



Diseño de la base de datos para la plataforma de televisión informativa PRIMICIA versión 2.0

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

Autor:

Glenda María Santos Hernández

Tutor:

Ing. Enrique Almeida Maldonado

Co-Tutor:

Ing. José Andrés Hernández Bustio

La Habana 27 de junio del 2011
"Año 53 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.

Agradecimientos

A mi abuela Georgina por darme todo el amor del mundo, por siempre estar ahí cuando más te necesitaba, por la educación que me diste durante toda mi vida y por quererme tanto. Eres lo más importante que tengo en la vida, siempre estarás en mi corazón...

A Mercedes, mi mamita linda, por ser mi guía, mi amiga incondicional, por darme tanto amor y cariño, por guiarme siempre por buenos caminos y darme tanta confianza. ¡Al fin lo logramos!

A mi papi Arturo, por ser el ejemplo a seguir en mi vida, sabes que significas mucho para mí. Gracias por darme tanto apoyo y siempre creer que yo sí podía lograrlo algún día...

A mi esposo Daniel por estar ahí día a día conmigo, por darme su vida, por sacrificar cada segundo para ayudarme, por comprenderme y por amarme tanto. Gracias por esos momentos tan lindos que hemos pasado juntos, por ser mi compañero en las buenas y en las malas, sabes que mi vida sin ti hubiera sido muy distinta. Te amo....

A mis tías Lucy y Naña, a mi prima Perla y a toda mi familia por apoyarme siempre y estar pendiente a todo momento de mí. Los quiero...

A mi suegra Tina, gracias por ser como una madre para mí, por darme tanto cariño y quererme tanto. Siempre te llevaré en mi corazón...

A mi merito Darilys, a mi cuñada Deisy, a mis sobrinitos lindos Alejandro, Arianne y José, gracias por dejarme entrar en sus vidas y ayudarme tanto.

A todos las amistades mías de Cienfuegos, a mis amigas Yuliet, Laurita y mi tata Yenia, gracias por brindarme la posibilidad de contar siempre con ustedes y demostrarme ser muy buenas amigas.

A las kankas (Ivelín, Lisandra, Yamilé, Yisell, Saily, Krysna, Cecilia y Adaily) por todos esos momentos lindos que compartimos juntas y por todas las travesuras que hicimos durante 2 años. Nunca las olvidaré...

Agradecimientos

A mi tutor y amigo Enrique agradecerle especialmente por poder contar con él para la tesis, por todo el apoyo que me brindó, por ser comprensivo y además por aguantar mis estresantes nervios. Kikillo al fin te libras de mí!!!

A todos los profesores del proyecto PRIMICIA que de una forma u otra han contribuido a mi educación profesional y que además me han educado para la vida. Gracias Geo, Carlos y Rafa.

A mis amistades de cinco años, José Andrés, Genry, Yelina, Carinefy, Dayanis, gracias por haberme brindado su amistad incondicional.

A mis amigos, Mario Raúl y Félix por sacarme de varios apuros y por estar siempre atento a mi tesis. Gracias.

A todas aquellas personas que se preocuparon por mí y que algún día me preguntaron ¿Cómo va esa tesis? Gracias por su interés.

A nuestro Comandante en Jefe Fidel Castro Ruz por darme la oportunidad de formar parte de estos jóvenes del futuro, por ser el eterno creador de esta genuina obra de la Revolución.

A todo ustedes mil gracias...

Dedicatoria

*A mi abuela Georgina.
A mi mamá Mercedes y mi papá Arturo.
A mi esposo Daniel.
A la futura luz que alumbrará mi vida.*

Declaración de Autoría

Declaro que yo Glenda María Santos Hernández soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (6) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de junio del 2011.

