		96
Glosario de Términos.		100



Índice de Figuras.

Figura 1: Estructura de una Base de Datos Jerárquica.-----	13
Figura 2: Estructura de una Base de Datos de Red. -----	14
Figura 3: Estructura de una Base de Datos de Relacional.-----	15
Figura 4: Representación espacial de una variable multidimensional.-----	16
Figura 5: Proceso de relación entre los componentes en una Arquitectura MVC. -----	34
Figura 6: Diagrama de ejecución de una consulta en PostgreSQL. -----	38
Figura 7: Diagrama de Clases Persistentes de la Plataforma VideoWeb. -----	51
Figura 8: Diagrama Entidad Relación de la Plataforma VideoWeb. -----	61



Índice de Tablas.

Tabla 1: Recursos de hardware para servidor de almacenamiento. -----	46
Tabla 2: Recursos de hardware para servidor de aplicaciones. -----	46
Tabla 3: Descripción de la clase Interfaz. -----	52
Tabla 4: Descripción de la clase Usuario.-----	52
Tabla 5: Descripción de la clase Rol. -----	53
Tabla 6: Descripción de la clase Permisos. -----	53
Tabla 7: Descripción de la clase Node.-----	53
Tabla 8: Descripción de la clase Categoria. -----	54
Tabla 9: Descripción de la clase Seccion. -----	54
Tabla 10: Descripción de la clase Artículo_Contenido. -----	54
Tabla 11: Descripción de la clase Publicacion_Archivo_Multimedia.-----	55
Tabla 12: Descripción de la clase Lista_de_Reproduccion. -----	55
Tabla 13: Descripción de la clase Archivo_Multimedia. -----	55
Tabla 14: Descripción de la clase Servidores_FTP.-----	56
Tabla 15: Descripción de la clase Tipologia_Archivo_Multimedia.-----	56
Tabla 16: Descripción de la clase Imagen. -----	56
Tabla 17: Descripción de la clase Pelicula.-----	57
Tabla 18: Descripción de la clase Serie. -----	57
Tabla 19: Descripción de la clase Temporada. -----	57
Tabla 20: Descripción de la clase Capitulo. -----	57
Tabla 21: Descripción de la clase VideoMusical.-----	58
Tabla 22: Descripción de la clase Documental. -----	58
Tabla 23: Descripción de la clase Musica.-----	58
Tabla 24: Descripción de la clase AudioVisualRealizado. -----	59
Tabla 25: Descripción de la clase AudioVisualVariado. -----	59
Tabla 26: Descripción de la clase Teleclase. -----	59
Tabla 27: Descripción de la clase Materia. -----	59
Tabla 28: Descripción de la tabla Interfaz. -----	62
Tabla 29: Descripción de la tabla Usuario. -----	63



Tabla 30: Descripción de la tabla Rol_Usuario.	63
Tabla 31: Descripción de la tabla Rol.	63
Tabla 32: Descripción de la tabla Rol_Perminos.	64
Tabla 33: Descripción de la tabla Perminos.	64
Tabla 34: Descripción de la tabla Node.	65
Tabla 35: Descripción de la tabla Categoria.	65
Tabla 36: Descripción de la tabla Categoria_Seccion.	65
Tabla 37: Descripción de la tabla Seccion.	66
Tabla 38: Descripción de la tabla Articulo_Contenido.	66
Tabla 39: Descripción de la tabla Publicacion_Archivo_Multimedia.	67
Tabla 40: Descripción de la tabla Publicacion_Archivo_Multimedia_Lista_de_Reproduccion.	67
Tabla 41: Descripción de la tabla Lista_de_Reproduccion.	67
Tabla 42: Descripción de la tabla Archivo_Multimedia.	68
Tabla 43: Descripción de la tabla Servidores_FTP.	68
Tabla 44: Descripción de la tabla Tipologia_Archivo_Multimedia.	69
Tabla 45: Descripción de la tabla Imagen_Tipologia_Archivo_Multimedia.	69
Tabla 46: Descripción de la tabla Imagen.	69
Tabla 47: Descripción de la tabla Pelicula.	70
Tabla 48: Descripción de la tabla Serie.	71
Tabla 49: Descripción de la tabla Temporada.	71
Tabla 50: Descripción de la tabla Capitulo.	71
Tabla 51: Descripción de la tabla VideoMusical.	72
Tabla 52: Descripción de la tabla Documental.	72
Tabla 53: Descripción de la tabla Musica.	73
Tabla 54: Descripción de la tabla AudioVisualRealizado.	73
Tabla 55: Descripción de la tabla AudioVisualVariado.	74
Tabla 56: Descripción de la tabla Teleclase.	74
Tabla 57: Descripción de la tabla Materia.	74



Introducción.

La informática se ha convertido en aliada indispensable del trabajo cotidiano, donde junto a otras disciplinas tecnológicas avanza a gran velocidad. Cada año se incrementa vertiginosamente la velocidad de los procesadores y la capacidad de los dispositivos de almacenamiento. De los primeros ordenadores que solo aceptaban entradas numéricas y textuales, a los actuales que poseen una interfaz ergonómica que facilita la interacción entre el usuario y el ordenador, el avance es considerable. Las computadoras se han convertido en una herramienta indispensable para la realización de tareas y la solución de problemas, tanto así, que sin ellas en la actualidad muchas de las actividades colapsarían.

La digitalización y la introducción de la fibra óptica en las redes de computadoras –ampliamente difundidas en los últimos años– han incrementado la capacidad y la calidad de las redes. Este paso en las tecnologías de la comunicación propició un acelerado aumento del ancho de banda, la integración de las informaciones de datos, audio y video. Además favoreció el despliegue de las aplicaciones de proceso distribuido cliente/servidor, multimedia y video interactivo.

El aumento acelerado del uso de las computadoras y el desarrollo de las redes ha propiciado que muchos usuarios utilicen con menos frecuencia los medios tradicionales de comunicación masiva tales como la radio y la televisión. Los usuarios desean acceder a los medios audiovisuales desde sus ordenadores. Por lo valioso que son los materiales audiovisuales en el proceso de enseñanza-aprendizaje, ya que enriquecen, profundizan y complementan la entrega de los contenidos, su transmisión a través de las redes de computadoras ha ido en ascenso en los últimos años.

La transmisión de video sobre redes de telecomunicaciones está llegando al punto de convertirse en un sistema habitual de comunicación debido al crecimiento masivo que ha supuesto Internet en estos últimos años. Se utiliza para ver películas o comunicarse con conocidos, también para dar clases remotas, hacer diagnósticos en medicina, impartir videoconferencia, como distribuidor de TV, video bajo demanda y multimedia en Internet(1).

No obstante, el audio y el video todavía resultan demasiado "pesados" para los anchos de banda con que cuentan en general los usuarios. Debido a la necesidad de su uso que se plantea en el presente y futuro, se han proporcionado distintas soluciones y sucesivos formatos para mejorar su transmisión(1).



Muchos de los centros educacionales, empresas e instituciones, que cuentan con una infraestructura de redes, desean transmitir a través de ella contenidos audiovisuales previamente almacenados y/o en vivo. Para lograr este objetivo se utilizan aplicaciones capaces de publicar y administrar contenidos multimedia en un entorno web a través del uso de sistemas de streaming de video, produciéndose así un aumento en la demanda de productos que satisfagan estas necesidades. Por esta razón en el polo productivo Video y Sonido Digital perteneciente a la Facultad 9 de la Universidad de las Ciencias Informáticas (UCI) de Cuba, han incluido dentro de sus prioridades la realización de una plataforma de publicación y administración de contenidos audiovisuales a través de la red de datos (Plataforma VideoWeb). Para su desarrollo se han empleado solamente herramientas de software libre, posibilitando su uso y comercialización, sin restricciones por concepto de licencias de software propietario.

En las aplicaciones web dinámicas –como la que se pretende realizar– la formación de una página, tanto de gráficos como de contenidos, se conforma a partir de la lectura de determinados datos residentes en una base de datos. La manipulación de la información dentro de aplicaciones como esta –que necesitan interactuar con mucha información–, constituye un eslabón fuerte y de gran importancia para la implementación satisfactoria de las diferentes funcionalidades que posee. Dichas funcionalidades pueden ser la realización de búsquedas de contenidos, foros, encuestas, sistemas de noticias y la emisión de reportes estadísticos. La pérdida de la información puede ocasionar que esta no llegue de manera eficiente, constante e inmediata, provocando inestabilidad en el producto.

De lo anterior surge el siguiente **problema científico**: “La no existencia de una estructura para el correcto almacenamiento de la información correspondiente a las medias, su gestión y administración que garanticen el funcionamiento de una Plataforma VideoWeb.”

Se tiene como **objeto de estudio** el diseño de bases de datos para guardar y gestionar información de medias. El **campo de acción** estará centrado en el diseño de bases de datos para la Plataforma VideoWeb.

Se plantea como **objetivo general** de la investigación: Desarrollar la base de datos para la Plataforma VideoWeb que permita el correcto almacenamiento de la información, su gestión y administración; y como **objetivos específicos** diseñar la base de datos para la Plataforma VideoWeb e implementar la base de datos para la Plataforma VideoWeb.



Como **idea a defender** se plantea: Si se diseña una base de datos para la Plataforma VideoWeb se garantizará el correcto almacenamiento de la información referente a las medias, su gestión y administración, asegurándose el correcto intercambio de información dentro de la plataforma y el desarrollo exitoso del proyecto.

Para cumplir el objetivo del presente trabajo se trazaron varias **tareas de investigación**, estas son:

1. Analizar los diferentes sistemas similares a la Plataforma VideoWeb existentes a nivel nacional e internacionalmente.
2. Analizar las tendencias y tecnologías actuales de bases de datos a nivel nacional e internacional.
3. Seleccionar el sistema gestor de base de datos.
4. Definir la estrategia y la tecnología de acceso a datos a utilizar.
5. Definir los modelos de despliegue de la base de datos.
6. Analizar la redundancia de la información en la base de datos.
7. Realizar el diseño lógico de la base de datos.
8. Implementar el modelo físico de la base de datos.
9. Comprobar la integridad de los datos en la base de datos.
10. Comprobar el funcionamiento de la base de datos.

Con el correcto cumplimiento de las tareas se esperan obtener como **posibles resultados o aportes prácticos** el diagrama de clases persistentes refinado, el modelo de datos, el diseño del modelo lógico y la implementación del modelo físico de la base de datos.

Para el desarrollo completo del trabajo y su total entendimiento se hizo necesario emplear métodos de investigación tanto teóricos como empíricos, dentro de los teóricos se encuentra el: **Análisis Histórico – Lógico**, que se utilizó para conocer la evolución de las tendencias y de las tecnologías de bases de datos, así como de los gestores de bases de datos y del almacenamiento de medias desde inicios del pasado curso escolar (septiembre de 2007) hasta la actualidad (diciembre de 2008), determinando cuáles son las tendencias actuales y qué gestor de bases de datos es el más indicado para el almacenamiento de las medias, seleccionando el que más se acoja a las necesidades del producto. El método **Analítico – Sintético** se utilizó para llegar a comprender con más exactitud cómo es el proceso de almacenamiento de medias en una base de datos, así como la estructura que debe poseer la información para que el



producto que se desea obtener trabaje de forma más eficiente, constante e inmediata. Esto permitirá conocer cómo realizar de forma más eficiente una base de datos para la Plataforma VideoWeb, permitiendo que este producto interactúe con una base de datos robusta y segura, además de que posea una estructura adecuada. Como método empírico se utilizó la **entrevista** realizada al desarrollador y actual administrador del Portal Inter-nos con el objetivo de comprender cómo se realiza el almacenamiento de la información y las medias, y de conocer cómo se encuentra conformada la base de datos en el Portal Inter-nos, que es una plataforma que brinda servicios de audio y video por la red dentro de la Universidad de las Ciencias Informáticas (UCI) y constituye uno de los principales antecedentes de la plataforma.

El presente documento se estructura en tres capítulos y las correspondientes conclusiones, recomendaciones, referencias bibliográficas, bibliografía y glosario de términos.

Capítulo 1: Fundamentación teórica.

En este capítulo se realiza un estudio acerca de las bases de datos, se analizan las herramientas que existen para su diseño, así como los principales gestores de estas y las herramientas más usadas para el diseño de la capa de acceso a datos, entre otros aspectos.

Capítulo 2: Descripción y análisis de la solución propuesta.

En este capítulo, basado en el estudio realizado en el capítulo anterior, se seleccionan y justifican cuáles herramientas se emplearán en el desarrollo de la solución. Además se explica la solución propuesta.

Capítulo 3: Validación del diseño realizado.

En este capítulo se realiza la validación del diseño de la base de datos y se analiza el nivel de normalización en que se encuentra el esquema de relación de la base de datos.



Capítulo 1

Fundamentación Teórica

1.1. Introducción.

Este capítulo presenta una breve reseña del tema de las Bases de Datos (BD) en la actualidad a nivel internacional, nacional y de la Universidad de las Ciencias Informáticas (UCI), así como las tendencias, técnicas, tecnologías y metodologías utilizadas en el mundo para su tratamiento. Se da una descripción de los principales conceptos relacionados con el manejo de la información más específica en las bases de datos, así como a los asociados al dominio del problema. Se explican además las diferentes fases por las que se debe transitar mediante el diseño de una base de datos, las cuales son: diseño conceptual, diseño lógico y diseño físico. También se detallan los modelos más utilizados en la actualidad para diseñar base de datos, haciendo énfasis en los dos que mayor auge han adquirido en los últimos años y se analizan los principales sistemas gestores de base de datos y herramientas de diseño de base de datos. Con todo esto se realiza un primer acercamiento a las tendencias y tecnologías sobre las que se apoyará la propuesta final.

1.2. Plataforma Web.

Una plataforma Web es un sistema formado por un conjunto de componentes hardware y software que proporcionan capacidades (servicios) sobre las que se deberá apoyar cualquier aplicación software y cuyo funcionamiento es a través de Internet.(2)

El resultado final debe ser un conjunto (plataforma) formado por componentes horizontales o herramientas de colaboración (gestor de usuarios, gestores de contenido y gestores documentales) que pueden reutilizarse en las distintas aplicaciones específicas que se introduzcan.(3)

Analizando las definiciones anteriores se puede llegar a la conclusión de que una Plataforma Web es un sistema informático formado por componentes de hardware y software que brinda servicios sobre Internet y que incorpora gestores de usuarios, gestores de contenido y gestores documentales, los cuales facilitan la seguridad y la gestión de la información.



1.3. Transmisión de vídeo y audio.

La transmisión de video y audio digital no es más que el flujo de bits enviado de un punto a otro mediante la red de datos. La transmisión digital y la distribución de información audiovisual permiten la comunicación multimedia sobre las redes que soportan la comunicación de datos, brindando la posibilidad de enviar imágenes en movimiento y sonido a lugares remotos.

1.4. Plataforma VideoWeb.

La Plataforma VideoWeb es un sistema informático formado por componentes de hardware y software que permite la transmisión de video y audio digital. Brinda servicios de video y audio sobre Internet e incorpora gestores de usuarios, gestores de contenido audiovisual, gestores de contenido y gestores documentales, los cuales facilitan la gestión de la información audiovisual.

1.5. Introducción a las Bases de Datos.

Los datos constituyen la parte esencial de un sistema de información y son los que justifican su existencia. Para organizar y gestionar estos datos en el computador, se han desarrollado técnicas cuya evolución ha estado determinada, principalmente, por el desarrollo de la tecnología de los computadores, así como por los requisitos y necesidades planteadas por los usuarios(4).

Los Sistemas de Bases de Datos ocupan hoy en día un lugar muy importante en todos los sistemas de información, ya que prácticamente todas las aplicaciones o soluciones a problemas utilizando informática, hacen uso de las bases de datos(5).

Gran parte de los sistemas informáticos interactúan con una amplia gama de información para lograr su correcto funcionamiento. Con frecuencia, los sitios generados dinámicamente utilizan una base de datos para almacenar su contenido. En estos sitios, las páginas se forman a partir del contenido, fusionado con plantillas en el momento de la solicitud, para crear la página que se va a entregar. Por esto los sistemas de bases de datos ocupan un lugar importante como medio de almacenamiento en los sistemas informáticos que necesitan la organización y gestión de grandes volúmenes de información.

Las técnicas de Bases de Datos representan la tecnología informática disponible para la organización y gestión de grandes volúmenes de datos en un computador. Se puede afirmar que el núcleo de todo



sistema de información actual es una Base de Datos, y que el diseño y creación de ésta constituyen una etapa importante en la construcción del sistema(4).

1.5.1. Breve historia de los sistemas de almacenamiento de datos.

Los sistemas de almacenamiento de datos han evolucionado considerablemente desde sus inicios con los archivos secuenciales hasta los Sistemas de Gestión de Bases de Datos que se conocen hoy en día. Los primeros brindaban un acceso muy rápido a la información, pero para acceder a una posición se debía hacer de forma secuencial, es decir, se tenía que recorrer el archivo entero, siendo esto su principal inconveniente. Como solución aparecieron los archivos indexados, donde el acceso ya podía ser aleatorio (acceder de una vez a la posición deseada del archivo).

Antes del surgimiento de las bases de datos, la manera en que se almacenaba y trabajaba con la información era en los sistemas de ficheros. Estos sistemas surgieron con el objetivo de informatizar el manejo de los archivadores manuales, logrando proporcionar un acceso más eficiente a los datos.

Un sistema de ficheros es un grupo de programas con los que trabajan los usuarios finales, donde cada programa controla sus propios datos, es decir, que no hay un lugar físico centralizado donde almacenen los datos y al cual puedan acceder todos los que lo necesiten. Esto trae consigo dificultades tales como: separación y aislamiento de los datos, duplicación de datos, dependencia de datos, formatos de ficheros incompatibles, consultas fijas y proliferación de programas de aplicación(6).

Por la necesidad de lograr una mayor organización de los datos y lograr compartirlos entre varias máquinas surgió el NFS (Sistema de Archivos de Red), posibilitando que distintos sistemas conectados a una misma red accedan a ficheros remotos como si fueran locales. Más tarde para evitar fallos en los sistemas de archivos y una mayor protección de datos, aparecieron los dispositivos RAID (Redundant Array Of Independent/Inexpensive Disks), lográndose aumentar la integridad de los datos en los discos.

Luego aparece la necesidad de realizar un trabajo más efectivo con la información y es que surgen las Bases de Datos (BD) en el año 1960, aunque no se empiezan a usar hasta el año 1971. Estas fueron creadas con el objetivo de solucionar los problemas que existían con el uso, en las aplicaciones software, de los ficheros como medio de almacenamiento.



Debido a que los programas eran cada vez más complejos y grandes, se requería almacenar los datos en sistemas que garantizaran un cierto número de condiciones, permitiendo operaciones complejas sin que se violaran estas restricciones y que garantizaran la seguridad de la información. Respondiendo a estas necesidades surgieron las Bases de Datos Jerárquicas, donde los datos se situaban siguiendo una jerarquía, pero tenían el problema de que los accesos a éstos eran unidireccionales(7).

Para dar absoluta libertad a las relaciones entre tablas surgieron las Bases de Datos Relacionales en los años 80, y con ellas los Sistemas de Bases de Datos Relacionales (SBDR). Esta nueva forma de almacenamiento de la información aportó dos características muy importantes: las propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) y un lenguaje común de acceso a los datos: SQL, facilitando la programación de aplicaciones con Bases de Datos, y consiguiendo que los programas sean independientes de los aspectos físicos de la Base de Datos(4).

Con el transcurso del tiempo aparecieron, en los años 90, las Bases de Datos Distribuidas. Estas BD no se almacenan completamente en una localidad central, sino que se distribuyen en una red de localidades que pueden estar geográficamente separadas y conectadas por enlaces de comunicaciones. Cada localidad tendrá su propia BD y está capacitada para acceder a los datos de otras localidades(4).

1.6. ¿Qué es una Base de Datos?

Existen varios criterios o definiciones de las bases de datos enunciadas por diferentes autores entre los que se encuentran:

- Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción(6).
- Un sistema de bases de datos es básicamente un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones(8).
- 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(9).

- Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones(5).

A modo de resumen una base de datos es una colección de datos variables en el tiempo interrelacionados entre sí a través relaciones lógicas. Puede definirse también como el conjunto de información relacionada que se encuentra agrupada o estructurada. Un archivo en sí no constituye una base de datos, sino la forma en que está organizada la información es lo que da origen a la base de datos. Puede referirse también al conjunto de datos homogéneos, ordenados de una forma determinada, refiriéndose a una organización, materia o problema determinado. En fin una base de datos no es más que un gran almacén de datos definido una sola vez y que se utiliza al mismo tiempo por muchos departamentos y usuarios.

Una base de datos está compuesta por varios componentes: (8)

- **Hardware:** Se refiere a los dispositivos de almacenamiento en donde reside la Base de Datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- **Software:** Está constituido por un conjunto de programas que se conoce como Sistema Gestor de Base de Datos, el cual maneja todas las solicitudes formuladas por los usuarios de la misma.
- **Usuarios:** Existen tres clases de usuarios relacionados con una Base de Datos:
 - El programador de aplicaciones, quien crea programas de aplicación que utilizan la BD.
 - El usuario final, quien accede a la BD por medio de un lenguaje de consulta o de programas de aplicación.
 - El Administrador de la BD, quien se encarga del control general del Sistema Gestor de Base de Datos.



1.7. Sistema Gestor de Base de Datos (SGBD).

Existen varias definiciones de que es un Sistema Gestor de Base de Datos, algunos de ellos se muestran a continuación:

- Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta(10).
- El sistema de gestión de la base de datos es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma(6).
- El software que permite la utilización y/o actualización de los datos almacenados en una o varias Bases de Datos, desde diferentes puntos de vista a la vez, por uno o varios usuarios, se denomina Sistema Gestor de Bases de Datos(8).
- Siendo los SGBD un tipo de software muy específico, dedicado a servir de interfaz entre la BD, el usuario y las aplicaciones que la utilizan y que prestan servicios para el desarrollo y el manejo de BD(11).

Un Sistema Gestor de Base de Datos no es más que un tipo de aplicación informática o software que actúa como interfaz entre el usuario y la Base de Datos. El SGBD funciona como un intermediario para que el usuario pueda comunicarse con la BD en términos abstractos y de una forma práctica y eficiente, abstrayéndolo del modo físico de almacenamiento de la información en la computadora, y del método de acceso empleado, permitiéndole definir, crear y mantener la base de datos, y proporcionándole el acceso controlado a la misma.

El SGBD es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. En general, un SGBD proporciona los siguientes servicios: (6)

- Permite la definición de la base de datos mediante el lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.
- Permite la inserción, actualización, eliminación y consulta de datos mediante el *lenguaje de manejo de datos*. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los



sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar.

- Proporciona un acceso controlado a la base de datos mediante:
 - un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos;
 - un sistema de integridad que mantiene la integridad y la consistencia de los datos;
 - un sistema de control de concurrencia que permite el acceso compartido a la base de datos;
 - un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del *hardware* o del *software*;
 - un diccionario de datos o catálogo accesible por el usuario que contiene la descripción de los datos de la base de datos.

1.8. Clasificaciones de las Bases de Datos.

Teniendo en cuenta varios criterios, las bases de datos pueden ser clasificadas de acuerdo a sus características y propósitos. Algunas de estas clasificaciones se muestran a continuación:

1.8.1. Según la forma en que cambia la información almacenada.

1.8.1.1. Bases de datos estáticas.

Son aquellas bases de datos en las que no se puede modificar la información que está guardada, es decir, solo se puede consultar, son utilizadas preferentemente para guardar datos históricos con los que luego se podrán realizar estudios sobre su comportamiento, lo cual ayudará a tomar decisiones(5). Es decir, son BD de sólo lectura, en las que solo se copia información, nunca se elimina. Comúnmente se les llama bases de datos históricas, porque recogen toda la información desde que se creó hasta el momento en que se está analizando.

1.8.1.2. Bases de datos dinámicas.

Son aquellas bases de datos en las que la información almacenada está en constante cambio, ya que sobre ella se pueden realizar operaciones tales como: adicionar, actualizar, eliminar, etc., así como las operaciones de consultas(5). Éstas son BD donde la información almacenada se modifica con el tiempo.



Un ejemplo de donde pueden ser utilizadas estas BD es en el almacenamiento de la información de los medicamentos en una farmacia, en un videoclub, etc.

1.8.2. Según la información que almacenan.

1.8.2.1. Bases de datos bibliográficas.

Almacenan la información necesaria para poder localizar la fuente primaria. Estas BD solo contienen una porción de la información de lo que se está almacenando y son utilizadas para guardar las referencias a un contenido dado. Un registro típico de una base de datos bibliográfica contiene información sobre el título, fecha de publicación, autor, editorial, de una determinada publicación, etc. Puede contener un fragmento de la publicación original, pero nunca el texto completo(5).

1.8.2.2. Bases de datos de texto completo.

Almacenan toda la información íntegra de las fuentes primarias, no fragmentos, ni referencias(5). Un ejemplo es el archivo de todo el contenido de las ediciones de una colección de revistas científicas.

1.8.2.3. Directorios.

Este es el caso de directorios digitales como los que edita la Empresa de Telecomunicaciones de Cuba (ETECSA) con los números telefónicos de sus usuarios, ya sean privados o estatales, donde se puede encontrar el nombre al que pertenece dicho número y la dirección dónde se encuentra(5). Ejemplo de ellas son las guías telefónicas en formato electrónico; los bancos de imágenes, audio, video, multimedia, etc.

1.8.2.4. Banco.

Estas, según el formato de la información que almacenan, pueden ser: de audio, video, imágenes, multimedia, etc., en dependencia de la información que almacenan.(5)

1.9. Modelos de Bases de Datos.

Un modelo de datos es básicamente una descripción de lo que se conoce como contenedor de datos – donde se guarda la información–, así como de los métodos para almacenar y recuperar información de esos contenedores(8).



Existen varios modelos que con frecuencia son utilizados para el diseño de las bases de datos, entre ellos el Modelo Orientado a Objetos y el Modelo Relacional. Otros modelos, no tan populares, son el Modelo Jerárquico, el Modelo de Red, el Modelo de Datos Objeto Relacional y el Modelo Multidimensional.

1.9.1. Bases de Datos Jerárquicas.

Las Bases de Datos Jerárquicas surgieron a mediados de 1960 y son Bases de Datos (BD) que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en forma similar a un árbol, donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres se le conoce como raíz, y a los nodos que no tienen hijos se les conoce como hojas. Una de las principales limitaciones de este modelo, es su incapacidad de brindar eficientemente una solución a la redundancia de datos(7).

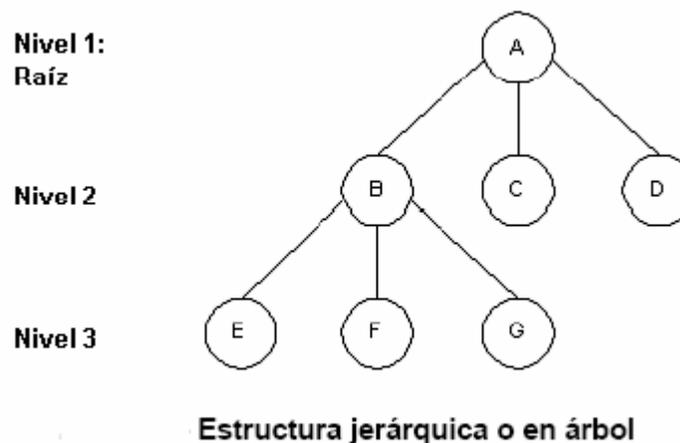


Figura 1: Estructura de una Base de Datos Jerárquica.

Las BD jerárquicas 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.

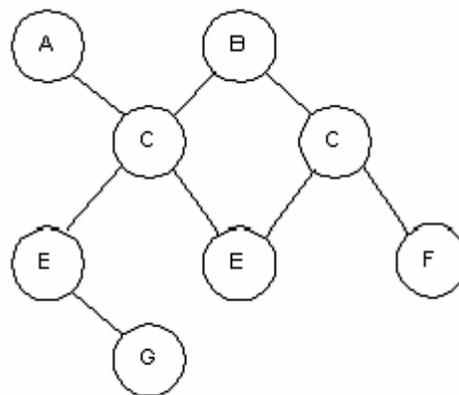
Los productos jerárquicos, exactamente el IMS y el DL/I de IBM como máximos exponentes de estos sistemas, consiguieron altas cuotas de mercado, aunque la actual difusión de la tecnología relacional los ha llevado a convertirse en sistemas superados. Lo cual no quiere decir que no persistan todavía importantes aplicaciones soportadas en estos productos que están trabajando, por su eficiente respuesta, a satisfacción de sus usuarios. Las aplicaciones desarrolladas sobre ellos se mantienen sin apenas cambios.



1.9.2. Bases de Datos de Red.

Es un modelo ligeramente distinto del jerárquico, y surge a finales de 1960. Su diferencia fundamental es la modificación del concepto de un nodo, permitiendo que un mismo nodo tenga varios padres (algo no permitido en el modelo jerárquico)(7).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos. Pero aún así, la dificultad que implica administrar la información en una Base de Datos de Red, ha significado que sea un modelo utilizado en su mayoría más por programadores que por usuarios finales. Un ejemplo de Base de Datos de Red es el sistema denominado IDMS/R.



Estructura de datos de red

Figura 2: Estructura de una Base de Datos de Red.

1.9.3. Bases de Datos Relacional.

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de entidades (tablas), compuestas por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla)(7).

El modelo relacional de datos tiene varios aspectos que lo caracterizan, desde el punto de vista estructural se puede plantear que este modelo organiza la información en forma de tablas y de esta manera es que sus usuarios la perciben y no de otra. Desde el punto de vista de la integridad se puede decir que el



modelo establece varias restricciones de integridad, las cuales deben ser cumplidas por todas sus tablas; y desde el punto de vista de la manipulación, tiene definidos un grupo de operadores a través de los cuales interactúa con la información almacenada en las tablas de la base de datos y sus resultados los devuelve en tablas. Entre estos operadores se encuentran proyectar, restringir y juntar. Debido a que la información consultada en cada operación se encuentra en forma de tabla, existe la posibilidad de concatenar operadores unos con otros. Este modelo también está capacitado para hacer procesamiento de conjunto. Esto significa que al devolver el resultado de cualquier operación, el resultado lo devuelve en tablas que están conformadas por un conjunto de filas, no por filas individuales. Además está compuesto por 5 componentes que lo identifican: un conjunto abierto de tipos escalares, un generador de tipos de relación y una interpretación propuesta de dichos tipos, herramientas para definir variables de dichos tipos de relación generados, una operación asignación relacional para asignar valores de relación a las variables de relación mencionadas y un conjunto abierto de operadores relacionales genéricos para derivar valores de relación de otros valores de relación(8).

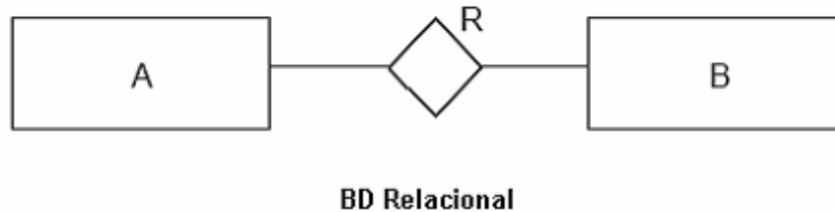


Figura 3: Estructura de una Base de Datos de Relacional.

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). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la Base de Datos. La información puede ser recuperada o almacenada por medio de “consultas” que ofrecen una amplia flexibilidad y poder para administrarla. Ejemplos de Bases de Datos Relacionales son: Oracle, Informix, MySql, etc.

1.9.4. Bases de Datos Multidimensionales.

Un enfoque que ha cobrado actualmente fuerza es el Procesamiento Analítico En Línea (en inglés, denominado On-Line Analytical Processing u OLAP). Su nombre se deriva del contraste con el procesamiento de transacciones en línea (On-Line Transaction Processing, OLTP). Mientras que el OLTP



depende de Bases de Datos Relacionales, el OLAP ha desarrollado una tecnología de Bases de Datos Multidimensionales. Estas Bases de Datos fundan los cimientos para el desarrollo de los cálculos y análisis multidimensionales que requiere la inteligencia empresarial(4).

Las Bases de Datos Multidimensionales son BD estructuradas como un hipercubo, con un eje por dimensión, de estructura basada en dimensiones orientadas a consultas complejas y alto rendimiento. Estos modelos tienen gran aplicación en la bioinformática, debido al almacenamiento de grandes cantidades de información biológica(12).

A continuación se muestra una representación espacial de una variable multidimensional con una, dos y tres dimensiones. En esta figura los cubitos representan valores de dimensión, y las esferas son datos(4).

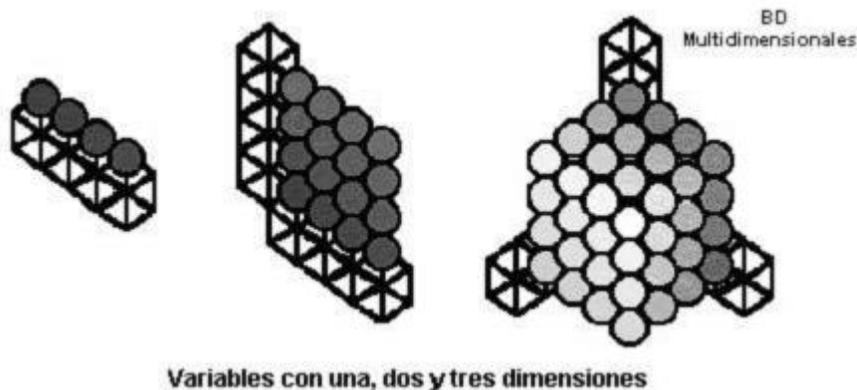


Figura 4: Representación espacial de una variable multidimensional.

Una variable unidimensional podría ser el cambio del euro con el dólar, que sólo varía en la dimensión <tiempo>. Los cubitos serían, por ejemplo, los días del año y las esferas serían los valores numéricos correspondientes al cambio monetario en cada momento. Un ejemplo de variable de dos dimensiones es el número de habitantes, que se mueve por las dimensiones <Geografía> y <tiempo>. Finalmente, los ingresos de una organización podrían almacenarse mediante una variable de tres dimensiones: <producto>, <Geografía> y <tiempo>.

1.9.5. Bases de Datos Orientadas a Objetos.

A finales de los 80's aparecieron las primeras Bases de Datos Orientadas a Objetos (BDOO). Diseñadas para ser eficaces, desde el punto de vista físico, con el objetivo de almacenar objetos complejos y



métodos(8).

El modelo de objetos conjuga de forma centralizada los conceptos de abstracción, jerarquía, encapsulación, modularidad, persistencia, tipos y concurrencia, donde los primeros cuatro conceptos son fundamentales, lo cual significa que si un modelo carece de estos elementos pues no es orientado a objetos. Además se basa en los principios de la ingeniería. Este modelo refleja una nueva forma de pensar acerca de la de descomposición, haciendo que el diseño orientado a objetos sea diferente a la forma que existía de diseño estructurado hasta el momento, y que por consiguiente su arquitectura sea diferente también. Esto ocurre sencillamente porque los métodos de diseño orientado a objetos utilizan la programación orientada a objetos, mientras que el diseño estructurado se basa en la programación estructurada.

El desarrollo del diseño orientado a objetos no niega las ideas existentes, sino que se basa en ellas y las mejora. Un modelo orientado a objetos tiene obligatoriamente que cumplir con lo siguiente: la estructura básica de trabajo son los objetos, no algoritmos; donde cada objeto no es más que una instancia de una clase ya definida y que dichas clases estarán relacionadas únicamente por relaciones de herencia. En caso de que no se cumpla uno de las condiciones antes mencionadas, pues no se estaría en presencia de un modelo orientado a objetos(13).

Una BD orientada a objetos es una BD que incorpora todos los conceptos importantes del paradigma de objetos:(14)

- 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.
- Polimorfismo: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Este modelo es propio de los modelos informáticos orientados a objetos. Él trata de almacenar en la Base de Datos los objetos completos –estado y comportamiento–. Combina las mejores cualidades de los archivos planos, las Bases de Datos Jerárquicas y Relacionales. Las Bases de Datos Orientadas a



Objetos representan la sucesiva evolución de las Bases de Datos para soportar análisis, diseño y programación orientada a objetos.

Entre los Gestores de Bases de Datos Orientados a Objetos más conocidos se encuentran los siguientes:
(7)

- **Gemstone:** Proporciona control de concurrencias y recuperación de la información almacenada, así como gestión de almacenamiento secundario. Soporta acceso concurrente y métodos para mantener la seguridad y la integridad de las BD.
- **Vbase:** Surge en 1987 y enfatiza algunas características de la orientación a objetos.

También existen otros como: Orion, PDM, Iris y O2, entre otros.

1.9.6. Bases de Datos Objeto Relacionales.

Las Bases de Datos Objeto Relacionales son Bases de Datos que desde el modelo relacional evolucionan hacia Bases de Datos más extensas y complejas incorporando, para obtener este fin, conceptos del modelo orientado a objetos. Se puede afirmar que un Sistema de Gestión Objeto-Relacional (SGBDOR) contiene dos tecnologías: la tecnología relacional y la tecnología de objetos(8).

Con las Bases de Datos Objeto-Relacional, se pueden crear nuevos tipos de datos, que permiten gestionar aplicaciones más complejas con una gran riqueza de dominios. Tienen la posibilidad de incluir el chequeo de las reglas de integridad referencial a través de los disparadores, entre otras características. Uno de los gestores de Bases de Datos Objeto Relacional que más se utiliza en la actualidad es PostgreSQL(4).

Los modelos de datos relacionales orientados a objetos extienden el modelo de datos relacional, proporcionando un sistema de tipos más rico y que se adapte mejor a las nuevas aplicaciones, que requieren guardar un tipo de datos más complejos. Permiten que los atributos de las tuplas tengan tipos complejos. Además, la BD es relacional por lo que conserva su rapidez y eficiencia, y permite hacer uso de nuevos elementos, como las relaciones de herencia, que modelan los objetos que serán guardados en dicha BD, por lo que el analista y el diseñador ven un modelo orientado a objetos(14).



1.9.7. Bases de Datos Documentales.

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesauros es un sistema de índices optimizado para este tipo de BD(14).

1.9.8. Bases de Datos Deductivas.

Un SBD deductivo, es un SBD pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la BD. También las BD deductivas son llamadas BD lógica, a raíz de que se basan en lógica matemática.(14)

1.9.9. Bases de Datos Distribuidas.

La base de datos distribuida consiste en un almacén de datos, es decir, en un conjunto de datos almacenado de manera sistemática siempre dispuesto a ser utilizado. Pero tiene una particularidad que la diferencia y consiste en que estos datos están almacenados en distintas máquinas que integran un sistema y que tienen conexión entre sí.

La BD está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las BD de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.(14)

1.10. Fases del Diseño de Base de Datos.

El proceso de diseño de una base de datos cuenta con tres fases fundamentales: Diseño Conceptual, Diseño Lógico y Diseño Físico.

1.10.1. Diseño Conceptual.

El objetivo del diseño conceptual, también conocido como modelo conceptual, y que constituye la primera fase de diseño, es obtener una buena representación de los recursos de información, con independencia de usuarios, aplicaciones en particular y sin considerar aspectos como eficiencia de la computadora. Esta primera fase consta de dos momentos: análisis de requisitos, donde se centra el trabajo en definir qué es lo que se va a representar, y la conceptualización, donde se piensa en cómo se va a proceder para representar lo antes definido.(5)



1.10.2. Diseño Lógico.

El diseño lógico parte del esquema conceptual y genera el esquema lógico. Un esquema lógico es la descripción de la estructura de la base de datos que puede procesarse por Sistema Gestor de Base de Datos. El objetivo principal del diseño lógico es obtener un esquema lógico eficiente en cuanto a operaciones de consulta y actualización. El modelo relacional es el único modelo que ha permitido abordar la fase de diseño lógico aplicando una teoría formal.(5)

1.10.3. Diseño Físico.

El objetivo del diseño físico es conseguir una instrumentación lo más eficiente posible del esquema lógico. Para esto se analizan aspectos como las características del Sistema Operativo, el Sistema Gestor de Base de Datos, la Herramienta para realizar el diseño, aspectos relacionados con el rendimiento y los requisitos de procesos así como las características del hardware, en fin, cualquier factor cercano con la computadora, para con ello lograr optimizar el consumo de recursos, minimizar el espacio de almacenamiento, proporcionar la seguridad máxima, disminuir los tiempos de respuesta y evitar las reorganizaciones.(5)

1.11. Principales Tendencias de Bases de Datos.

La utilización de Bases de Datos como plataforma para el desarrollo de Sistemas de Aplicación en las organizaciones se ha incrementado notablemente en nuestros días. Actualmente las BD ofrecen numerosas funcionalidades de gran importancia para el almacenamiento y la manipulación de la información en las instituciones, pero siguen creciendo las demandas de nuevas operaciones que faciliten una mayor eficiencia en el manejo de los datos. Esto trae consigo el desarrollo de técnicas y herramientas que brinden una solución eficiente a las solicitudes planteadas por los usuarios.(4)

Algunas de las tendencias de Bases de Datos son citadas a continuación.(4)

- La explotación efectiva de la información brinda ventajas competitivas a las organizaciones.
- El uso de las Bases de Datos Distribuidas y Multidimensionales se incrementa de manera considerable en la medida en que la tecnología de comunicación de datos brinde más facilidades para ello.



- El uso de las Bases de Datos facilita y soporta en gran medida a los sistemas de información para la toma de decisiones.
- Los lenguajes de consulta (SQL) permitirán el uso del lenguaje natural para solicitar información de la Base de Datos, haciendo más rápido y fácil su manejo.

En los últimos años se ha visto un gran crecimiento en la capacidad de generación y almacenamiento de información, debido a la creciente automatización de procesos. Desgraciadamente no ha existido un desarrollo equivalente en las técnicas de análisis de información, por lo que existe la necesidad de una nueva generación de técnicas y herramientas computacionales con la capacidad de asistir a usuarios en el análisis automático e inteligente de datos. El procesar automáticamente grandes cantidades de datos para encontrar conocimiento útil para un usuario y satisfacerle sus metas, es el objetivo principal del área de Descubrimiento de Conocimiento en Bases de Datos o KDD (Knowledge Discovery from Data Base). Este es el campo que está evolucionando para proporcionar soluciones al análisis automatizado de datos.

1.12. Principales Gestores de Bases de Datos.

El software de BD ha experimentado un auge extraordinario, además de rapidez y efectividad en los procesos. Con la existencia de múltiples entornos de desarrollo y la notable demanda de soluciones informáticas, han surgido multitud de gestores de BD.

Siendo los SGBD un tipo de software muy específico, dedicado a servir de interfaz entre la BD, el usuario y las aplicaciones que la utilizan y que prestan servicios para el desarrollo y el manejo de BD. Dentro de los principales SGBD propietarios más potentes se encuentra Oracle y Microsoft SQL Server, y dentro del software libre, existen opciones muy completas como: MySQL y PostgreSQL, aunque el MySQL presenta el uso de una *licencia dual* o doble licenciamiento.

Algunas de las Ventajas y Desventajas de los SGBD.(11)

Ventajas:

- Eliminan las inconsistencias en los datos.
- Permiten compartir los mismos datos entre diferentes aplicaciones con distintas necesidades.
- Ahorran espacio de almacenamiento al no existir redundancia o ser ésta escasa.



- Mejoran la seguridad de los datos, pues normalmente incorporan mecanismos de seguridad en el propio SGBD.
- Permiten la creación de entornos de alta disponibilidad, pues con algunos SGBD es posible llegar a disponer de aplicaciones funcionando ininterrumpidamente.
- Mejora en los servicios de copias de seguridad y de recuperación ante fallos.

Desventajas:

- Requieren una gran cantidad de espacio en disco y de memoria para trabajar de forma eficiente.
- Vulnerable a los fallos. El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse.

1.12.1. Sistema Gestor de Bases de Datos Oracle.

Es un sistema de gestión de BD (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), que utiliza arquitectura cliente/servidor, fabricado por Oracle Corporation. Se considera como uno de los sistemas de BD más completos, destacando su: soporte de transacciones, estabilidad, escalabilidad y el ser multiplataforma.(15)

Ventajas: (11)

- Es manejador de BD relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.
- Puede ser usado tanto para el manejo de información personal, como para gigantescas bibliotecas multimedia, y corre en laptops, computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.
- Soporta la mayoría de los lenguajes de computación al igual que 26 idiomas diferentes.
- Corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código, esto se debe a que más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas.
- Soporta datos alfanuméricos ubicados en las tradicionales "filas y columnas" de las BD, sino que también soporta textos sin estructura, imágenes, audio y video.



- Incluye mejoras de rendimiento y de utilización de recursos.
- Soporta aplicaciones de procesamiento de transacciones online (OLTP) y de data warehousing mayores y más exigentes.
- Está disponible en múltiples plataformas como Windows, Linux, todas las versiones de Unix.
- Ofrecidas por diversas empresas como IBM, Sun, Digital, HP, Sequent, etc. y también en VAX-VMS, así como en MVS. El ambiente multiplataforma de Oracle, lo convierte en una verdadera solución empresarial.
- Bloqueo y concurrencia: El lector verá los datos tal y como estaban antes de que el que escribe comience a cambiarlos (hasta que este haga commit).
- Disponibilidad. Soporta ambientes de cluster en modo activo-pasivo, es decir que un solo nodo utiliza la base de datos mientras el/los otro/s nodo/s está/n pendientes de entrar en funcionamiento en el momento que el servidor primario tenga una falla. Permite tener copias de la base de datos productiva en lugares lejanos a la ubicación principal, esta funcionalidad se le denomina Oracle Standby Database. Las copias de la Base de Datos productiva pueden estar en modo de lectura solamente.

Desventajas: (11)

- Es un software propietario, además su elevado precio (la BD más cara) hace que solo se vea en empresas muy grandes y multinacionales, por norma general.
- Elevado costo de soporte técnico.
- Infinidad de cursos para poder manejar una sola línea (DBA, Desarrollo, Recursos Humanos, Minería de Datos, entre otras muchas).
- La seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos del 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre del 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

1.12.2. Sistema Gestor de Bases de Datos Microsoft SQL Server 2000.

SQL Server fue desarrollado por Microsoft y permite la gestión de un entorno de base de datos relacional. A través de una interfaz cómoda y fácil de trabajar SQL Server abarca tanto el área de diseño como la de



administración. Es nombrado SQL porque hace uso de este lenguaje para la definición y manejo de los datos, y se llama Server porque dispone de una parte servidora que es responsable de atender a los procesos clientes, que son aquellos que realizan las peticiones al servidor, cumpliendo con una arquitectura cliente-servidor.(5)

La versión más reciente y completa en estos momentos es Microsoft SQL Server 2005.

Ventajas: (11)

- Soporta funciones definidas por el usuario. Esto soluciona los problemas de reutilización del código y da mayor flexibilidad al programar las consultas de SQL.
- Arquitectura de almacenamiento en disco permite la escalabilidad desde BD de equipos portátiles hasta bases de datos empresariales de tamaño de terabyte.
- Permite acceso y la realización de consultas desde URL a través de HTTP, donde el optimizador de consultas con múltiples fases busca el plan óptimo de consultas para mejorar el rendimiento de consultas complejas.
- Soporta consultas en múltiples servidores.
- Simplifica la configuración y gestión de un cluster de caídas. Permite que las BD permanezcan online durante la mayoría de las operaciones. Activa backups instantáneos.
- Estadísticas automáticas. Extrae estadísticas mediante el análisis rápido de una muestra, habilitando el optimizador de consultas para utilizar la información más reciente e incrementar la eficacia de las consultas.
- Los servicios de transformación de datos sirven para importar, exportar y transformar datos heterogéneos.
- Soporte enriquecido de XML. Los desarrolladores Web pueden acceder a los datos utilizando XML sin tener que emplear compleja programación, y los administradores de BD pueden manipular datos fácilmente en formato XML mediante Transaction SQL (T-SQL) y procedimientos almacenados.

Desventajas: (11)

- Las funciones definidas por el usuario tienen algunas restricciones.
- No todas las sentencias SQL son válidas dentro de una función.



- Es un SGBD propietario.

1.12.3. Sistema Gestor de Bases de Datos MySQL.

MySQL es un Sistema de Gestión de Bases de Datos Relacional. Es, probablemente, el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación se debe, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Soporta gran cantidad de tipos de datos para las columnas; tiene gran portabilidad entre sistemas y gestiona usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos.

Es un SGBD cliente/servidor, multihilo y multiusuario. Se compone de un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar. Es el servidor de BD relacionales más popular, desarrollado y proporcionado por MySQL AB. Esta es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de BD MySQL.

Ventajas:

- Dispone de borrados multi-tablas. Al especificar múltiples tablas y la cláusula WHERE, se pueden agregar opciones ORDER BY y LIMIT a las consultas DELETE, para obtener un mejor control sobre cuántos registros son eliminados y el orden en el que son eliminados dichos registros.
- Mejores utilidades de administración (backup, recuperación de errores, etc.), contando con un sistema de replicación multihilo en los servidores esclavos. Si el servidor principal llega a fallar, es ahora mucho más probable que cada esclavo tenga los datos necesarios para hacer por sí mismo una recuperación de los datos y trabajar como si fuera el servidor maestro. Los registros de replicación ahora contienen los marcadores de transacción necesarios para asegurarse que las transacciones son replicadas apropiadamente.
- Soporta cinco tipos de tablas: MyISAM, ISAM, HEAP, BDB (Base de Datos Berkeley), e InnoDB. InnoDB y BDB son tablas transaccionales. Se puede utilizar la sentencia estándar BEGIN WORK seguida de varias consultas y finalizar con un COMMIT o ROLLBACK para completar la transacción



ó se pueden correr en modo AUTOCOMMIT, así que cada consulta es efectivamente una transacción separada.

- Recuperación automática ante fallas. Si MySQL se cuelga o se da de baja de una forma anormal, no suele perder información ni corromper los datos, además InnoDB automáticamente completará las transacciones que quedaron incompletas.
- Integridad referencial. Ahora se pueden definir llaves foráneas entre tablas InnoDB relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra tabla.

Desventajas:

- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.
- Carece de soporte para transacciones, rollback's y subconsultas.

1.12.4. Sistema Gestor de Bases de Datos PostgreSQL.

Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la Universidad de Berkeley.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.(11)

Ventajas:

- DBMS Objeto-Relacional. Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multiusuario, transacciones, optimización de consultas, herencia, y arrays.



- Cliente/Servidor. Usa una arquitectura proceso por usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- Altamente Extensible. Soporta los tipos de datos base, así como: tipo, fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. Además operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Soporte SQL Comprensivo. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- Integridad Referencial. Es utilizada para garantizar la validez de los datos de la BD.
- Lenguajes Procedurales. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Además tiene habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- MVCC (Multi-Version Concurrency Control) Control de Concurrencia Multi-Versión. Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios, es decir permite la lectura sin que sea bloqueada por los que escriben que están actualizando registros.
- Write Ahead Logging (WAL) Esta característica incrementa la dependencia de la BD al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en caso de que la BD se caiga, existirá un registro de las transacciones a partir del cual se podrá restaurar la BD desde el punto en que se quedó.
- Es un gestor magnífico bajo licencia Berkeley Software Distribution (BSD), que posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios Web que posean alrededor de 500.000 peticiones por día. Además por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.

Desventajas:

- Consume bastantes recursos y carga con mucha facilidad el sistema.
- Velocidad de respuesta un poco deficiente al gestionar BD relativamente pequeñas, aunque esta misma velocidad la mantiene al gestionar BD realmente grandes.



1.13. Herramientas CASE (Computer Aided Software Engineering).

Las herramientas CASE son de gran ayuda para el desarrollo de un software. Estas son un grupo de programas que utilizan las personas que intervienen en el desarrollo de un software, como es el caso de los diseñadores, desarrolladores, analistas, entre otros, durante las fases del desarrollo del software (análisis, diseño, implementación, etc.), para agilizar y facilitar su trabajo, ya que dichas herramientas proveen de métodos, técnicas y utilidades que ayudan al perfeccionamiento del desarrollo de sistemas de información, de forma total o parcial.(5)

Entre las herramientas CASE más utilizadas para el diseño de base de datos se encuentran:(5)

1.13.1. ERwin.

PLATINUM ERwin es una CASE empleada en el diseño de base de datos, con la cual se puede hacer un trabajo productivo por las facilidades que brinda para la generación y mantenimiento de aplicaciones de forma sencilla para el diseñador. ERwin permite realizar el diseño del modelo lógico de los requerimientos de información, así como el diseño del modelo físico, ya con nivel mayor, refinando las características de la base de datos diseñada. Con esta herramienta CASE es posible visualizar la estructura de la base de datos diseñada, lo cual tiene como ventaja que el diseñador pueda observar en su totalidad el trabajo realizado, para realizar un análisis y si fuera necesario hacer cambios en busca de optimizar el diseño final. Esta herramienta permite generar, de forma automática, las tablas y el código referente a los stored procedure (procedimientos almacenados) y triggers para los principales tipos de bases de datos. La migración automática garantiza la integridad referencial de la base de datos, es decir, evita la pérdida de información durante este proceso. ERwin puede hacer una conexión entre dos bases de datos, una diseñada y otra sin diseño, estableciendo una transferencia entre ellas y la aplicación de ingeniería reversa. Mediante esta conexión puede automáticamente generar índices, vistas, tablas, reglas de integridad referencial (llaves foráneas, llaves primarias), restricciones de dominios y campos y valores por defecto. ERwin básicamente soporta bases de datos del tipo relacionales SQL, y es compatible con otras como: Oracle, Microsoft SQL Server, Sybase, Microsoft Access, dBase, FoxPro, Informix, SQL Base y SQL Anyware, entre otras. Con un mismo modelo es posible generar múltiples bases de datos o hacer una conversión de una aplicación de una base de datos a otra. Es compatible con los sistemas operativos de la familia Windows, como Windows 95, Windows 98, Windows XP y Windows NT.



1.13.2. EasyCASE.

EasyCASE Profesional es un producto destinado a la generación de esquemas de base de datos e ingeniería reversa. Su objetivo es proporcionar a los usuarios una solución comprensible para el diseño, documentación y consistencia del sistema que se va a desarrollar. Brinda la posibilidad de automatizar las primeras fases del desarrollo de un software (análisis y diseño), para lograr el desarrollo de un trabajo óptimo y eficaz. Forma parte del grupo de herramientas multi-usuarios, permitiendo que varias personas trabajen en un proyecto al mismo tiempo. Esta herramienta permite la conexión de varias personas a un servidor donde está la información que necesitan los integrantes del proyecto para su trabajo. Esta conexión se hará de forma segura, ya que la herramienta controlará el nivel de acceso que podrá tener cada usuario, según el rol que desempeñe, a través del diagrama y diccionario de los datos que bloquean por niveles al registro, al archivo y al proyecto. EasyCASE soporta las siguientes base de datos: Oracle, Postgres, SQL Server, ANSI SQL, Fox Pro, SQL Base, Access, Paradox, Sybase, Clipper, entre otras.

1.13.3. Oracle Designer.

Es un grupo de aplicaciones que automatizan la construcción de aplicaciones cliente/servidor y permiten además el almacenamiento de las definiciones que necesita el usuario de forma rápida y sencilla. Oracle Designer hace uso de un repositorio central para almacenar, en cualquier fase de desarrollo, toda la información generada por cualquier herramienta de Oracle Designer, posibilitando un trabajo fácil para la dirección del proyecto y su equipo de desarrollo. Esta herramienta CASE, puede en el lado del servidor definir, generar y capturar el diseño, por conexión Oracle, los siguientes tipos de bases de datos: Oracle8, Oracle7, Personal Oracle Lite, Microsoft SQL Server, Sybase y ANSI 92, entre otras.

1.13.4. Case Studio.

Case Studio es una herramienta profesional para el diseño de base de datos. Soporta Oracle, MySQL, MS SQL, MaxDB, Firebird, PostgreSQL y otros sistemas. El software permite generar rápidamente diagramas gráficos de las bases de datos relacionales con las que se trabaja, simplificando mucho el trabajo del programador. Permite realizar Diagramas Entidad-Relación (4) y Diagramas de Flujos de Datos (DFD) para distintos motores de base de datos. Su principal característica, es su potente sistema de ingeniería



inversa, o sea, a partir del modelo de tablas llegar al modelo lógico. Logra identificar y estructurar BD ya existentes para poder trabajar con ellas sin problemas.

Su intuitiva interfaz gráfica hace fácil las tareas más complicadas. Mediante Case Studio se podrá realizar diagramas de flujo con muy poco tiempo y esfuerzo. Admite la generación de disparadores (triggers) para realizar validaciones en las entidades que formarán parte de las tablas de la BD.

1.13.5. Visual Paradigm.

Es una plataforma de modelado de sistema diseñado para arquitectos, desarrolladores y diseñadores de UML para acelerar el proceso de análisis y diseño de aplicaciones empresariales complejas a través del modelado que facilita la visualización de UML. 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. Presenta notación para 13 diferentes tipos de diagramas, así como los diagramas ER, diagramas Requisito de la Convención y los diagramas de análisis textual. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es un software de modelado UML que ayuda a la construcción más rápida de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Posibilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, ingeniería inversa de bases de datos. Es un software multiplataforma, está disponible para Win98; WinME; WinNT 4.x; Windows2000; WinXP; Windows2003; Mac OS; GNU/Linux.

1.14. Conclusiones.

Los Sistemas de Bases de Datos han evolucionado considerablemente desde sus inicios en 1960, mejorando cada vez más los servicios que brindan, lográndose con ellos que la información sea almacenada sin redundancia, centralizada, y de forma estructurada. Además garantizan el acceso controlado a la información.

Los Sistemas Gestores de Bases de Datos más utilizados a nivel mundial son los relacionales, pero se ha evolucionado mucho en este tema. Hoy en día existen otros modelos de datos como las Bases de Datos Orientadas a Objetos y las Objeto Relacionales, que implementan características de la orientación a



Capítulo 1: Fundamentación Teórica

objeto. Las Bases de Datos Distribuidas y Multidimensionales también se han ido expandiendo en la medida que se va desarrollando la tecnología. El uso de cada una de ellas está condicionado por las prestaciones que debe brindar la base de datos dependiendo de las necesidades para lo cual fue creada.

En el mundo existen diversas herramientas para el diseño y procesamiento de las BD, las que se van ampliando y perfeccionando con el tiempo. Escoger cuál de ellas utilizar se vuelve cada vez más difícil. Por tal motivo en este capítulo se analizan las principales clasificaciones, modelos, tendencias, gestores y herramientas de base de datos en el ámbito internacional, nacional y en la Universidad de las Ciencias Informáticas (UCI).



Capítulo 2

Descripción y análisis de la solución propuesta

2.1. Introducción.

En el presente capítulo se selecciona el modelo de base de datos, el Gestor de Base de Datos y la herramienta CASE a utilizar para el diseño de la base de datos. Se define la estrategia y la tecnología de acceso a datos a utilizar y el modelo de despliegue. Se describe la solución propuesta, haciendo énfasis en los requisitos funcionales y no funcionales y en el diagrama de clases persistentes obtenido a partir del diagrama de clases del diseño. Se realiza la descripción de todas las clases y el diseño de la base de datos, incluyendo el diagrama entidad relación y la descripción de aquellas tablas importantes. Se presenta además una pequeña fundamentación de la descripción de la arquitectura propuesta.

2.2. Estrategia de integración de la solución con otros módulos o sistemas.

La persistencia no es más que el almacenamiento de los datos en memoria, para una posterior recuperación de los mismos, por lo que para el desarrollo de las aplicaciones informáticas en la actualidad, es muy importante la implementación de una capa de persistencia que permita la comunicación entre la capa del negocio y la capa de los datos, sin que haya pérdida de información. De este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.

La Plataforma VideoWeb está estructurada en dos subsistemas, el subsistema de administración y el de gestión de contenidos. Estos tendrían una interacción de forma indirecta con la base de datos física, es decir, todas las peticiones se realizarían a través de la capa de abstracción de base de datos que incorpora el CMS Drupal –CMS que se utilizará en el desarrollo de la plataforma–, que actualmente está implementada y mantenida para MySQL y PostgreSQL. Esta capa gestionaría el acceso a las fuentes de información, permitiendo separar en una capa las funciones que tienen que ver con el acceso, modificación e inserción de los datos que se realizan en la base de datos, es decir, separando el acceso a la base de datos desde la plataforma en sí. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.



2.3. Descripción de la arquitectura y fundamentación.

La arquitectura de software crea la base fundamental en el desarrollo de una aplicación informática, esta favorece a que todos los implicados en la creación de la misma lleguen a un entendimiento común y todo se desarrolle siguiendo estándares y patrones ya definidos.

2.3.1. Modelo Vista Controlador (MVC).

Para el desarrollo de la Plataforma VideoWeb se emplea el patrón de arquitectura de software Modelo-Vista-Controlador (MVC). En este, cada una de las partes son independientes, la comunicación entre ellas es mediante interfaces, que abstraen sus estructuras internas. Esto permite desarrollar el modelo, la vista y la controladora de forma independiente, así como realizar modificaciones en sus partes, sin afectar a las demás.

La arquitectura Modelo-Vista-Controlador, divide su estructura en tres componentes diferentes:

El Modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista: enmarca la interfaz, la página HTML, el código que provee datos dinámicos a la página; aprovechar la separación de código. Las páginas web suelen contener elementos que se visualizan de igual forma durante toda la navegación: cabeceras de la página, el layout genérico, el pie de página. Generalmente sólo cambia el interior de la página, es por ello que se separará la vista en un layout y en una plantilla.

El Controlador: responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Entre las tareas comunes que realiza este componente se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, y cargar la configuración de la aplicación.

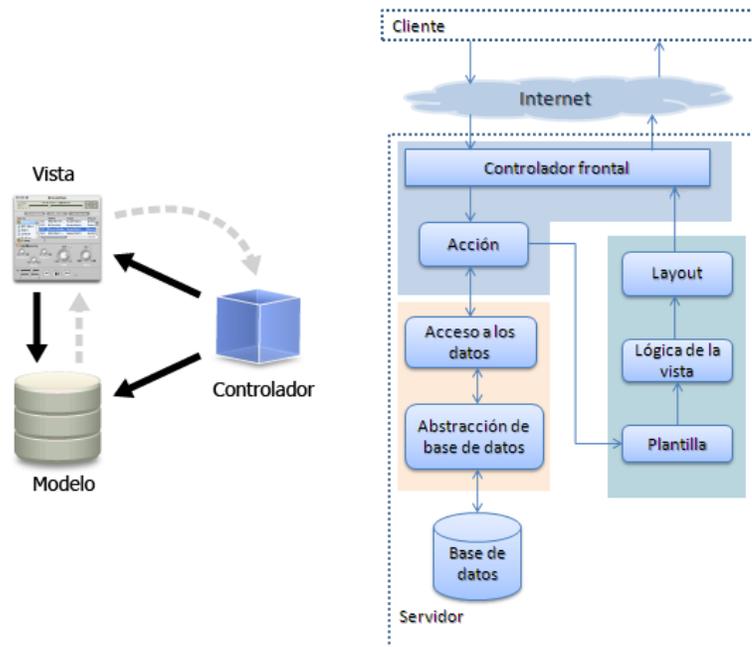


Figura 5: Proceso de relación entre los componentes en una Arquitectura MVC.

Este patrón de arquitectura de software contribuye con el trabajo en equipo, pues posibilita la especialización del personal en un componente específico, siendo solo necesario conocer las interfaces de acceso disponibles de los demás componentes. Cumple con el patrón de diseño de alta cohesión. El procesamiento es independiente de la forma en que se muestra o almacena la información.

En el modelo es donde se ubican las funcionalidades relacionadas con la manipulación de los datos, así como el Sistema Gestor de Base de Datos (SGBD). Este puede estar compuesto por uno o varios gestores de base de datos que realizan el almacenamiento de toda la información y reciben solicitudes de almacenamiento o recuperación de datos desde la capa de negocio. En la solución para la Plataforma VideoWeb el modelo está conformado solo por un SGBD que es PostgreSQL.

2.3.2. Sistema Gestor de Base de Datos (SGBD).

El Sistema Gestor de Base de Datos seleccionado para desarrollar la Plataforma VideoWeb es **PostgreSQL**, debido a que este es el servidor de bases de datos de código abierto más potente que existe en la actualidad. Es una alternativa de otros sistemas de bases de datos de código abierto como: MySQL, Firebird y MaxDB, así como de sistemas como Oracle o DB2(16). Está disponible bajo una



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Licencia BSD que permite disímiles usos, entre ellos el comercial. Es el servidor de bases de datos mayor trabajado por todos aquellos que realizan aplicaciones cliente-servidor, complejas o críticas, en el mundo Linux/Unix a nivel internacional, nacional y en la Universidad de las Ciencias Informáticas (UCI). Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre tales como Oracle. PostgreSQL es uno de los Sistemas Gestores de Bases de Datos más utilizados por la comunidad de software libre por las razones siguientes: Cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Aproxima los datos a un modelo Objeto-Relacional, y es capaz de manejar completas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones y optimización de consultas.

La UCI creó una comunidad de desarrollo de PostgreSQL, con el objetivo de garantizar soporte y mantenimiento a las aplicaciones que usen este SGBD. Esto contribuye en gran medida con el prestigio y la calidad de la solución informática ya que ante cualquier problema o necesidad de información para optimizar las funcionalidades asociadas a la base de datos, la comunidad garantiza la ayuda y apoya a los desarrolladores. Teniendo en cuenta lo anteriormente expuesto, unido a la calidad del PostgreSQL y a que la plataforma debe realizarse totalmente en software libre, se escoge para el uso en la Plataforma VideoWeb este sistema gestor de base de datos.

2.3.3. Modelo de Bases de Datos.

El modelo de base de datos utilizado es el modelo **objeto-relacional**, siendo este el que utiliza el sistema gestor de base de datos seleccionado. Con las bases de datos objeto-relacional, se pueden crear nuevos tipos de datos, que permiten gestionar aplicaciones más complejas con una gran riqueza de dominios. Tienen la posibilidad de incluir el chequeo de las reglas de integridad referencial a través de los disparadores, entre otras características. Uno de los gestores de bases de datos objeto-relacional que más se utiliza en la actualidad es PostgreSQL(4). Dándole al PostgreSQL mayor fortaleza, ya que le permite poseer interfaces nativas para lenguajes orientados a objetos como C++, PHP, PERL, PYTHON, además de traer soporte para la herencia y la seguridad de la capa de dispositivo de transportación de datos SSL (Secure Sockets Layer).



Capítulo 2: Descripción y Análisis de la Solución Propuesta

El modelo objeto-relacional es uno de los modelos más utilizados en la actualidad, junto al relacional. Las bases de datos objeto-relacionales son aquellas que desde el modelo relacional evolucionan hacia bases de datos más extensas y complejas, incorporando para obtener este fin, conceptos del modelo orientado a objetos como la herencia. Se puede afirmar que el modelo objeto-relacional contiene dos tecnologías: la tecnología relacional y la tecnología de objetos.

2.3.4. Herramienta CASE (Computer Aided Software Engineering).

Se propone como herramienta para realizar el diagrama de clases persistentes obtenido a partir del diagrama de clases del diseño el **Visual Paradigm**, herramienta CASE que ofrece un entorno de creación de diagramas para UML. El diseño es centrado en casos de uso y enfocado al negocio, lo que permite generar software de gran calidad. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la de base de datos a partir del esquema de clases. Constituye una herramienta para el modelado de UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. A partir de un modelo relacional en SQL Server, MySQL y otros, es capaz de desplegar todas las clases asociadas a las tablas. Para gestionar la persistencia y el mapeo de estas clases con la base de datos utiliza Hibernate para Java y NHibernate para la plataforma .Net. Posibilita el diseño de los modelos del negocio, requerimientos y el de la base de datos. Permite realizar todo tipo de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además facilita la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos e ingeniería inversa de bases de datos. Es un software multiplataforma, está disponible para Win98; WinME; WinNT 4.x; Windows2000; WinXP; Windows2003; Mac OS; GNU/Linux.

Para el desarrollo de la Plataforma VideoWeb es fundamental que pueda ejecutarse en GNU/Linux ya que se desea un software implementado completamente con herramientas libres. Existe una buena documentación de la herramienta lo que posibilita una familiarización con la misma en un corto plazo de tiempo. Es una herramienta colaborativa, pues soporta múltiples usuarios interactuando sobre el mismo proyecto, lo que posibilita el trabajo en equipo, además permite control de versiones.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

2.3.5. Herramienta de administración y desarrollo para PostgreSQL.

Se propone la utilización del **pgAdmin III**, siendo la aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en GNU/Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres(17). PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad(18). Posee un agente de SQL/Shell para tareas programadas y una ayuda para nuevos usuarios. Permite la creación de roles, procedimientos almacenados, transacciones preparadas y registrar un servidor sin establecer conexión.

2.4. Análisis de optimización de consultas.

Las consultas son solicitudes que se realizan a la BD, para almacenar, visualizar y actualizar los datos. Una consulta puede expresarse de varias maneras, donde cada una propone una forma diferente de hallar el resultado, de ahí que unas sean más óptimas que otras.(19)

Durante la ejecución de las consultas se pasa por 4 fases:(20)

- **Parser:** Chequea la sintaxis de la consulta y genera el árbol de consulta (Query tree).
- **Rewriter:** Procesa el árbol de consulta y comprueba la existencia de reglas reescribiendo de nuevo la consulta si es necesario.
- **Planer:** Genera el plan de consulta a partir del árbol de consulta analizando todos los posibles caminos y escogiendo el de menor coste.
- **Executor:** Recorre recursivamente el árbol del plan de consulta y accede al disco para obtener los datos necesarios. Realiza las uniones entre tablas, cálculos y ordenamientos necesarios para



Capítulo 2: Descripción y Análisis de la Solución Propuesta

devolver las tuplas resultado de las consultas.

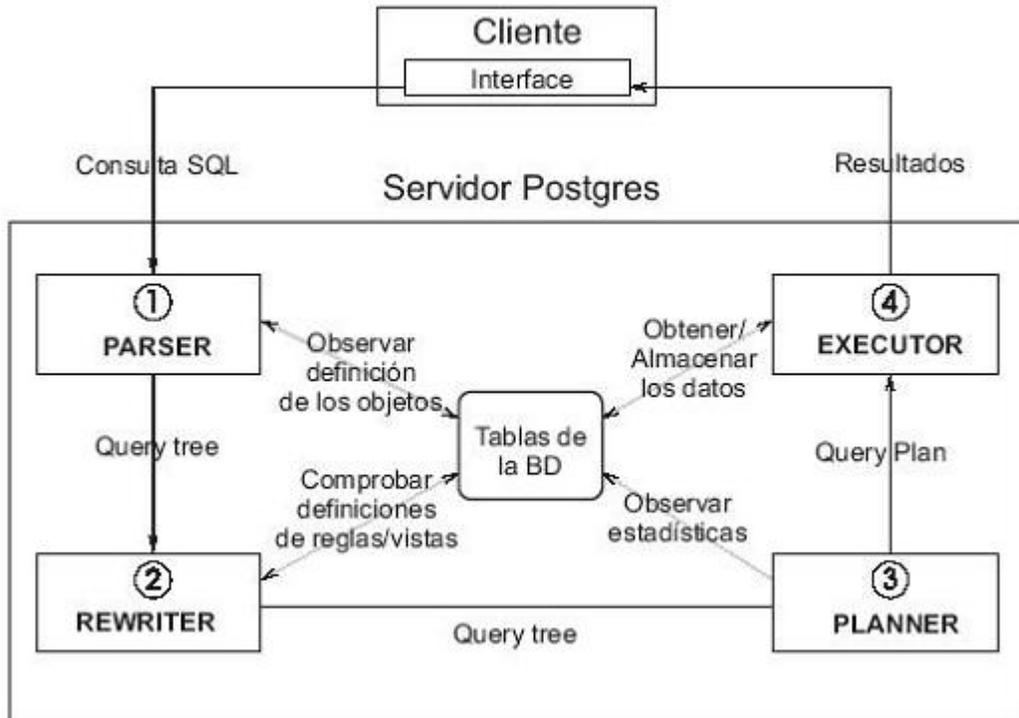


Figura 6: Diagrama de ejecución de una consulta en PostgreSQL.

Al conjunto de técnicas, algoritmos y reglas que le permiten, al motor de base de datos, elegir una alternativa entre varias sentencias con la cual pueda rescatar los datos de la manera óptima, se le conoce como proceso optimización de consultas. El problema que se plantea entonces es encontrar la manera más adecuada de resolver la consulta, sin que el proceso de determinar este resultado exceda un tiempo razonable y un gasto de recursos adicional. A continuación se presentan algunos aspectos específicos que pueden ser útiles para mejorar el rendimiento de una consulta:

- Es preferible no usar la cláusula **HAVING** si la condición deseada se puede expresar en la cláusula **WHERE**.
- Se creen índices en tantos campos como sea posible. El índice permite, dado un valor de cada uno de los atributos del conjunto, acceder todos los registros de la tabla sobre la cual se construye el índice que tienen esos valores en los atributos de la clave del índice.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

- Es importante conocer las particularidades del manejador, esto se refiere a conocer como han sido implementadas las rutinas de procesamiento de las consultas. Por ejemplo, es importante saber, al existir una condición disyuntiva (con **OR**) en la consulta si se usan o no los índices existentes.
- Las condiciones de **JOIN** se pueden evaluar más eficientemente contra un índice primario. En general, en términos de rendimiento, es preferible evaluar una condición de igualdad numérica que una condición de igualdad sobre cadenas (strings) de caracteres.
- La cláusula **DISTINCT** es costosa de ejecutar porque generalmente involucra un ordenamiento de las tuplas resultantes para eliminar duplicados. Es necesario analizar si realmente hace falta usar esa cláusula.

La utilización para el desarrollo de la plataforma el patrón de arquitectura de software Modelo-Vista-Controlador (MVC), posibilita que la capa del modelo se pueda dividir en la capa de acceso a los datos y en la capa de abstracción de la BD –la que incorpora el Drupal–. De esta forma, las funciones que acceden a los datos, no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.

2.5. Requisitos funcionales y no funcionales del sistema propuesto.

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 hay que hacer, lo que constituye la clave del éxito en la producción de un software. Aquí radica la importancia que en los últimos años se le ha dado a la identificación de los requerimientos, como parte del proceso de desarrollo del software.

La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes. El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar. La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos.(11)



Capítulo 2: Descripción y Análisis de la Solución Propuesta

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 que propiedades o cualidades se relacionen y los segundos son propiedades o cualidades que el producto debe cumplir, pensándose en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

2.5.1. Requisitos funcionales (RF).

El sistema debe ser capaz de:

RF1. Gestionar usuarios: El sistema debe permitir gestionar los datos referentes a los usuarios.

RF1.1. Adicionar usuario: El sistema debe permitir crear nuevos usuarios.

RF1.2. Modificar información de usuario: El sistema debe permitir modificar la información de los usuarios.

RF1.3. Modificar rol de usuario: El sistema debe permitir modificar los roles de cada uno de los usuarios.

RF1.4. Eliminar usuario: El sistema debe permitir eliminar los usuarios.

RF2. Autenticar usuario: El sistema debe permitir que los usuarios se autenticquen, una vez chequeado el rol que poseen, se les dará su acceso al sistema.

RF3. Gestionar secciones: El sistema debe permitir a los usuarios con permisos gestionar las secciones.

RF3.1. Adicionar sección: El sistema debe permitir a los usuarios con permisos crear una nueva sección de contenido.

RF3.2. Modificar sección: El sistema debe permitir a los usuarios con permisos modificar los datos de una sección de contenido.

RF3.3. Eliminar sección: El sistema debe permitir a los usuarios con permisos eliminar una sección de contenido.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RF4. Gestionar categorías: El sistema debe permitir a los usuarios con permisos gestionar las categorías.

RF4.1. Adicionar categoría: El sistema debe permitir a los usuarios con permisos crear una nueva categoría de contenido dentro de una sección existente.

RF4.2. Modificar categoría: El sistema debe permitir a los usuarios con permisos modificar los datos de una categoría de contenido.

RF4.3. Eliminar categoría: El sistema debe permitir a los usuarios con permisos eliminar una categoría de contenido.

RF5. Gestionar tipologías de archivos multimedia: El sistema debe permitir a los usuarios con permisos gestionar las tipologías de archivos multimedia.

RF5.1. Adicionar tipología de archivo multimedia: El sistema debe permitir a los usuarios con permisos crear una nueva tipología de archivo multimedia y definir los datos que va a incluir dicha tipología.

RF5.2. Modificar tipología de archivo multimedia media: El sistema debe permitir a los usuarios con permisos modificar los datos de una tipología de archivo multimedia.

RF5.3. Eliminar tipología de archivo multimedia: El sistema debe permitir a los usuarios con permisos eliminar una tipología de archivo multimedia.

RF6. Gestionar archivos multimedia: El sistema debe permitir a los usuarios con permisos gestionar archivos multimedia y los datos asociados a estos.

RF6.1. Adicionar archivos multimedia: El sistema debe permitir a los usuarios con permisos almacenar los archivos multimedia en el servidor de medias y crear las referencias pertinentes en la base de datos con todos los datos asociados a estos.

RF6.2. Modificar archivos multimedia: El sistema debe permitir a los usuarios con permisos modificar los datos asociados a los archivos multimedia.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RF6.3. Eliminar archivos multimedia: El sistema debe permitir a los usuarios con permisos eliminar archivos multimedia con los datos asociados a estos.

RF7. Gestionar publicaciones multimedia: El sistema debe permitir a los usuarios con permisos gestionar publicaciones multimedia.

RF7.1. Crear publicación multimedia: El sistema debe permitir a los usuarios con permisos crear publicaciones multimedia a partir de los archivos multimedia previamente almacenados en el servidor de medias, permitiendo predefinir el tiempo de publicación acotado por un inicio y fin.

RF7.2. Modificar publicación multimedia: El sistema debe permitir a los usuarios con permisos modificar los datos de la publicación multimedia.

RF7.3. Eliminar publicación multimedia: El sistema debe permitir a los usuarios con permisos eliminar publicaciones multimedia.

RF8. Publicar archivo multimedia: El sistema debe permitir a los usuarios con permisos activar las publicaciones multimedia previamente elaboradas, copiando el archivo multimedia asociado a la publicación al servidor de streaming.

RF9. Dejar de publicar archivo multimedia: El sistema debe permitir a los usuarios con permisos desactivar las publicaciones multimedia previamente publicadas, eliminando el archivo multimedia asociado a la publicación del servidor de streaming.

RF10. Reproducir archivos multimedia: El sistema debe ser capaz de reproducir los archivos multimedia, previamente publicados, especificados por el usuario.

RF10.1. Reproducir archivos multimedia desde un reproductor externo: El sistema debe ser capaz de reproducir los archivos multimedia, previamente publicados, especificados por el usuario desde un reproductor externo instalado en el sistema.

RF10.2. Reproducir archivos multimedia desde un reproductor embebido: El sistema debe ser capaz de reproducir los archivos multimedia, previamente publicados, especificados por el usuario mediante un reproductor embebido en el navegador.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RF11. Gestionar artículos de contenido: El sistema debe permitir gestionar artículos de contenido.

RF11.1. Adicionar artículo de contenido: El sistema debe permitir a los usuarios con permisos adicionar artículos de contenido permitiendo predefinir el tiempo de publicación de los mismos, acotado por un inicio y fin, aunque no estarán publicados hasta que un usuario con permisos de publicación los publique.

RF11.2. Modificar artículo de contenido: El sistema debe permitir a los usuarios con permisos modificar artículos de contenido.

RF11.3. Eliminar artículo de contenido: El sistema debe permitir a los usuarios con permisos eliminar artículos de contenido.

RF12. Publicar artículos de contenido: El sistema debe permitir a los usuarios con permisos publicar los artículos de contenido previamente adicionados.

RF13. Dejar de publicar artículo de contenido: El sistema debe permitir a los usuarios con permisos dejar de publicar los artículos de contenido previamente publicados.

RF14. Visualizar artículos de contenido: El sistema debe permitir a los usuarios visualizar artículos de contenido previamente publicados.

RF15. Gestionar lista de reproducción: El sistema debe permitir gestionar listas de reproducción.

RF15.1. Crear lista de reproducción: El sistema debe permitir a los usuarios crear listas de reproducción.

RF15.2. Reproducir lista de reproducción: El sistema debe permitir a los usuarios reproducir sus listas de reproducción.

RF15.3. Modificar lista de reproducción: El sistema debe permitir a los usuarios modificar sus listas de reproducción.

RF15.4. Eliminar lista de reproducción: El sistema debe permitir a los usuarios eliminar sus listas de reproducción.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RF16. Realizar búsquedas: El sistema debe permitir realizar búsquedas de los contenidos que posee la plataforma.

RF16.1. Realizar búsqueda de artículos de contenido: El sistema debe permitir realizar búsquedas de los artículos de contenido publicados en la plataforma.

RF16.2. Realizar búsqueda de archivos multimedia: El sistema debe permitir realizar búsquedas de los archivos multimedia almacenados en el servidor de medias.

RF16.3. Realizar búsqueda de los usuarios registrados: El sistema debe permitir a los usuarios con permisos realizar búsquedas de los usuarios registrados.

RF16.4. Realizar búsqueda de publicaciones de archivos multimedia: El sistema debe permitir realizar búsquedas de publicaciones de archivos multimedia.

RF17. Gestionar cuenta de usuario: El sistema debe permitir a los invitados gestionar una cuenta de usuario.

RF17.1. Crear cuenta de usuario: El sistema debe permitir a los invitados crearse una cuenta de usuario.

RF17.2. Modificar cuenta de usuario: El sistema debe permitirle al usuario modificar su cuenta.

RF17.3. Eliminar cuenta de usuario: El sistema debe permitirle al usuario eliminar su cuenta.

RF18. Gestionar video en vivo: El sistema debe permitir gestionar una señal de video que se esté transmitiendo en vivo.

RF18.1. Publicar señal de video en vivo: El sistema debe permitir a los usuarios con permisos publicar una señal de video en vivo.

RF18.2. Modificar señal de video en vivo: El sistema debe permitir a los usuarios con permisos modificar una señal de video en vivo previamente publicada.

RF18.3. Eliminar señal de video en vivo: El sistema debe permitir a los usuarios con permisos eliminar una señal de video en vivo previamente publicada.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RF19. Gestionar audio en vivo: El sistema debe permitir gestionar una señal de audio que se esté transmitiendo en vivo.

RF19.1. Publicar señal de audio en vivo: El sistema debe permitir a los usuarios con permisos publicar una señal de audio en vivo.

R19.2. Modificar señal de audio en vivo: El sistema debe permitir a los usuarios con permisos modificar una señal de audio en vivo previamente publicada.

RF19.3. Eliminar señal de audio en vivo: El sistema debe permitir a los usuarios con permisos eliminar una señal de audio en vivo previamente publicada.

RF20. Gestionar Roles: El sistema debe permitir a los usuarios con permisos gestionar los roles.

RF20.1. Crear Roles: El sistema debe permitir a los usuarios con permisos crear un nuevo rol.

RF20.2. Modificar Roles: El sistema debe permitir a los usuarios con permisos modificar un rol.

RF20.3. Eliminar Roles: El sistema debe permitir a los usuarios con permisos eliminar un rol.

RF21. Gestionar Interfaz: El sistema debe permitir a los usuarios con permisos gestionar interfaces.

RF21.1. Crear Interfaz: El sistema debe permitir a los usuarios con permisos crear una nueva interfaz.

RF21.2. Modificar Interfaz: El sistema debe permitir a los usuarios con permisos modificar una interfaz.

RF21.3. Eliminar Interfaz: El sistema debe permitir a los usuarios con permisos eliminar una interfaz.

RF22. Cambiar interfaz visual: El sistema debe permitir a los usuarios cambiar la interfaz visual de la plataforma.

2.5.2. Requisitos no funcionales (RNF).

RNF de Software.

- Se debe utilizar como servidor web el “Apache”.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

- Se debe utilizar como servidor Streaming el “Darwin Streaming Server”.
- Se debe utilizar para captura y transmisión en vivo el “Mp4live”.
- Se debe utilizar como sistema operativo en los servidores el “Nova-Linux”.
- La Base de Datos será implementada en el Gestor de Bases de datos PostgreSQL.
- Para la reproducción de los archivos multimedia se debe utilizar el reproductor VLC, como reproductor externo y como plugin embebido en el navegador, por ser multiplataforma y soportar tanto el protocolo de transmisión como el formato de fichero empleado.

RNF de Hardware.

- Las PCs Clientes deben poseer una tarjeta de red, espacio en disco duro superior o igual a 1.0 GB, con procesador Intel Pentium III a 1 GHz o superior y 128 MB mínimo de memoria RAM.
- Servidor de almacenamiento con recursos de hardware en dependencia de la cantidad de archivos a almacenar (Ver Tabla 1).

Cantidad de ficheros (1 hora promedio)	CPU	RAM	Almacenamiento	Tarjeta de Red
0 - 1000	Intel Dual-Core (3.0GHz 1333MHz, 6MB L2).	512 MB	300 GB - SCSI	Gigabit Ethernet
1000 - 4000	Intel Core 2 Duo Processor E8400 (6M Cache, 3.00 GHz, 1333 MHz FSB).	1 GB	1 TB - SCSI	Gigabit Ethernet
4000 - 10000	Intel Core 2 Duo Processor E8400 (6M Cache, 3.00 GHz, 1333 MHz FSB).	2 GB	3 TB – SCSI	Gigabit Ethernet

Tabla 1: Recursos de hardware para servidor de almacenamiento.

- Servidor de aplicaciones con recursos de hardware en dependencia de la cantidad de conexiones simultáneas (Ver Tabla 2).

Nº de conexiones	CPU	RAM	Almacenamiento	Tarjeta de Red
0 - 500	Intel Dual-Core Xeon 3085 (L2 - 4 MB, 3.00 GHz, 1333 MHz).	2 GB – DDR2	500 GB SCSI	Gigabit Ethernet
+ 500	Intel Quad-Core Xeon E5450 (3 GHz, 12 MB, 1333 MHz).	4 GB - DDR3	1 TB SCSI	Gigabit Ethernet

Tabla 2: Recursos de hardware para servidor de aplicaciones.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

- Al menos una estación de captura por cada canal de Transmisión en Vivo con un disco duro SCSI de 250 GB, 2 GB de memoria RAM, un microprocesador Intel Dual-Core Xeon 3085 (L2 - 4 MB, 3.00 GHz, 1333 MHz) y una tarjeta de red Gigabit Ethernet.
- Al menos dos tarjetas de captura HAUPPAGE WINTV-PVR 350 PCI WATCH RECORD TV.
- Al menos una Cámara de video Sony HDR-FX1000E con sus accesorios.

RNF Restricciones en el Diseño y la Implementación.

- Diseño e implementación de una arquitectura flexible, que permita la fácil integración o desintegración de componentes.
- La arquitectura debe soportar migrar la interfaz de usuario sin impactos considerables en re-implementación.
- El patrón arquitectónico que se debe emplear en el desarrollo es el modelo-vista-controlador.
- El lenguaje de programación que se debe utilizar es PHP.
- Los protocolos para la comunicación que se deben usar son HTTP entre el cliente y el servidor web y RTSP entre el cliente y el servidor streaming.
- Se trabajará con la herramienta CASE Visual Paradigm para el modelado de los artefactos que se generen en cada uno de los flujos de trabajo.
- Se utiliza UML como lenguaje de modelado.
- Se usa como Gestor de Base de Datos PostgreSQL.

RNF de Apariencia o Interfaz Externa.

- Interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario.

RNF de Seguridad.

RNF de Confidencialidad.

- Los datos no pueden viajar de forma transparente por la red, deben ser encriptados.
- La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RNF de Integridad.

- La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción.
- La información sólo podrá ser modificada (incluyendo su creación y borrado) por el personal autorizado.
- Se implementarán políticas de resguardo de información, así como la realización de copias periódicas de seguridad.
- Se debe mantener la integridad de la información, es decir que no se pierda durante su almacenamiento o transporte.

RNF de Disponibilidad.

- El sistema debe de estar disponible las 24 horas del día.
- Los usuarios autorizados tendrán acceso a la información en todo momento.

RNF de Usabilidad.

- El módulo de administración debe caracterizarse por un alto grado de flexibilidad.
- Disponibilidad de visualizar los archivos multimedia en un reproductor.
- Mostrar la información de forma lógica y correctamente estructurada.
- El servidor streaming debe mantener buenas prestaciones (demanda de archivos).

RNF de Soporte.

- El soporte y/o mantenimiento del sitio no debe detener el servicio.
- La capa de acceso a datos, debe soportar una migración del Gestor de Bases de Datos en proyecciones futuras.
- Se requiere que esté instalado un Gestor de Base de Datos que soporte grandes volúmenes de datos y velocidad de procesamiento.

RNF de Rendimiento.

- El sistema debe responder en un tiempo relativamente rápido a las peticiones del usuario (menos de 5 segundos).



Capítulo 2: Descripción y Análisis de la Solución Propuesta

RNF de Portabilidad.

- El sistema podrá ser usado en los sistemas operativos Windows, Mac OS y GNU/Linux, siendo accesible mediante un navegador.

RNF de Confiabilidad.

- La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación.
- El sistema debe prevenir posibles fallos y/o errores y presentar facilidades para una rápida recuperación en dichos casos.

2.6. Diagrama de clases persistentes.

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Las clases persistentes son aquellas cuyos objetos deben ser almacenados en algún repositorio como una base de datos. Estos objetos deben contener sólo los datos de las entidades que se modelan y sus relaciones con otros objetos persistentes. Lo contrario a esto son las clases temporales que son manejadas y almacenadas por el sistema en tiempo de ejecución, por lo que dejan de existir cuando termina el programa.

Las clases persistentes, por lo general, son originadas por las clases clasificadas como entidad, porque ellas modelan la información y el comportamiento asociado de algún fenómeno o concepto, ya sea una persona, un objeto del mundo real o un suceso. Todas las clases identificadas en el dominio del análisis no son persistentes.

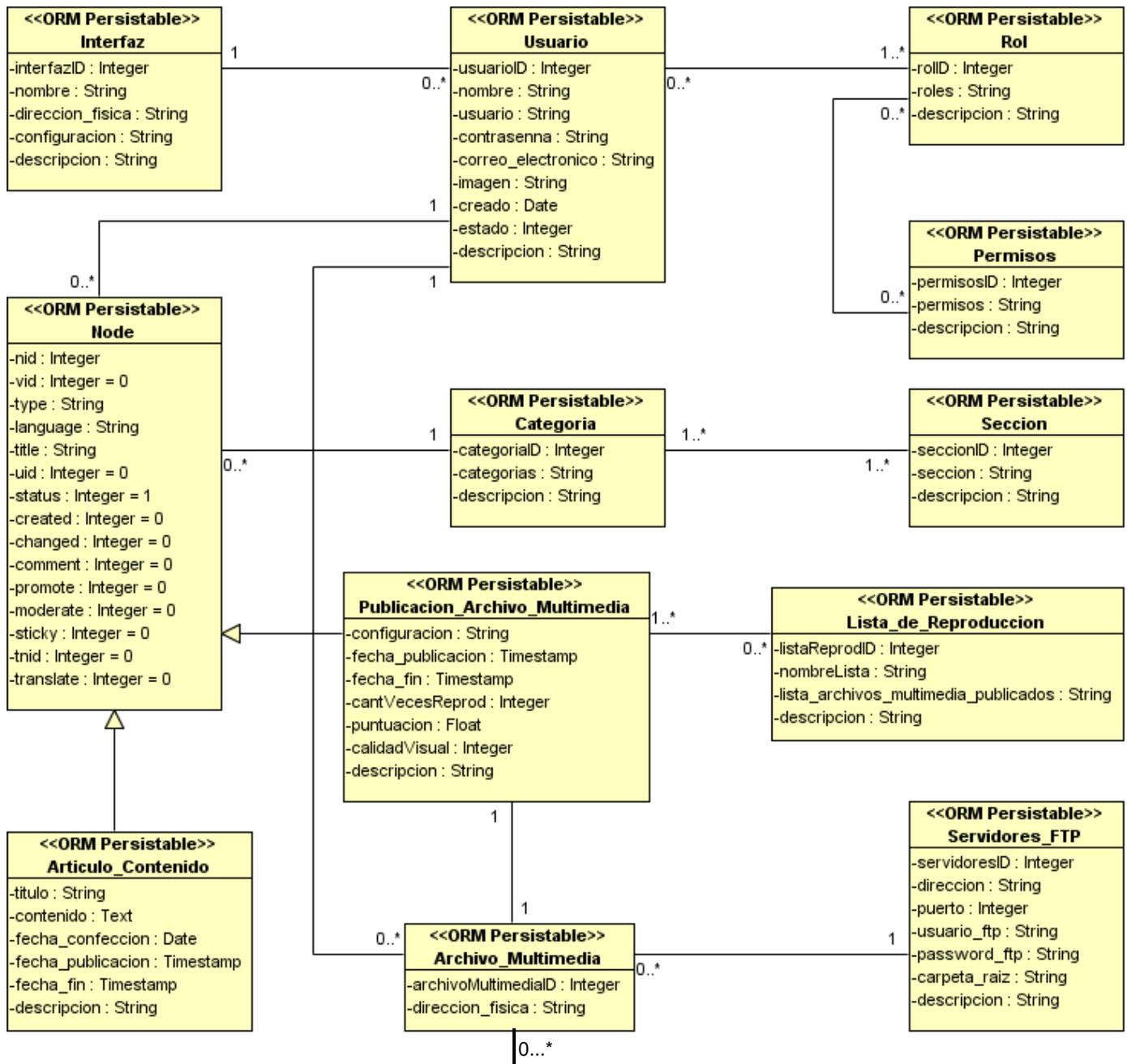
El diagrama de clases persistentes se utiliza para modelar la estructura lógica de la BD, donde las clases son representadas por tablas y los atributos de clase por columnas. Las clases persistentes y sus atributos hacen referencia directamente a las entidades lógicas y a sus atributos.

A continuación se muestra el diagrama de clases persistentes de la Plataforma VideoWeb obtenido a partir del diagrama de clases del diseño. En él se representan las clases persistentes en su totalidad, así como las relaciones existentes entre ellas y los atributos que las componen.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

DIAGRAMA DE CLASES PERSISTENTES



El diagrama de clases persistentes continúa en la próxima página. Ambas partes se relacionan a través de una relación de 1 a 0..* entre las clases Archivo_Multimedia y Tipología_Archivo_Multimedia respectivamente.



Nombre: Interfaz	
Tipo de clase: Entidad	
Atributo	Tipo
interfazID	Integer
nombre	String
direccion_fisica	String
configuracion	String
descripcion	String

Tabla 3: Descripción de la clase Interfaz.

Nombre: Usuario	
Tipo de clase: Entidad	
Atributo	Tipo
usuarioid	Integer
nombre	String
usuario	String
contrasenna	String
correo_electronico	String
imagen	String
creado	Date
estado	Integer
descripcion	String

Tabla 4: Descripción de la clase Usuario.

Nombre: Rol	
Tipo de clase: Entidad	
Atributo	Tipo
rolID	Integer
roles	String



Capítulo 2: Descripción y Análisis de la Solución Propuesta

descripcion	String
-------------	--------

Tabla 5: Descripción de la clase Rol.

Nombre: Permisos	
Tipo de clase: Entidad	
Atributo	Tipo
permisosID	Integer
permisos	String
descripcion	String

Tabla 6: Descripción de la clase Permisos.

Nombre: Node	
Tipo de clase: Entidad	
Atributo	Tipo
nid	Integer
vid	Integer = 0
type	String
language	String
title	String
uid	Integer = 0
status	Integer = 1
created	Integer = 0
changed	Integer = 0
comment	Integer = 0
promote	Integer = 0
moderate	Integer = 0
sticky	Integer = 0
translate	Integer = 0

Tabla 7: Descripción de la clase Node.

Nombre: Categoria



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Tipo de clase: Entidad	
Atributo	Tipo
categorialD	Integer
categorias	String
descripcion	String

Tabla 8: Descripción de la clase Categoría.

Nombre: Seccion	
Tipo de clase: Entidad	
Atributo	Tipo
seccionID	Integer
seccion	String
descripcion	String

Tabla 9: Descripción de la clase Seccion.

Nombre: Articulo_Contenido	
Tipo de clase: Entidad	
Atributo	Tipo
titulo	String
contenido	Text
fecha_confeccion	Date
fecha_publicacion	Timestamp
fecha_fin	Timestamp
descripcion	String

Tabla 10: Descripción de la clase Articulo_Contenido.

Nombre: Publicacion_Archivo_Multimedia	
Tipo de clase: Entidad	
Atributo	Tipo
configuracion	String
fecha_publicacion	Timestamp



Capítulo 2: Descripción y Análisis de la Solución Propuesta

fecha_fin	Timestamp
cantVecesReprod	Integer
puntuacion	Float
calidadVisual	Integer
descripcion	String

Tabla 11: Descripción de la clase Publicacion_Archivo_Multimedia.

Nombre: Lista_de_Reproduccion	
Tipo de clase: Entidad	
Atributo	Tipo
listaReprodID	Integer
nombreLista	String
lista_archivos_multimedia_publicados	String
descripcion	String

Tabla 12: Descripción de la clase Lista_de_Reproduccion.

Nombre: Archivo_Multimedia	
Tipo de clase: Entidad	
Atributo	Tipo
archivoMultimediaID	Integer
direccion_fisica	String

Tabla 13: Descripción de la clase Archivo_Multimedia.

Nombre: Servidores_FTP	
Tipo de clase: Entidad	
Atributo	Tipo
servidoresID	Integer
direccion	String
puerto	Integer
usuario_ftp	String
password_ftp	String



Capítulo 2: Descripción y Análisis de la Solución Propuesta

carpeta_raiz	String
descripcion	String

Tabla 14: Descripción de la clase Servidores_FTP.

Nombre: Tipologia_Archivo_Multimedia	
Tipo de clase: Entidad	
Atributo	Tipo
tipologiaID	Integer
titulo_original	String
anno	Integer
duracion	Integer
descripcion	String

Tabla 15: Descripción de la clase Tipologia_Archivo_Multimedia.

Nombre: Imagen	
Tipo de clase: Entidad	
Atributo	Tipo
imagenID	Integer
nombre	String
dirección_fisica	String
descripcion	String

Tabla 16: Descripción de la clase Imagen.

Nombre: Pelicula	
Tipo de clase: Entidad	
Atributo	Tipo
titulo_espannol	String
director	String
pais	String
genero	String
reparto	String



Capítulo 2: Descripción y Análisis de la Solución Propuesta

guionista	String
productor	String
musica	String
fotografía	String
sinopsis	String

Tabla 17: Descripción de la clase Pelicula.

Nombre: Serie	
Tipo de clase: Entidad	
Atributo	Tipo
titulo_espannol	String
creador	String
pais	String
genero	String
reparto	String
sinopsis	String

Tabla 18: Descripción de la clase Serie.

Nombre: Temporada	
Tipo de clase: Entidad	
Atributo	Tipo
numero	Integer
descripcion	String

Tabla 19: Descripción de la clase Temporada.

Nombre: Capitulo	
Tipo de clase: Entidad	
Atributo	Tipo
numero	Integer
nombre	String
descripcion	String

Tabla 20: Descripción de la clase Capitulo.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Nombre: VideoMusical	
Tipo de clase: Entidad	
Atributo	Tipo
interprete	String
album	String
genero	String
discografica	String
director	String

Tabla 21: Descripción de la clase VideoMusical.

Nombre: Documental	
Tipo de clase: Entidad	
Atributo	Tipo
titulo_espannol	String
pais	String
tematica	String
sinopsis	String

Tabla 22: Descripción de la clase Documental.

Nombre: Musica	
Tipo de clase: Entidad	
Atributo	Tipo
interprete	String
album	String
genero	String
discografica	String

Tabla 23: Descripción de la clase Musica.

Nombre: AudioVisualRealizado	
Tipo de clase: Entidad	
Atributo	Tipo



Capítulo 2: Descripción y Análisis de la Solución Propuesta

fecha_produccion	Date
fecha_edicion	Date
editor	String
realizador	String
camarografo	String
digitalizador	String
sonidista	String

Tabla 24: Descripción de la clase AudioVisualRealizado.

Nombre: AudioVisualVariado	
Tipo de clase: Entidad	
Atributo	Tipo
causa	String
lugar	String
personalidades	String

Tabla 25: Descripción de la clase AudioVisualVariado.

Nombre: Teleclase	
Tipo de clase: Entidad	
Atributo	Tipo
sumario	String
profesor	String

Tabla 26: Descripción de la clase Teleclase.

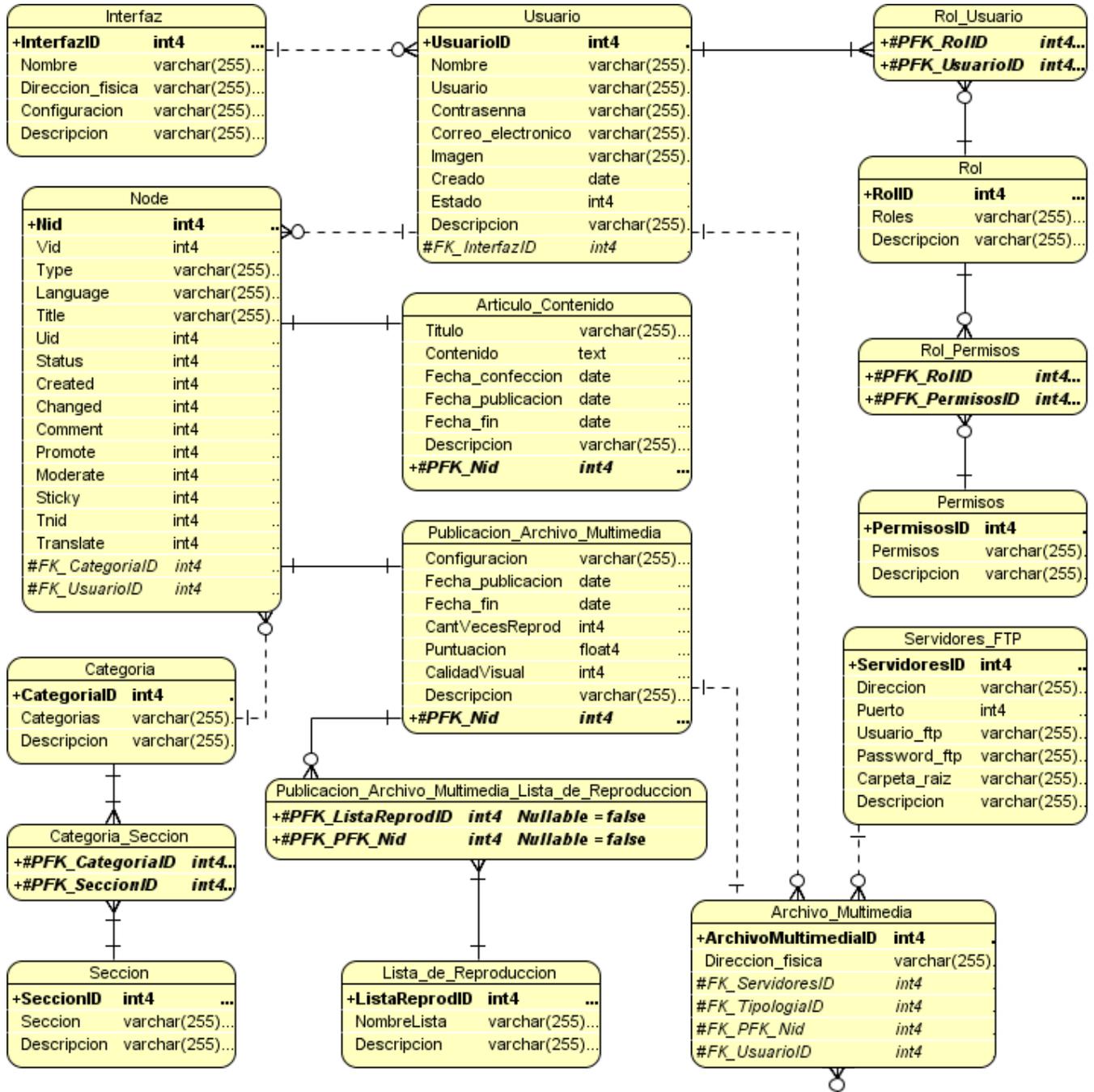
Nombre: Materia	
Tipo de clase: Entidad	
Atributo	Tipo
materialID	Integer
nombre	String
descripcion	String

Tabla 27: Descripción de la clase Materia.



2.7. Diseño de la Base de Datos.

2.7.1. Diagrama Entidad Relación de la Base de Datos.



El diagrama entidad relación continúa en la próxima página. Ambas partes se relacionan a través de una



Capítulo 2: Descripción y Análisis de la Solución Propuesta

relación de 1 a 0...* entre las entidades Archivo_Multimedia y Tipología_Archivo_Multimedia respectivamente.

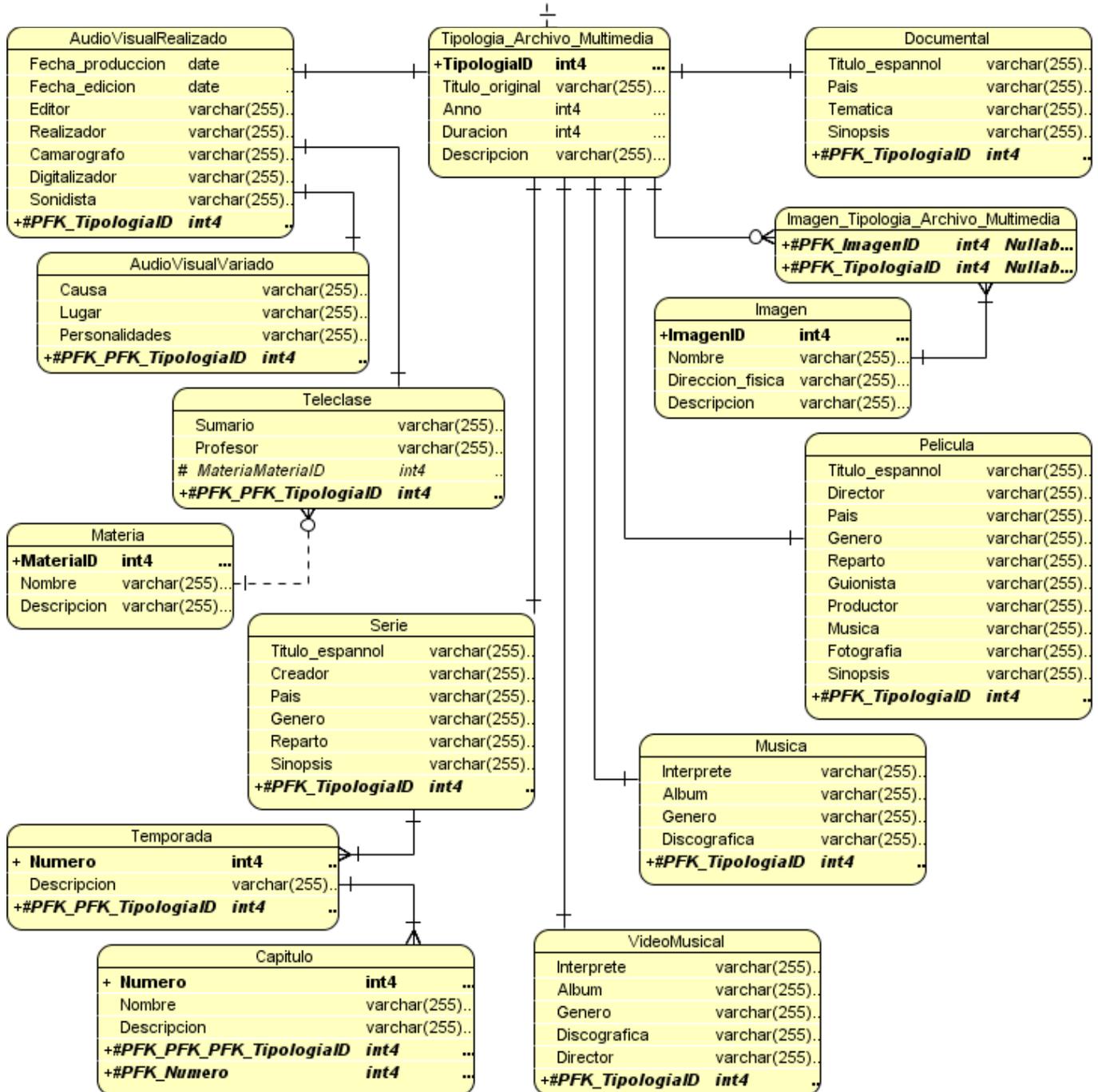


Figura 8: Diagrama Entidad Relación de la Plataforma VideoWeb.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

2.7.2. Descripción de las Tablas.

En esta sección se describen de forma general los datos que se almacenan en cada tabla del diagrama entidad relación de la Base de Datos de la Plataforma VideoWeb, al igual que se explica brevemente lo que representan todos sus atributos.

Nombre: Interfaz		
Descripción: En esta tabla se almacenan los datos de las interfaces que posee el sistema.		
Atributo	Tipo	Descripción
InterfazID	int4	Valor numérico único y auto incremental que identifica a cada una de las interfaces que posee el sistema.
Nombre	varchar(255)	Nombre de la interfaz.
Direccion_fisica	varchar(255)	Dirección física donde se encuentra almacenada la interfaz.
Configuracion	varchar(255)	Se almacena la configuración realizada a la interfaz.
Descripcion	varchar(255)	Breve descripción de la interfaz.

Tabla 28: Descripción de la tabla Interfaz.

Nombre: Usuario		
Descripción: En esta tabla se almacenan los datos de los usuarios que posee el sistema.		
Atributo	Tipo	Descripción
UsuarioID	int4	Valor numérico único y auto incremental que identifica a cada uno de los usuarios que posee el sistema.
Nombre	varchar(255)	Nombre y apellidos de los usuarios que forman parte del sistema.
Usuario	varchar(255)	Identificador único no llave de cada uno de los usuarios del sistema.
Contraseña	varchar(255)	Código personal de cada usuario para acceder al sistema.
Correo_electronico	varchar(255)	Dirección de correo electrónico que posee el usuario.
Imagen	varchar(255)	Vínculo a la imagen seleccionada por el usuario para utilizarla en su perfil.
Creado	date	Fecha en la que fue creado el usuario.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Estado	int4	Valor numérico entre 0 y 1 que representa el estado activo o no del usuario.
Descripcion	varchar(255)	Breve descripción del usuario.
FK_InterfazID	int4	Identificador de la interfaz seleccionada por el usuario.

Tabla 29: Descripción de la tabla Usuario.

Nombre: Rol_Usuario		
Descripción: En esta tabla se almacena la relación existente entre un usuario y un rol determinado. Esta tabla es resultado de la relación mucho a mucho existente entre usuario y rol.		
Atributo	Tipo	Descripción
PFK_RolID	int4	Valor numérico único y auto incremental que identifica a cada uno de los roles que posee el sistema.
PFK_UsuarioID	int4	Valor numérico único y auto incremental que identifica a cada uno de los usuarios que posee el sistema.

Tabla 30: Descripción de la tabla Rol_Usuario.

Nombre: Rol		
Descripción: En esta tabla se almacenan los datos de los roles que cumple determinado usuario en el sistema.		
Atributo	Tipo	Descripción
RollID	int4	Valor numérico único y auto incremental que identifica a cada uno de los roles que posee el sistema.
Roles	varchar(255)	Nombre de los roles que posee el sistema.
Descripcion	varchar(255)	Breve descripción de los roles del sistema.

Tabla 31: Descripción de la tabla Rol.

Nombre: Rol_Perminos		
Descripción: En esta tabla se almacena la relación existente entre un rol y los permisos que posee. Esta tabla es resultado de la relación mucho a mucho existente entre permisos y rol.		
Atributo	Tipo	Descripción
PFK_RolID	int4	Valor numérico único y auto incremental que identifica a cada uno de los



Capítulo 2: Descripción y Análisis de la Solución Propuesta

		roles que posee el sistema.
PFK_PerminosID	int4	Valor numérico único y auto incremental que identifica a cada uno de los permisos que posee el sistema.

Tabla 32: Descripción de la tabla Rol_Perminos.

Nombre: Perminos		
Descripción: En esta tabla se almacenan los datos de los permisos que posee determinado rol en el sistema.		
Atributo	Tipo	Descripción
PerminosID	int4	Valor numérico único y auto incremental que identifica a cada uno de los permisos que posee el sistema.
Perminos	varchar(255)	Nombre de los permisos que posee el sistema.
Descripcion	varchar(255)	Breve descripción de los permisos del sistema.

Tabla 33: Descripción de la tabla Perminos.

Nombre: Node		
Descripción: En esta tabla se almacenan los datos de generales de los contenidos, de la cual heredan todos los contenidos que posee el sistema. Esta tabla pertenece al Drupal.		
Atributo	Tipo	Descripción
Nid	int4	Valor numérico único y auto incremental que identifica a cada uno de los nodos que posee el sistema.
Vid	int4	Valor numérico auto incremental que identifica la revisión de cada uno de los contenidos.
Type	varchar(255)	Tipo de contenido.
Language	varchar(255)	El idioma del contenido.
Title	varchar(255)	Título que aparecerá.
Uid	int4	El id del autor del contenido.
Status	int4	1 publicado, 0 no se marca como publicado.
Created	int4	Cuándo fue creado el contenido.
Changed	int4	Cuándo fue modificado algún contenido creado con anterioridad.
Comment	int4	Si lleva comentarios, 0 sin comentarios, 1 con comentarios.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Promote	int4	Promocionado, 0 sin promocionar, 1 promocionado.
Moderate	int4	Moderado, 0 no moderado, 1 moderado.
Sticky	int4	Pegajoso, 0 no pegajoso, 1 pegajoso
Tnid	int4	
Translate	int4	Traducción del contenido en otros idiomas.
FK_CategorialID	int4	Valor numérico único y auto incremental que identifica a cada uno de las categorías que posee el sistema.

Tabla 34: Descripción de la tabla Node.

Nombre: Categoria		
Descripción: En esta tabla se almacenan los datos de las categorías a la que pertenece determinado node en el sistema.		
Atributo	Tipo	Descripción
CategorialID	int4	Valor numérico único y auto incremental que identifica a cada uno de las categorías que posee el sistema.
Categorias	varchar(255)	Nombre de las categorías que posee el sistema.
Descripcion	varchar(255)	Breve descripción de las categorías del sistema.

Tabla 35: Descripción de la tabla Categoria.

Nombre: Categoria_Seccion		
Descripción: En esta tabla se almacena la relación existente entre una categoría y la sección a la que pertenece. Esta tabla es resultado de la relación mucho a mucho existente entre categoría y sección.		
Atributo	Tipo	Descripción
PFK_CategorialID	int4	Valor numérico único y auto incremental que identifica a cada uno de las categorías que posee el sistema.
PFK_SeccionID	int4	Valor numérico único y auto incremental que identifica a cada uno de las secciones que posee el sistema.

Tabla 36: Descripción de la tabla Categoria_Seccion.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Nombre: Seccion		
Descripción: En esta tabla se almacenan los datos de las secciones que posee el sistema.		
Atributo	Tipo	Descripción
SeccionID	int4	Valor numérico único y auto incremental que identifica a cada uno de las secciones que posee el sistema.
Seccion	varchar(255)	Nombre de las secciones que posee el sistema.
Descripcion	varchar(255)	Breve descripción de las secciones del sistema.

Tabla 37: Descripción de la tabla Seccion.

Nombre: Artículo_Contenido		
Descripción: En esta tabla se almacenan los datos de los artículos de contenido que posee el sistema.		
Atributo	Tipo	Descripción
Titulo	varchar(255)	Título del artículo de contenido.
Contenido	text	Contenido del cuerpo del artículo.
Fecha_confeccion	date	Fecha en la que fue confeccionado el artículo.
Fecha_publicacion	date	Fecha en la que fue publicado el artículo.
Fecha_inicio	date	Fecha en la que dejará de estar publicado el artículo de contenido.
Descripcion	varchar(255)	Breve descripción de los artículos de contenidos del sistema.
PFK_Nid	int4	Valor numérico único y auto incremental que identifica a cada uno de los nodos que posee el sistema.

Tabla 38: Descripción de la tabla Artículo_Contenido.

Nombre: Publicacion_Archivo_Multimedia		
Descripción: En esta tabla se almacenan los datos de las publicaciones de archivos multimedia que posee el sistema.		
Atributo	Tipo	Descripción
Configuracion	varchar(255)	Configuración que posee la publicación para ser mostrada.
Fecha_publicacion	date	Fecha en la que fue publicado el archivo multimedia.
Fecha_fin	date	Fecha en la que dejará de estar publicado el archivo multimedia.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Puntuacion	float4	Puntuación recibida por los usuarios.
CalidadVisual	Integer	Valor número entre 1 y 5 equivalente a la calidad que posee el archivo publicado.
Descripcion	varchar(255)	Breve descripción de las publicaciones de archivos multimedia del sistema.
PFK_Nid	int4	Valor numérico único y auto incremental que identifica a cada uno de los nodos que posee el sistema.

Tabla 39: Descripción de la tabla `Publicacion_Archivo_Multimedia`.

Nombre: <code>Publicacion_Archivo_Multimedia_Lista_de_Reproduccion</code>		
Descripción: En esta tabla se almacena la relación existente entre una lista de reproducción y los archivos multimedia publicados en el sistema. Esta tabla es resultado de la relación mucho a mucho existente entre <code>lista_de_reproduccion</code> y <code>publicacion_archivo_multimedia</code> .		
Atributo	Tipo	Descripción
PFK_PFK_Nid	int4	Valor numérico único y auto incremental que identifica a cada uno de los nodos que posee el sistema.
PFK_ListaReprodID	int4	Valor numérico único y auto incremental que identifica a cada una de las listas de reproducción que posee el sistema.

Tabla 40: Descripción de la tabla `Publicacion_Archivo_Multimedia_Lista_de_Reproduccion`.

Nombre: <code>Lista_de_Reproduccion</code>		
Descripción: En esta tabla se almacenan los datos de las listas de reproducción posee el sistema.		
Atributo	Tipo	Descripción
ListaReprodID	int4	Valor numérico único y auto incremental que identifica a cada una de las listas de reproducción que posee el sistema.
NombreLista	varchar(255)	Nombre de la lista de reproducción.
Descripcion	varchar(255)	Breve descripción de las listas de reproducción del sistema.

Tabla 41: Descripción de la tabla `Lista_de_Reproduccion`.

Nombre: <code>Archivo_Multimedia</code>		
Descripción: En esta tabla se almacenan los datos de los archivos multimedia que posee el		



Capítulo 2: Descripción y Análisis de la Solución Propuesta

sistema.		
Atributo	Tipo	Descripción
ArchivoMultimediaID	int4	Valor numérico único y auto incremental que identifica a cada uno de los archivos multimedia que posee el sistema.
Direccion_fisica	varchar(255)	Dirección física donde se encuentra almacenado el archivo multimedia.
FK_ServidoresID	int4	Valor numérico único y auto incremental que identifica a cada uno de los servidores FTP que posee el sistema.
FK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.
FK_PFK_Nid	int4	Valor numérico único y auto incremental que identifica a cada uno de los nodos que posee el sistema.

Tabla 42: Descripción de la tabla Archivo_Multimedia.

Nombre: Servidores_FTP		
Descripción: En esta tabla se almacenan los datos de los servidores FTP que posee el sistema.		
Atributo	Tipo	Descripción
ServidoresID	int4	Valor numérico único y auto incremental que identifica a cada uno de los servidores FTP que posee el sistema.
Direccion	varchar(255)	Dirección que posee el servidor FTP.
Puerto	int4	Puerto por el que se accede al servidor FTP.
Usuario_ftp	varchar(255)	Usuario para acceder al servidor FTP.
password_ftp	varchar(255)	Contraseña del usuario para acceder al servidor FTP.
Carpeta_raiz	varchar(255)	Carpeta raíz en la que se encuentran almacenados los archivos.
Descripcion	varchar(255)	Breve descripción de los servidores FTP del sistema.

Tabla 43: Descripción de la tabla Servidores_FTP.

Nombre: Tipologia_Archivo_Multimedia		
Descripción: En esta tabla se almacenan los datos de las tipologías de archivo multimedia que posee el sistema.		



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Atributo	Tipo	Descripción
TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.
Titulo_original	varchar(255)	Título original del archivo multimedia.
Anno	int4	Año en que fue creado el archivo multimedia.
Duracion	int4	Duración en minutos del archivo multimedia.
Descripcion	varchar(255)	Breve descripción de los archivos multimedia del sistema.

Tabla 44: Descripción de la tabla Tipologia_Archivo_Multimedia.

Nombre: Imagen_Tipologia_Archivo_Multimedia		
Descripción: En esta tabla se almacena la relación existente entre una tipología de archivo multimedia y las imágenes asociadas a la misma. Esta tabla es resultado de la relación mucho a mucho existente entre tipologia_archivo_multimedia e imagen.		
Atributo	Tipo	Descripción
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.
PFK_ImagenID	int4	Valor numérico único y auto incremental que identifica a cada una de las imágenes que posee una tipología de archivo multimedia.

Tabla 45: Descripción de la tabla Imagen_Tipologia_Archivo_Multimedia.

Nombre: Imagen		
Descripción: En esta tabla se almacenan los datos de las imágenes que posee una tipología de archivo multimedia.		
Atributo	Tipo	Descripción
ImagenID	int4	Valor numérico único y auto incremental que identifica a cada una de las imágenes que posee una tipología de archivo multimedia.
Nombre	varchar(255)	Nombre de la imagen.
Direccion_fisica	varchar(255)	Dirección física donde se encuentra almacenada la imagen.
Descripcion	varchar(255)	Breve descripción de la imagen.

Tabla 46: Descripción de la tabla Imagen.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Nombre: Pelicula		
Descripción: En esta tabla se almacenan los datos de las películas que posee el sistema.		
Atributo	Tipo	Descripción
Titulo_espagnol	varchar(255)	Título en español de la película.
Director	varchar(255)	Nombre del director de la película.
Pais	varchar(255)	Nombre del país al que pertenece la película.
Genero	varchar(255)	Género de la película.
Reparto	varchar(255)	Nombre de los actores que interpretan los personajes de la película.
Guionista	varchar(255)	Nombre del escritor del guión de la película.
Productor	varchar(255)	Nombre del productor de la película.
Música	varchar(255)	Nombre del compositor de la música de la película.
Fotografía	varchar(255)	Nombre del encargado de la fotografía de la película.
Sinopsis	varchar(255)	Breve descripción o resumen del tema y argumento que trata la película. Donde se incluyen las características mayores de los protagonistas.
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 47: Descripción de la tabla Pelicula.

Nombre: Serie		
Descripción: En esta tabla se almacenan los datos de las series que posee el sistema.		
Atributo	Tipo	Descripción
Titulo_espagnol	varchar(255)	Título en español de la serie.
Creador	varchar(255)	Nombre del creador o creadores de la serie.
Pais	varchar(255)	Nombre del país al que pertenece la serie.
Genero	varchar(255)	Género de la serie.
Reperto	varchar(255)	Nombre de los actores que interpretan los personajes de la serie.
Sinopsis	varchar(255)	Breve descripción o resumen del tema y argumento que trata la serie. Donde se incluyen las características mayores de los



Capítulo 2: Descripción y Análisis de la Solución Propuesta

		protagonistas.
PFK_TipologíaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 48: Descripción de la tabla Serie.

Nombre: Temporada		
Descripción: En esta tabla se almacenan los datos de las temporadas que posee una serie.		
Atributo	Tipo	Descripción
numero	int4	Número de la temporada de la serie. Valor único que identifica cada temporada.
Descripcion	varchar(255)	Breve descripción de la temporada.
PFK_PFK_TipologíaID	varchar(255)	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 49: Descripción de la tabla Temporada.

Nombre: Capitulo		
Descripción: En esta tabla se almacenan los datos de los capítulos que posee una temporada.		
Atributo	Tipo	Descripción
Numero	int4	Número del capítulo de la serie. Valor único que identifica cada capítulo.
Nombre	varchar(255)	Nombre del capítulo.
Descripcion	varchar(255)	Breve descripción del capítulo.
PFK_PFK_PFK_TipologíaID	varchar(255)	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.
PFK_Numero	int4	Número de la temporada de la serie. Valor único que identifica cada temporada.

Tabla 50: Descripción de la tabla Capitulo.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Nombre: VideoMusical		
Descripción: En esta tabla se almacenan los datos de los videos musicales que posee el sistema.		
Atributo	Tipo	Descripción
Interprete	varchar(255)	Nombre del cantante o agrupación que interpreta la canción del video.
Album	varchar(255)	Nombre del álbum al que pertenece la canción del video.
Genero	varchar(255)	Género al que pertenece la canción.
Discografica	varchar(255)	Nombre de la discográfica bajo la cual se realizó el álbum al que pertenece la canción.
Director	varchar(255)	Nombre del director del video.
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 51: Descripción de la tabla VideoMusical.

Nombre: Documental		
Descripción: En esta tabla se almacenan los datos de los documentales que posee el sistema.		
Atributo	Tipo	Descripción
Titulo_espagnol	varchar(255)	Título en español del documental.
Pais	varchar(255)	Nombre del país que realizó el documental.
Tematica	varchar(255)	Tema del que trata el documental.
Sinopsis	varchar(255)	Breve descripción o resumen del tema y argumento que trata el documental.
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 52: Descripción de la tabla Documental.

Nombre: Musica		
Descripción: En esta tabla se almacenan los datos de la música que posee el sistema.		
Atributo	Tipo	Descripción
Interprete	varchar(255)	Nombre del cantante o agrupación que interpreta la canción.
Album	varchar(255)	Nombre del álbum al que pertenece la canción.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Genero	varchar(255)	Género al que pertenece la canción.
Discografica	varchar(255)	Nombre de la discográfica bajo la cual se realizó el álbum al que pertenece la canción.
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 53: Descripción de la tabla Musica.

Nombre: AudioVisualRealizado		
Descripción: En esta tabla se almacenan los datos de los audiovisuales realizados en la institución que posee el sistema.		
Atributo	Tipo	Descripción
Fecha_produccion	date	Fecha en la que se filmó o concibió el material.
Fecha_edicion	date	Fecha en la que se editó el material.
Editor	varchar(255)	Nombre del editor del material.
Realizador	varchar(255)	Nombre del realizador del material
Camarografo	varchar(255)	Nombre del camarógrafo que filmó el material.
Digitalizador	varchar(255)	Nombre del que digitalizó el material.
Sonidista	varchar(255)	Nombre del sonidista de la filmación.
PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 54: Descripción de la tabla AudioVisualRealizado.

Nombre: AudioVisualVariado		
Descripción: En esta tabla se almacenan los datos de los audiovisuales realizados en la institución que no se enmarcan en otra tipología.		
Atributo	Tipo	Descripción
Causa	varchar(255)	Causa por la cual se realizó el material.
Lugar	varchar(255)	Lugar donde se realizó el material.
Personalidades	varchar(255)	Personalidades que estuvieron presentes.
PFK_PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a



Capítulo 2: Descripción y Análisis de la Solución Propuesta

		cada una de las tipologías de archivo multimedia que posee el sistema.
--	--	--

Tabla 55: Descripción de la tabla AudioVisualVariado.

Nombre: Teleclase		
Descripción: En esta tabla se almacenan los datos de las teleclases que posee el sistema.		
Atributo	Tipo	Descripción
Sumario	varchar(255)	Sumario o tema de la teleclase.
Profesor	varchar(255)	Nombre del profesor que imparte la teleclase.
MateriaMaterialID	int4	Valor numérico único y auto incremental que identifica a cada una de las materias que posee el sistema.
PFK_PFK_TipologiaID	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.

Tabla 56: Descripción de la tabla Teleclase.

Nombre: Materia		
Descripción: En esta tabla se almacenan los datos de las materias que posee el sistema.		
Atributo	Tipo	Descripción
MaterialID	int4	Valor numérico único y auto incremental que identifica a cada una de las materias que posee el sistema.
Nombre	varchar(255)	Nombre de la materia o asignatura.
Descripcion	varchar(255)	Breve descripción de la materia o asignatura.

Tabla 57: Descripción de la tabla Materia.

2.8. Conclusiones.

La Plataforma VideoWeb está estructurada en dos subsistemas, el subsistema de administración y el de gestión de contenidos. Estos tendrían una interacción de forma indirecta con la base de datos física, es decir, todas las peticiones se realizarían a través de la capa de abstracción de base de datos que incorpora el CMS Drupal –CMS que se utilizará en el desarrollo de la plataforma–, que actualmente está implementada y mantenida para MySQL y PostgreSQL.



Capítulo 2: Descripción y Análisis de la Solución Propuesta

Para el desarrollo de la plataforma se emplea el patrón de arquitectura de software Modelo-Vista-Controlador (MVC). En este, cada una de las partes son independientes, la comunicación entre ellas es mediante interfaces, que abstraen sus estructuras internas. Lo cual permite desarrollar el modelo, la vista y la controladora de forma independiente, así como realizar modificaciones en sus partes, sin afectar a las demás.

El Sistema Gestor de Base de Datos seleccionado para desarrollar la BD es **PostgreSQL**, debido a que este es el servidor de BD de código abierto más potente que existe en la actualidad. Cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Además la UCI creó una comunidad de desarrollo de PostgreSQL, con el objetivo de garantizar soporte y mantenimiento a las aplicaciones que usen este SGBD, contribuyendo en gran medida con el prestigio y la calidad de la solución informática.

Se recoge cada uno de los requisitos funcionales y no funcionales planteados por los usuarios para que un sistema de software funcione adecuadamente. Se muestra el diagrama de clases persistentes, definiendo un total de 25 clases. Se analiza cada una de ellas, de las cuales se explica brevemente cada uno de sus atributos y su tipo. Se expone además el diagrama entidad relación para la BD, puntualizando un total de 30 entidades. Para facilitar el entendimiento de cada una de sus partes se muestra la descripción de las entidades con sus atributos, el tipo y una pequeña descripción de cada una de ellas.



Capítulo 3

Validación del diseño realizado

3.1. Introducción.

En el presente capítulo se realiza la validación teórica y funcional del diseño de la Base de Datos (BD). De lo primero se recogen aspectos como la integridad, el análisis de la redundancia de la información, la trazabilidad de las acciones, así como la normalización y seguridad de la BD. Lo segundo se basa en las pruebas realizadas a la BD para comprobar su funcionamiento a través de la generación de código de programas para un llenado voluminoso e inteligente de la BD y de pruebas de carga intensiva.

3.2. Validación teórica del diseño.

Para realizar un buen diseño de una BD hay que tener en cuenta varios aspectos importantes como: la integridad de los datos, la normalización del diseño, la redundancia de información, la trazabilidad de las acciones que se realizan y sobre todo la seguridad en la BD, la cual es la base principal de los aspectos anteriormente mencionados, permitiendo controlar el acceso a los datos y que los mismos no se corrompan como resultado de acciones no controladas. No basta solamente con realizar el diseño de una BD, se tienen que tener en cuenta estos aspectos para garantizar la consistencia, integridad y seguridad de la misma.

3.2.1. Integridad.

La integridad de los datos se refiere a la corrección y completitud de los datos en una BD. Al modificar el contenido de la BD con sentencias **INSERT**, **DELETE** o **UPDATE**, la integridad de los datos puede verse afectada añadiendo datos no válidos, modificando datos existentes tomando un valor incorrecto o eliminando datos que al hacerlo violan alguna regla.

La integridad de los datos se contempla en diferentes niveles. Las restricciones de dominio, transacciones y entidades definen las reglas para el mantenimiento de la integridad de las relaciones individuales. Las relaciones de integridad referencial aseguran que se mantienen las asociaciones necesarias entre las relaciones.



Las restricciones de integridad de la BD gobiernan la misma como un todo y las restricciones de integridad de transacciones controlan la forma en que se manipulan los datos, dentro de una o entre múltiples BD.(21)

Integridad de Dominio: Una restricción de integridad de dominio es una regla que define valores válidos para los atributos de las diferentes tablas de una BD. Puede ser necesario definir más de una restricción de dominio para describir por completo un dominio. Mientras más sean las reglas que se definan mejor será el funcionamiento de la base de datos.

Existe integridad de dominio básica, como no poder introducir letras en campos donde se va a almacenar números. Estas normas o reglas de integridad de dominio pueden indicar que campos son necesarios tener obligatoriamente con valores, para que la base de datos no tenga datos sin conectar en el caso de tener relaciones o dependencias entre tablas.

Algunas herramientas que asegurar la integridad de dominio son: definir los tipos de datos correctamente para cada campo, definir campos no nulos (NOT NULL) para evitar resultados inconsistentes, definir reglas (CHECK), que pueden ser aplicadas a nivel de tabla y columna y que restringen los valores durante las inserciones y modificaciones.

Para realizar estas validaciones PostgreSQL presenta los obligadores (CONSTRAINTS), que se encargan de chequear (CHECK) el cumplimiento de una o varias condiciones que deben cumplir el o los campos de una tabla y disparan excepciones, en caso de ser violadas. También existen los disparadores (TRIGGERS), los cuales, pueden validar campos de más de una tabla a la vez; como su nombre lo indica disparan una excepción, en caso de no cumplirse alguna(s) de las condiciones que valida. Estos hacen uso de funciones especiales, las cuales validan las condiciones que debe cumplir una tupla para ser aceptada como correcta e insertarla en la BD o retornan valor nulo (NULL), en caso de violarse alguna condición. Además pueden ser programadas para dispararse antes (BEFORE) o después (AFTER) de ocurrir eventos de eliminar (DELETE), insertar (INSERT) o actualizar (UPDATE) de una tupla en la tabla.

Para velar por la integridad de dominio de los datos de la BD, se definió como nulos (NULL) solo aquellos campos que no son de vital importancia para la misma, siendo estos la minoría. Se crearon reglas para



validar aquellos datos que poseen un rango de valores, validándose que solo sean introducidos datos válidos en el sistema.

Integridad de Transacciones: Define los estados por los que una tupla puede pasar válidamente, como son introducido, pendiente, seleccionado, enviado, cancelado y terminado. Está encargada de asegurar que el estado de una determinada tupla, no pase de un estado inicial a uno final, sin haber pasado por los estados intermedios. En el caso de la BD de la Plataforma VideoWeb no se encuentran restricciones de este tipo.

Integridad de Entidades: Las restricciones de entidades aseguran la integridad de las entidades que son modeladas por el sistema. Un caso de este tipo de restricción es el de las llaves primarias para cada tabla. En el caso de la BD que se está analizando cada entidad tiene definida su llave primaria, que no puede ser nula ni se puede repetir. Otra forma de esta integridad es la definida en el expediente de fase preparatoria, donde se chequea que la fecha de los hechos sea menor o igual que la de denuncia, al igual que ésta sea menor que la fecha de entrada del expediente al sistema.

Un clave principal constituye un hecho, o grupo de hechos, que distinguen esa entidad del resto de entidades que comparten un conjunto de hechos, o sea cada registro de una tabla debe tener un campo que identifique de modo exclusivo ese registro respecto al resto de registros de esa tabla. Definir correctamente el identificador principal para cada registro, es de vital importancia, ya que esto representa la principal herramienta que utiliza el servidor de BD para seleccionar la información que se necesite, evitando la duplicidad de registros, y el lanzamiento de excepciones por errores en los registros.

Para velar por la integridad de entidad de los datos de la BD, se definió una llave primaria para cada tabla, donde la mayor parte de ellas son de tipo autoincremental (SERIAL) y no permiten valores nulos (NOT NULL).

Integridad Referencial: Establece que una tupla en una relación que haga referencia a otra, deberá referirse a un valor existente en esa relación. Se define para asegurarse que las filas relacionadas entre tablas, no desaparezca ni varíe, cuando se lleve a cabo una modificación en los datos. Con esta integridad se limita la actividad que puede realiza un usuario sobre la base de datos. Por lo tanto, con la misma, se



da a comprender que existen relaciones entre tablas que deben permanecer invariables sea cual sea la actividad sobre ellas.

Para mantener esta integridad en la base de datos se utilizó: las llaves foráneas o externas, las cuales obligan a que los valores introducidos en las columnas marcadas por esta restricción correspondan a valores en la tabla referenciadas, así como permite realizar acciones en caso de actualización o eliminación de los valores a los que se hacen referencia, para esto, se definió que al realizar la actualización o eliminación de datos, las mismas se realizarían en cascada.

3.2.2. Normalización de la base de datos.

- La normalización de la BD es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener.(11)
- La normalización es un proceso mediante el cual se descompone una relación que no cumple con una serie de requisitos en relaciones más pequeñas que si las cumplan. Tiene como objetivo eliminar redundancias y anomalías de inserción, eliminación y actualización.(22)
- La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes que eviten las anomalías de actualización. Es la expresión formal del modo de realizar un buen diseño y provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.(4)

A modo de resumen la normalización es el proceso que permite eliminar la redundancia de los datos y evitar los problemas al insertar, eliminar y actualizar los datos, es decir, optimizar el trabajo con la información almacenada. Es un proceso en el cual se va comprobando el cumplimiento de una serie de reglas o restricciones por parte de un esquema de relación, donde cada regla que se cumple, aumenta el grado de normalización del esquema de relación. Si una regla no se cumple, el esquema de relación se debe descomponer en varios esquemas de relación que sí la cumplan por separado.

En este proceso existen varios niveles de normalización, pero los tres primeros son los más usados. Las reglas de normalización se relacionan entre ellas de forma dependiente, lo que indica que para que una



base de datos se encuentre en una determinada forma normal, primeramente tiene que cumplir con las reglas de los niveles inferiores de normalización.

Los niveles de normalización existentes son:(23)

- Primera Forma Normal (1NF).
- Segunda Forma Normal (2NF).
- Tercera Forma Normal (3NF).
- Forma Normal Boyce-Codd.
- Cuarta Forma Normal.
- Quinta Forma Normal o Forma Normal de Proyección-Unión.
- Forma Normal de Proyección-Unión Fuerte.
- Forma Normal de Proyección-Unión Extra Fuerte.
- Forma Normal de Clave de Dominio.

Cada nuevo nivel o forma acerca más el diseño a una BD verdaderamente relacional. Cada una tiene sus propias reglas. Durante el proceso de la normalización se fue comprobando que cada relación (tabla) cumpliera con un conjunto de reglas basadas en la clave primaria y las dependencias funcionales. Se llevaron a cabo una serie de pasos donde cada paso responde a una forma normal.

3.2.2.1. Primera forma normal (1FN).

La primera forma normal se definió para prohibir los atributos multievaluados, los atributos compuestos y sus combinaciones, es decir, establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Lo anterior indica que todos los atributos deben tener valores atómicos. Esta forma normal constituye el primer nivel en el proceso de normalización y es en ella donde se eliminarán los atributos multievaluados, los atributos compuestos y sus combinaciones.

La BD de la Plataforma VideoWeb cumple con la primera forma normal, ya que los datos de las relaciones involucradas son atómicos, o sea, no son ni multievaluados, ni compuestos, ni ninguna combinación de estos.



3.2.2.2. Segunda forma normal (2FN).

La segunda forma normal plantea que si todos los atributos que no forman parte de la llave primaria, dependen de manera total de los atributos que forman la llave primaria. Se dice que una entidad está en 2FN si está en 1FN y si sus atributos no llaves (ni primarias ni candidatas) son funcional y completamente dependientes de la llave primaria. Esta forma normal asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria, es decir, establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas.

La BD de la Plataforma VideoWeb cumple con la segunda forma normal, pues ya está en primera forma normal y cada atributo no primo de la relación es completamente dependiente de la clave primaria. La 2FN se aplicó a las relaciones que tenían claves primarias compuestas por dos o más atributos. De esta forma se evitan anomalías a la hora de realizar actualizaciones y se controlan la mayoría de los problemas de lógica.

3.2.2.3. Tercera forma normal (3FN).

La tercera forma normal plantea que la relación debe estar en segunda forma normal y que cada atributo de la relación que no está contenido dentro de la llave primaria dependa solo de la llave primaria y no de ningún otro atributo. Una relación está en tercera forma normal si, y sólo si, está en 2FN y además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. En esta forma normal se elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas.

Una vez llevada cada una de las relaciones a 2FN se pasó a eliminar las dependencias transitivas para obtener la 3FN con el objetivo de eliminar las redundancias que aún quedaban de la 2FN para evitar sufrir anomalías frente a las actualizaciones. Para ello, se eliminaron los atributos de cada relación que dependían transitivamente y se pusieron en nuevas relaciones con una copia del atributo o los atributos no clave de los que dependen.

Después de este proceso, la BD de la Plataforma VideoWeb, ha quedado normalizada hasta la 3FN. La BD está en 1FN puesto que se puede asegurar que cada tupla contiene exactamente un valor para cada



atributo de las tablas de la BD. También se cumple que los esquemas de relación están en 2FN, porque se encuentran en 1FN, y además todos los atributos que no son claves en las tablas, dependen totalmente de la clave primaria. Y finalmente se puede plantear que se encuentra en 3FN, porque está en 2FN, y además no existen dependencias transitivas entre llaves candidatas y atributos no primos.

3.2.3. Análisis de redundancia de 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 coste y tiempo de acceso a los datos. La redundancia siempre debe evitarse, aunque en proyectos grandes es imposible evitarla al 100%, lo que a veces es deseable por cuestiones de rendimiento.

Con un buen diseño de una base de datos se logrará evitar la aparición de información repetida o redundante. Lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

Después de llevar la BD de la Plataforma VideoWeb a 3FN, es decir, al normalizarla, queda libre de redundancias, ya que se elimina completamente la presencia de datos repetidos innecesariamente. La normalización es una técnica que elimina en gran medida esta dificultad, aunque en base de datos grandes es imposible eliminarla al ciento por ciento.

3.2.4. 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 por mala intención. La información que almacena una BD, está en constante riesgo de sufrir ataques que puedan provocar su modificación o pérdida, por ello es de vital importancia velar la seguridad de la misma, protegiéndola, en un primer momento, contra accesos no autorizados o cualquier acción que puedan violar la integridad de los datos o la confidencialidad de los mismos. En un segundo momento es necesario proteger los datos que se almacenan en la BD, para lo cual se deben realizar salvallas.



Capítulo 3: Validación del Diseño Realizado

La seguridad constituye una de las características más importantes por la que se debe velar al diseñar una BD. En la Plataforma VideoWeb, los datos que se almacenan en la BD constituyen la base fundamental del funcionamiento del sistema, ya que una pérdida de los mismos ocasionaría un paro en el funcionamiento de la plataforma, debido a que esta depende directamente de la información almacenada.

El administrador de la base de datos juega un papel importante en la seguridad de la misma porque es quien otorga privilegios a los usuarios y los clasifica, así como a los datos. Las órdenes de los administradores de bases de datos incluyen las siguientes acciones: creación de cuentas, concesión de privilegios y revocación de privilegios. La creación de cuentas sirve para controlar el acceso al SGBD en general, la concesión y revocación de privilegios para controlar autorizaciones discrecionales. Estas autorizaciones constituyen un importante mecanismo de seguridad.

El SGBD a utilizar para diseñar las bases de datos de la Plataforma VideoWeb –PostgreSQL– incluye formas de restringir el acceso al sistema. Esta función se denomina control de acceso y se pone en prácticas creando cuentas de usuarios y contraseñas para que el SGBD controle el proceso de entrada al sistema.

Como punto de partida se deben presentar los datos solo a quien esté autorizado. PostgreSQL, permite realizar configuraciones para controlar los permisos de los usuarios, utilizando tres niveles de acceso. En un primer nivel (nivel 0) se configuran los permisos de conexión para los host y los usuarios a la o las BDs, datos que se recogen en el archivo `pg_hba.conf` (24), en el mismo se define que PCs –dirección o direcciones IP– tendrán acceso, además a cuál o cuáles BDs y el modo en que podrán conectarse, que puede ser: conexión sin contraseña, validando el usuario y la contraseña para conectarse, o que rechace cualquier conexión desde el IP o rangos IP y usuarios seleccionados. En un segundo nivel (nivel 1), PostgreSQL permite configurar a que BD pueden acceder determinados usuarios, utilizando las opciones del archivo `pg_ident.conf` (24). Por último existe un tercer nivel (nivel 2), que permite configurar dentro de las BDs los accesos a las tablas, utilizando los comandos GRANT y REVOKE para permitir o denegar los permisos, respectivamente (24).

En la BD propuesta se establecen controles de seguridad para todos los datos almacenados, garantizando que solo los usuarios autorizados puedan efectuar operaciones correctas sobre algunas tablas o sobre toda la BD. Dentro de la BD el registro de los usuarios sigue un orden que permite controlar los accesos



de los mismos. Inicialmente la inserción de cada usuario lo realiza un administrador, el cual cuenta con los permisos necesarios para realizar operaciones de administración en la BD. Las contraseñas de los usuarios –tanto los de la aplicación como el administrador de la BD– viajarán y se almacenarán encriptadas, de modo que solo el usuario tenga conocimiento de la misma y evitando que intrusos se apoderen de ellas con fines maliciosos. Esto permite identificar cada usuario de forma única, lo que garantiza recoger constancia de quién realiza determinadas acciones en el sistema; acciones que estarán condicionadas por el rol que le sea asignado a cada usuario registrado. Los usuarios registrados tienen los permisos restringidos dentro de la BD, es decir, acceden a las tablas según el rol que desempeñan.

También es necesario proteger los datos que se almacenan en la BD, para lo cual se deben realizar salvadas. Para validar que no exista pérdida de información en caso de corrupción de los datos, PostgreSQL, permite la realización de salvadas (backups) mediante el volcado (dump) de la BD, los cuales pueden realizarse de forma automática o dejarlos a elección del cliente. Para el volcado de la BD se utiliza el comando `pg_dump` el cual indica al SGBD que se va a realizar la salva de la BD, además cuenta con una serie de parámetros adicionales que permiten indicar el nombre de la BD, el usuario y contraseña para conectarse a la misma, el nombre que tomará el archivo de salva y donde se desea localizar. El volcado de la BD puede realizarse en caliente, es decir, con el servicio postgres corriendo, lo cual constituye una ventaja, además se puede realizar a una o varias BD, además mantiene compatibilidad entre versiones, sus desventajas están marcadas por la lentitud que alcanza con BD de gran volumen y el consumo de recursos del sistema. En caso de desear restaurar una salva para su análisis o uso, se utiliza el comando `pg_restore` indicándole la salva que se desea restaurar, además de algún otro parámetro de los mencionados anteriormente.

3.2.5. Trazabilidad de las acciones.

- La trazabilidad es un sistema que deja rastro de las acciones que el usuario realiza con los datos. De esta forma, puede hacerse un seguimiento de todas las acciones que se han realizado sobre un registro concreto o todas las acciones que ha realizado un usuario determinado, pudiendo conformar un histórico de las acciones sobre los datos. La trazabilidad puede realizarse sobre tablas o sobre procesos. Incluso pueden crearse trazabilidades sobre errores o conflictos de la aplicación, (logs).(25)



En resumen la trazabilidad es un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada consulta desde su origen hasta su destino final. Es la capacidad para reconstruir la historia, recorrido o aplicación de una determinada consulta.

Los logs son un excelente mecanismo que tienen los SGBD para seguir las trazas en bases de datos. Estos son usados para registrar datos o información sobre (quién, qué, cuándo, dónde) convirtiéndose en un registro oficial de eventos durante un período de tiempo en particular. De esta manera el administrador de BD tendrá un efectivo archivo de evidencias que podrá utilizar en cualquier momento que necesite.

El SGBD que se utilizó para implementar la BD de la Plataforma VideoWeb (PostgreSQL) almacena en la carpeta pg_log, los registros –en archivos de texto– con la información del día, hora, usuario, máquina de origen, consulta, datos y tiempo que invirtió en la consulta, lo que permite llevar un control estricto de que se está haciendo en la BD.

Los logs son el mecanismo que trae por defecto el SGBD para garantizar la trazabilidad de las acciones, sin embargo se pueden utilizar los triggers (desencadenadores) que son un tipo de procedimiento almacenado que va a entrar en vigor antes (BEFORE), después (AFTER) o en lugar de (INSTEAD OF) efectuarse una operación cualquiera de modificación de datos sobre una tabla específica, que puede ser eliminar (DELETE), insertar (INSERT) o actualizar (UPDATE).

3.3. Validación funcional.

Para comprobar que la BD de la plataforma cumple con los requisitos funcionales definidos, es necesario realizarle pruebas antes de dar por terminado el proceso de diseño y creación de la misma. El propósito de estas pruebas es simular una carga de producción real y observar cómo se comporta la base de datos bajo cargas intensivas. Esto permite solucionar los problemas de rendimiento, antes de poner la BD en marcha. El objetivo general es ejecutar las consultas que se supone que sean las más utilizadas y percibir cómo responde el sistema a ellas.

3.3.1. Llenado voluminoso e inteligente de la base de datos.

Las pruebas de volumen centran su trabajo en poblar a la BD de grandes cantidades de información para determinar si existe algún momento donde la misma alcance sus límites de almacenamiento y cause fallas



en el sistema, identificándose así la carga máxima que puede manejar la base de datos en un período dado. En el mundo existen algunas herramientas para el llenado voluminoso de BD. Algunas personas utilizan scripts encontrados en Internet para llenar sus BD y realizarles pruebas, aunque no son recomendables debido a que se debe tener conocimientos de programación para esto.

Para la realización de pruebas de llenado voluminoso e inteligente a la BD se utilizó la herramienta EMS Data Generator 2008 for PostgreSQL, la cual permite la generación de datos para una o varias tablas a la vez, definiendo para cada campo el rango de valores admisibles, dependiendo del tipo, además de la cantidad de tuplas que se desean generar, validando de forma automática la integridad referencial, ya que genera los datos que provienen de otras tablas para evitar errores. La aplicación permite definir rangos de valores, máscaras, que permiten llenar la base de datos de forma sencilla y rápida. La generación de los datos es de forma aleatoria e incremental. Permite la utilización de resultados de consultas para generar listas de valores.

Las tablas fueron llenadas con un rango de datos equivalente al que tendrá la Base de Datos en un año aproximadamente. Se generó un volumen de 1500 y 3140 datos para las tablas Tipologia_Archivo_Multimedia, Archivo_Multimedia, AudioVisualRealizado, AudioVisualVariado, Serie, Temporada, Capitulo, Documental, Imagen, Imagen_Tipologia_Archivo_Multimedia, Materia, Musica, Servidores_FTP, Pelicula, Teleclase y VideoMusical, los cuales se insertaron en un intervalo de tiempo de 6 y 14 segundos respectivamente. Para las tablas Permisos, Rol, Interfaz, Usuario, Rol_Usuario, Rol_Permisos, Categoría, Node, Artículo_Contenido, Lista_de_Reproduccion, Seccion, Publicacion_Archivo_Multimedia, Publicacion_Archivo_Multimedia_Lista_de_Reproduccion y Categoría_Seccion, se generaron alrededor de 1400 y 3300 datos, insertándose en un tiempo de 5 y 11 segundos respectivamente. Por último se generaron un total de 4350, 8700 y 14500 datos, los cuales fueron insertados en todas las tablas que componen la BD de la Plataforma VideoWeb en un tiempo de 19, 31 y 51 segundos respectivamente. Los resultados obtenidos muestran un buen comportamiento y una rápida respuesta de la BD frente a un gran flujo de información.

3.3.2. Pruebas de carga intensiva.

Las pruebas de carga intensiva centran su trabajo en probar el funcionamiento de la BD través del grado de concurrencia de las consultas. En esta se comprueba el comportamiento de la BD frente a las



consultas más frecuentes que se realizarán a la misma, verificando que esta se comporte adecuadamente.

Para la realización de este tipo de pruebas a la BD de la Plataforma VideoWeb se utilizó la herramienta EMS SQL Query 2007 for PostgreSQL, con el objetivo de probar el correcto funcionamiento de la misma, así como los tiempos de ejecución. Las consultas generadas para la realización de pruebas, se escogen de acuerdo a las operaciones que se esperan sean las más utilizadas una vez implantada la BD.

- Listar los usuarios activos del sistema.

SELECT

```
usuario.usuarioid,  
usuario.nombre,  
usuario.usuario,  
usuario.estado,  
usuario.contrasenna
```

FROM

```
usuario
```

WHERE

```
usuario.estado = 1
```

ORDER BY

```
usuario.nombre
```

Para un volumen de 494 tuplas se generaron 258 resultados en un tiempo de 214 ms y para un volumen de 992 se arrojaron 514 resultados en un tiempo de 231 ms.

- Listar las publicaciones de archivos multimedia con calidad visual mayor o igual que 4.

SELECT

```
publicacion_archivo_multimedia.configuracion,  
publicacion_archivo_multimedia.fecha_publicacion,  
publicacion_archivo_multimedia.fecha_fin,  
publicacion_archivo_multimedia.cantvecesreprod,
```



```
publicacion_archivo_multimedia.puntuacion,  
publicacion_archivo_multimedia.calidadvisual
```

FROM

```
publicacion_archivo_multimedia
```

WHERE

```
publicacion_archivo_multimedia.calidadvisual >= 4
```

Para un volumen de 500 tuplas se generaron 259 resultados en un tiempo de 156 ms y para un volumen de 1000 se arrojaron 531 resultados en un tiempo de 163 ms.

- Listar los archivos multimedia existentes en el servidor de medias con tipología serie.

SELECT

```
public.archivo_multimedia.direccion_fisica,  
public.tipologia_archivo_multimedia.titulo_original,  
public.tipologia_archivo_multimedia.anno,  
public.serie.creador,  
public.serie.pais,  
public.serie.genero,  
public.serie.reparto
```

FROM

```
public.archivo_multimedia
```

```
INNER JOIN public.tipologia_archivo_multimedia
```

```
ON (public.archivo_multimedia.fk_tipologiaid = public.tipologia_archivo_multimedia.tipologiaid)
```

```
INNER JOIN public.serie
```

```
ON (public.tipologia_archivo_multimedia.tipologiaid = public.serie.pfk_tipologiaid)
```

Para un volumen de 3500 tuplas se generaron 500 resultados en un tiempo de 770 ms y para un volumen de 7000 se arrojaron 1000 resultados en un tiempo de 998 ms.

Las pruebas realizadas a una BD, nunca pueden tomarse como resultados reales, aunque dan un aproximado al mismo, dependiendo del grado de acercamiento que se utilice a los procesos de la vida



real. Aún así, la implantación y utilización de una BD, está marcada por varios factores entre ellos: cantidad de usuarios a conectarse y el grado de concurrencia de las solicitudes hechas por los mismos, los cuales hay que tener en cuenta siempre. Además, es necesario velar el cumplimiento de todos los requerimientos no funcionales propuestos para la BD, ya que validan el funcionamiento correcto y el máximo rendimiento de la misma. De aquí que, aunque las pruebas realizadas brinden resultados que puedan parecer alentadores, no pueden ser motivo de confianza, todo lo contrario, la mejor prueba que se le puede realizar a cualquier resultado informático lo constituye la interacción directa del usuario, de aquí los procesos de mantenimiento de software y la realización de versiones del mismo buscando mejorar o resolver, posibles errores que se detecten durante su vida útil.

3.4. Conclusiones.

Con la validación teórica del diseño realizado se chequearon una serie de parámetros, aspectos y consideraciones importantes a tener en cuenta para realizar un correcto diseño de una BD. Se definieron restricciones de integridad para garantizar la validez de los datos introducidos o modificados en la misma, así como el proceso de normalización hasta la Tercera Forma Normal (3NF), que dio lugar a la eliminación total de la redundancia de información. También se analizó la seguridad, realizando un control de acceso a la BD, haciendo uso en primer lugar de los archivos de configuración del SGBD PostgreSQL, y por otra parte, velando el cumplimiento de los requisitos funcionales propios del sistema, de validar los accesos de acuerdo al IP desde donde se realiza la conexión y el rol asignado al usuario en la misma. Para tener constancia de las acciones realizadas por los usuarios en la BD en cada momento, se hizo un análisis de la trazabilidad de las acciones, almacenándose los datos necesarios para realizar auditorías de control.

Para la validación funcional se procedió a realizar pruebas de carga intensiva y llenando voluminoso e inteligente a la base de datos. Para lo primero se utilizó la herramienta EMS SQL Query 2007 for PostgreSQL, comprobándose el tiempo de respuesta de la base de datos a las consultas más frecuentes. Para lo segundo se utilizó la herramienta EMS Data Generator 2008 for PostgreSQL, identificándose el funcionamiento de la BD al ser poblada de grandes cantidades de información. Ambas pruebas realizadas brindaron resultados alentadores, demostrado que cumple con los requisitos planteados por el analista del sistema, y que la misma está acta para su funcionamiento.



Conclusiones.

Los Sistemas de Bases de Datos han evolucionado considerablemente desde sus inicios en 1960, mejorando cada vez más los servicios que brindan, lográndose con ellos que la información sea almacenada sin redundancia, centralizada, y de forma estructurada. Además garantizan el acceso controlado a la información. Los Sistemas Gestores de Bases de Datos (SGBD), que facilitan la manipulación y actualización de los datos de una aplicación, han aumentado su potencialidad con el transcurso del tiempo, siendo hoy en día los SGBD Relacionales los más utilizados a nivel mundial y nacional. En la actualidad existen otros modelos de datos como las Bases de Datos Orientadas a Objetos y las Objeto Relacionales, que implementan características de la orientación a objeto. Las Bases de Datos Distribuidas y Multidimensionales también se han ido expandiendo en la medida que se va desarrollando la tecnología. El uso de cada una de ellas está condicionado por las prestaciones que debe brindar la base de datos dependiendo de las necesidades para lo cual fue creada.

Con el desarrollo de este trabajo se ha dotado a la Plataforma VideoWeb de una estructura para almacenar la información correspondiente a las medias, su gestión y administración que garantice el funcionamiento de la misma, dando respuesta así al problema que dio pie a este trabajo de diploma.

Fue seleccionado para desarrollar la Base de Datos (BD) de la Plataforma VideoWeb el Sistema Gestor de Base de Datos **PostgreSQL**, debido a que este es el servidor de BD de código abierto más potente que existe en la actualidad. Cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Además la UCI creó una comunidad de desarrollo de PostgreSQL, con el objetivo de garantizar soporte y mantenimiento a las aplicaciones que usen este SGBD, contribuyendo en gran medida con el prestigio y la calidad de la solución informática.

Se dio cumplimiento a los objetivos generales y específicos planteados, diseñándose e implementándose la base de datos para la Plataforma VideoWeb. Se desarrolló una BD que cumple con el proceso de normalización hasta la Tercera Forma Normal (3FN), posibilitando que en las tablas se encuentren datos que realmente pertenecen a ellas y que se disminuya la redundancia e inconsistencia en la información. Además la BD posee restricciones de integridad para garantizar la validez de los datos introducidos o modificados, análisis de la trazabilidad de las acciones, para tener constancia de las acciones realizadas



por los usuarios en la misma en cada momento y análisis de aspectos de seguridad, para evitar ataques e impedir cualquier acceso no autorizado, que pueda violar la integridad de los datos o la confidencialidad de los mismos.

Se analizó el funcionamiento de la BD, y la misma respondió satisfactoriamente a todos los casos de pruebas. Se puede concluir planteando que este trabajo ha dado lugar a una BD que vela por la integridad y la consistencia de los datos y que da cumplimiento a los requerimientos funcionales que demanda la Plataforma VideoWeb. Se empleó para su desarrollo y despliegue solamente herramientas de código abierto, lo que posibilita su uso y comercialización, sin restricciones por concepto de licencias de software propietario.



Recomendaciones.

Los logs son el mecanismo que trae por defecto el SGBD PostgreSQL para garantizar la trazabilidad de las acciones, elemento que se utilizó para garantizar ésta en la BD de la Plataforma VideoWeb. Se recomienda utilizar los triggers (desencadenadores) que son un tipo de procedimiento almacenado que va a entrar en vigor antes (BEFORE), después (AFTER) o en lugar de (INSTEAD OF) efectuarse una operación cualquiera de modificación de datos sobre una tabla específica, que puede ser eliminar (DELETE), insertar (INSERT) o actualizar (UPDATE) para garantizar la trazabilidad de las acciones en la misma. Para ello se recomienda crear un trigger llamado Registro_Traza que se active después (AFTER) de ejecutar cualquiera de las siguientes operaciones: eliminar (DELETE), insertar (INSERT) o actualizar (UPDATE) sobre una relación. Además de crear una nueva tabla llamada Registro_Traza, en la cual se almacene el resultado del trigger, es decir, las operaciones realizadas por los usuarios, la fecha y hora, la tabla en que se ejecutó la operación y el nombre del usuario que la llevó a cabo. En fin que se recopile el historial de las actividades de los usuarios en todo momento.

La implantación de este desencadenador añadiría a la Plataforma VideoWeb un mayor control de la trazabilidad de las acciones realizadas sobre una relación, favoreciendo en gran medida la seguridad de la misma.



Referencias Bibliográficas.

1. Valdés Gómez, Abel. *Servicio de Media Streaming para la Web. Portal Inter-nos. Módulos: Teleclases y TV*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2006.
2. Aportela Rodríguez, Ivett M. Intranets: Las tecnologías de información y comunicación en función de la organización. *Acimed: revista cubana de los profesionales de la información y la comunicación en salud*, 2007, vol. 16, nº 4, [Consultado el: 10 de Enero de 2009]. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=2472109>. ISSN 1024-9435.
3. Sarduy Domínguez, Yanetsys y Urra González, Pedro. *Sistemas de gestión de contenidos: En busca de una plataforma ideal*. [Consultado el: 20 de enero de 2009]. Disponible en: http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm.
4. Cardero Tasé, Licet y Bicet Zayas, Daliagna. *Diseño e implementación de la Base de Datos del Sistema Informático de los Tribunales Militares de Región*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
5. Albuquerque Arias, Amado. *Sistema Integrado de Gestión Estadística (SIGE). Rol Diseñador de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
6. Marqués Andrés, María Mercedes. *Apuntes de Ficheros y Bases de Datos*. Castellón de la Plana, España: Ingeniería Técnica en Informática de Gestión de la Universidad Jaime I, 2001.
7. Hansen, Gary W. y Hansen, James V. *Diseño y Administración de Bases de Datos*. 2da ed. Madrid: Pearson Educación S.A., 2000. ISBN 84-8322-002-4.
8. Date, Christopher J. *Introducción a los Sistemas de Bases de Datos*. Ciudad de La Habana: Editorial Félix Varela, 2003.



9. Miguel Castaño, Adoración De y Piattini Velthuis, Mario G. *Fundamentos y Modelos de Bases de Datos*. 2da ed. Madrid, España: Editorial RA-MA S.A., 1999. ISBN 9788478973613.
10. Labrada Batista, Arcel y Góngora Mora, Damir. *Sistema de Gestión de Información de la Facultad 8. Diseño de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
11. Domínguez Vaillant, Arodys E. y Miranda Gutiérrez, Duniel. *SIMDEC Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
12. Herrán Gascón, Manuel De La. *Administración y Optimización de Bases de Datos Oracle*. Red Científica, Última actualización: 2004. [Consultado el: 25 de enero de 2009]. Disponible en: <http://www.redcientifica.com/oracle>.
13. Booch, Grady. *Análisis y Diseño Orientado a Objetos con Aplicaciones*. 2da ed. Madrid, España: Pearson Prentice Hall S.A., 1996. ISBN 9684443528.
14. Castillo Santana, Leonel De Jesús y Alvarez Sánchez, Yaniel. *Base de Datos para la Residencia de la UCI*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
15. Masip, David. *Qué es Oracle*. DesarrolloWeb, Última actualización: 19 de julio de 2002. [Consultado el: 26 de enero de 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/840.php>.
16. Burbano Proaño, Diego Javier. *Análisis comparativo de bases de datos de código abierto*. Quito: 2006.
17. Contribuyentes De Todo El Mundo. *pgAdmin PostgreSQL Tools*. Última actualización: 13 de marzo de 2009. [Consultado el: 20 de marzo de 2009]. Disponible en: <http://www.pgadmin.org/>.
18. *PgAdmin III Guía_Ubuntu*, Última actualización: 10 de marzo de 2008. [Consultado el: 20 de marzo de 2009]. Disponible en: http://guia-ubuntu.org/index.php?title=PgAdmin_III.



19. Castillo Pérez, Idelsis y Valiente Mesa, Rosayda. *Lims De Calidad Del Centro De Ingeniería Genética Y Biotecnología: Desarrollo De La Base De Datos Del Módulo Sección De Mejoramiento De La Calidad*. Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2009.
20. Rocamonde Quintela, José Antonio. *Puesta en marcha de un servidor de bases de datos utilizando PostgreSQL 8.0*. 2001.
21. Riordan, Rebecca M. *Diseño de Base de Datos Relacionales Con Access y SQL Server*. 1ra ed. España: Editorial Mcgraw-Hill, 2000. ISBN 978-84-481-2770-1.
22. Hernández Castellanos, Yeneirys y Pérez García, Wendysh. *Diseño de Bases de Datos para la Intranet 2*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
23. Eduardo. *Normalización de bases de datos*. MySQL Hispano, Última actualización: 31 marzo de 2003. [Consultado el: 30 de abril de 2009]. Disponible en: <http://www.mysql-hispano.org/page.php?id=16&pag=1>.
24. Drake, Joshua y Worsley, John. *Practical PostgreSQL*. Headquarters: Sebastopol, California: Editorial O'Reilly Media, Inc., 2002. ISBN 978-1565928466.
25. Monte Galiano, Jose Luis. *Generadores de código*. 2006.



Bibliografía.

- Valdés Gómez, Abel. *Servicio de Media Streaming para la Web. Portal Inter-nos. Módulos: Teleclases y TV*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2006.
- Aportela Rodríguez, Ivett M. Intranets: Las tecnologías de información y comunicación en función de la organización. *Acimed: revista cubana de los profesionales de la información y la comunicación en salud*, 2007, vol. 16, nº 4, [Consultado el: 10 de enero de 2009]. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=2472109>. ISSN 1024-9435.
- Sarduy Domínguez, Yanetsys y Urra González, Pedro. *Sistemas de gestión de contenidos: En busca de una plataforma ideal*. [Consultado el: 20 de enero de 2009]. Disponible en: http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm.
- Cardero Tasé, Licet y Bicet Zayas, Daliagna. *Diseño e implementación de la Base de Datos del Sistema Informático de los Tribunales Militares de Región*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Alburquerque Arias, Amado. *Sistema Integrado de Gestión Estadística (SIGE). Rol Diseñador de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Marqués Andrés, María Mercedes. *Apuntes de Ficheros y Bases de Datos*. Castellón de la Plana, España: Ingeniería Técnica en Informática de Gestión de la Universidad Jaume I, 2001.
- Hansen, Gary W. y Hansen, James V. *Diseño y Administración de Bases de Datos*. 2da ed. Madrid: Pearson Educación S.A., 2000. ISBN 84-8322-002-4.
- Date, Christopher J. *Introducción a los Sistemas de Bases de Datos*. Ciudad de La Habana: Editorial Félix Varela, 2003.



- Miguel Castaño, Adoración De y Piattini Velthuis, Mario G. *Fundamentos y Modelos de Bases de Datos*. 2da ed. Madrid, España: Editorial RA-MA S.A., 1999. ISBN 9788478973613.
- Labrada Batista, Arcel y Góngora Mora, Damir. *Sistema de Gestión de Información de la Facultad 8. Diseño de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Domínguez Vaillant, Arodys E. y Miranda Gutiérrez, Duniel. *SIMDEC Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Herrán Gascón, Manuel De La. *Administración y Optimización de Bases de Datos Oracle*. Red Científica, Última actualización: 2004. [Consultado el: 25 de enero de 2009]. Disponible en: <http://www.redcientifica.com/oracle>.
- Booch, Grady. *Análisis y Diseño Orientado a Objetos con Aplicaciones*. 2da ed. Madrid, España: Pearson Prentice Hall S.A., 1996. ISBN 9684443528.
- Castillo Santana, Leonel De Jesús y Alvarez Sánchez, Yaniel. *Base de Datos para la Residencia de la UCI*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Masip, David. *Qué es Oracle*. DesarrolloWeb, Última actualización: 19 de julio de 2002. [Consultado el: 26 de Enero de 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/840.php>.
- Burbano Proaño, Diego Javier. *Análisis comparativo de bases de datos de código abierto*. Quito: 2006.
- Contribuyentes De Todo El Mundo. *pgAdmin PostgreSQL Tools*. Última actualización: 13 de marzo de 2009. [Consultado el: 20 de marzo de 2009]. Disponible en: <http://www.pgadmin.org/>.
- *PgAdmin III* Guia_Ubuntu, Última actualización: 10 de marzo de 2008. [Consultado el: 20 de marzo de 2009]. Disponible en: http://guia-ubuntu.org/index.php?title=PgAdmin_III.



- Castillo Pérez, Idelsis y Valiente Mesa, Rosayda. *Lims De Calidad Del Centro De Ingeniería Genética Y Biotecnología: Desarrollo De La Base De Datos Del Módulo Sección De Mejoramiento De La Calidad*. Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2009.
- Rocamonde Quintela, José Antonio. *Puesta en marcha de un servidor de bases de datos utilizando PostgreSQL 8.0*. 2001.
- Riordan, Rebecca M. *Diseño de Base de Datos Relacionales Con Access y SQL Server*. 1ra ed. España: Editorial Mcgraw-Hill, 2000. ISBN 978-84-481-2770-1.
- Hernández Castellanos, Yeneirys y Pérez García, Wendysh. *Diseño de Bases de Datos para la Intranet 2*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2007.
- Eduardo. *Normalización de bases de datos*. MySQL Hispano, Última actualización: 31 marzo de 2003. [Consultado el: 30 de abril de 2009]. Disponible en: <http://www.mysql-hispano.org/page.php?id=16&pag=1>.
- Drake, Joshua y Worsley, John. *Practical PostgreSQL*. Headquarters: Sebastopol, California: Editorial O'Reilly Media, Inc., 2002. ISBN 978-1565928466.
- Monte Galiano, Jose Luis. *Generadores de código*. 2006.
- Hector Bayarre, R. H., Ed. (2004). *METODOLOGIA DE LA INVESTIGACION*. Ciudad de La Habana, Cuba. Editorial Ciencias Medicas, Centro Nacional de Información de Ciencias Medicas.
- Roberto Hernández Sampieri, C. F. C., Pilar Baptista Lucio (2003). *METODOLOGIA DE LA INVESTIGACION*, Ciudad de La Habana, Cuba. Editorial Félix Varela.
- Charte Ojeda, Francisco. *Bases de Datos con Microsoft Visual Basic.Net*. Madrid: Ediciones Anaya Multimedia. [Consultado el: 3 de abril de 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00581.pdf>



- Rolando Alfredo Hernández León, S. C. G. (2002). EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. Universidad de las Ciencias Informáticas. Ciudad de la Habana. Cuba.
- Espinosa, Humberto. *PostgreSQL: una alternativa de DBMS Open Source*. Última actualización: 2005. [Consultado el: 11 de marzo de 2009]. Disponible en: http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
- Zayas., C. A. d. (1995). METODOLOGIA DE LA INVESTIGACION CIENTIFICA. M. F. G. CENTRO DE ESTUDIOS DE EDUCACION SUPERIOR. SANTIAGO DE CUBA, CUBA.
- Díaz Villanueva, Wladimiro. *Base de datos objeto-relacionales*. Última actualización: 2002. [Consultado el: 16 de febrero de 2009]. Disponible en: http://informatica.uv.es/iiguia/DBD/Practicas/boletin_1.pdf.



Glosario de Términos.

.Net: Proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

ACID: (Atomicity, Consistency, Isolation, Durability). Propiedades importantes de las BD relacionales.

- **Atomicidad (Indivisible):** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la Base de Datos.
- **Aislamiento:** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

Apache: Servidor web HTTP de código abierto con versiones para varias plataformas incluyendo Unix (BSD, GNU/Linux, etc.), Windows y Macintosh.

BD: (Base de Datos). Colección de datos interrelacionados que se puede utilizar por uno o más programas de aplicación, puede considerarse una colección de datos variables en el tiempo.

CASE: (Computer-Aided Software Engineering, Ingeniería de Software Asistida por Ordenador). Diversas 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.

CMS: Un Sistema de Gestión de Contenidos (Content Management System en inglés, abreviado CMS) es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

Ergonomía: Estudio de datos biológicos y tecnológicos aplicados a problemas de mutua adaptación entre el hombre y la máquina. Estudio de la adaptación de las máquinas, muebles y utensilios a la persona que



los emplea habitualmente, para lograr una mayor comodidad y eficacia.

Ergonómica: Perteneiente o relativo a la ergonomía. Dicho de un utensilio, de un mueble o de una máquina: Adaptados a las condiciones psicofisiológicas del usuario.

Fibra óptica: Filamento fino de cristal u otro material transparente a través del que se puede transmitir, mediante reflexión total interna, un haz de luz de una señal codificada.

FTP: (File Transfer Protocol, Protocolo de Transferencia de Archivos). Protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

GNU: Proyecto para la creación de un sistema operativo de libre distribución. Como núcleo del sistema se suele usar el de Linux, dando lugar al conjunto llamado "GNU/Linux".

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

Hibernate: Herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto).

HTTP: (HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto): Protocolo de Internet que se utiliza para proporcionar información mediante la World Wide Web.

IP: (Internet Protocol, Protocolo de Internet). Protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Java: Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

Linux: Núcleo o kernel del sistema operativo libre denominado GNU/Linux. Lanzado bajo la licencia pública general (GPL - General Public License) de GNU y desarrollado gracias a contribuciones



provenientes de todo el mundo.

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

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario.

NHibernate: Conversión de Hibernate de lenguaje Java a C# para su integración en la plataforma .NET.

Nova: Distribución de GNU/Linux hecha en la Universidad de las Ciencias Informáticas.

PHP: Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Plugin: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Python: Lenguaje de programación interpretado creado por Guido van Rossum en el año 1991.

RAID: (Redundant Array of Independent/Inexpensive Disks). Es un término que hace referencia a un conjunto de discos redundantes e independientes. Se utiliza para mejorar el rendimiento, la tolerancia, fallos y errores en los discos, así como también mejora la integridad de los datos.

Reproductor: Programa o control de cliente que recibe contenido multimedia digital transmitido desde un servidor o reproducido a partir de archivos locales. El VLC es un ejemplo de reproductor.

RTSP: (Real Time Streaming Protocol, Protocolo de Transmisión en Tiempo Real): Protocolo de Internet que suministra secuencias de audio y video en tiempo real, en directo o almacenado a través de una red.

Script: Se refiere a un archivo de texto que recoge todo el código de la BD, que muestra un guión y conjunto de instrucciones. Es ejecutado por el PostgreSQL para su uso y crear todas las tablas.

SGBD: (Sistema Gestor de Base de Datos). Conjunto coordinado de programas, procedimientos, lenguajes. Que suministra tanto a usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos, manteniendo su integridad, confidencialidad y seguridad.



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).

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.

Streaming: Término que se refiere a ver u oír un archivo directamente en una página web sin necesidad de descargarlo antes al ordenador.

TCP: (Transmission Control Protocol, Protocolo de Control de Transmisión). Protocolo de comunicación orientado a conexión y fiable ubicado en el nivel de transporte del modelo OSI, es uno de los protocolos fundamentales en Internet.

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.

URL: (Uniform Resource Locator, Localizador Uniforme de Recurso): Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos como documentos e imágenes en Internet para su localización.

VLC: (VideoLAN Client). Es un reproductor multimedia del proyecto VideoLAN. Es software libre, soporta numerosos códecs de audio y video, así como diferentes tipos de archivos y puede ser utilizado como servidor en unicast o multicast, en IPv4 o IPv6, en una red de banda ancha.

Windows: Familia de sistemas operativos desarrollados y comercializados por Microsoft.