Glenda M. Santos Hernández

Autora

Ing. Enrique Almeida Maldonado

Tutor

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 PRIMICIA del centro productivo GEYSED (Geoinformática y Señales Digitales) perteneciente a la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI). Dicho proyecto tiene como objetivo personalizar la transmisión de un canal informativo, el cual se soporta sobre la plataforma de televisión informativa PRIMICIA, empleando para el desarrollo de este producto solamente herramientas de software libre que posibilite su uso y comercialización, evitando así restricciones de licencias de software propietario.

Actualmente existe una primera versión de dicha plataforma, pero esta no es lo suficientemente configurable como se quisiera. Lo que se pretende realizar es un producto que sea lo más genérico posible y que se pueda ajustar completamente a cualquier necesidad de los usuarios. Por esta razón se ha decidido migrar hacia la segunda versión de la plataforma, la cual contará además con mayor robustez, eficiencia y genericidad que la versión anterior.

Para lograr lo dicho anteriormente este trabajo centra su objetivo en dotar a la nueva versión de la plataforma PRIMICIA de una estructura en la cual se pueda almacenar toda la información referente a la gestión del canal.

En este documento se plasma todo el proceso de diseño, implementación y prueba de la base de datos de la versión 2.0 de la plataforma de televisión informativa PRIMICIA, así como el desarrollo de los artefactos correspondientes al rol Diseñador de Base de Datos.

Palabras claves: base de datos, canal, información, plataforma, PRIMICIA, televisión informativa.

Índice de Figuras

| | |
|---|----|
| Figura 1: Niveles de una base de datos..... | 16 |
| Figura 2: Comparación entre sistemas gestores de base de datos..... | 17 |
| Figura 3: Fases e Iteraciones de la Metodología RUP. | 21 |
| Figura 4: Diagrama de clases persistentes. | 31 |
| Figura 5: Diagrama Entidad- Relación. | 34 |
| Figura 6: Funcionamiento del Streaming Replication. | 43 |
| Figura 7: Funcionamiento del Hot Standby..... | 44 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Comparación entre sistemas gestores de base de datos..... | 17 |
| Tabla 2: Descripción de la tabla signal..... | 35 |
| Tabla 3: Descripción de la tabla signal_changes..... | 35 |
| Tabla 4: Descripción de la tabla section..... | 35 |
| Tabla 5: Descripción de la tabla headband..... | 35 |
| Tabla 6: Descripción de la tabla image..... | 35 |
| Tabla 7: Descripción de la tabla media..... | 35 |
| Tabla 8: Descripción de la tabla audio..... | 35 |
| Tabla 9: Descripción de la tabla music_count..... | 36 |
| Tabla 10: Descripción de la tabla video..... | 36 |
| Tabla 11: Descripción de la tabla author..... | 36 |
| Tabla 12: Descripción de la tabla template..... | 36 |
| Tabla 13: Descripción de la tabla news..... | 36 |
| Tabla 14: Descripción de la tabla default_news..... | 36 |
| Tabla 15: Descripción de la tabla configurable_news..... | 36 |
| Tabla 16: Descripción de la tabla archived_news..... | 37 |
| Tabla 17: Descripción de la tabla modified_news..... | 37 |
| Tabla 18: Descripción de la tabla published_news..... | 37 |
| Tabla 19: Descripción de la tabla block..... | 37 |
| Tabla 20: Descripción de la tabla block_published_news..... | 37 |
| Tabla 21: Descripción de la tabla keyword..... | 37 |
| Tabla 22: Descripción de la tabla media_keyword..... | 37 |
| Tabla 23: Descripción de la tabla screen..... | 38 |
| Tabla 24: Descripción de la tabla archived_screen..... | 38 |
| Tabla 25: Descripción de la tabla format..... | 38 |
| Tabla 26: Descripción de la tabla modified_screen..... | 38 |

| | |
|---|----|
| Tabla 27: Descripción de la tabla genre..... | 38 |
| Tabla 28: Descripción de la tabla layout..... | 38 |
| Tabla 29: Descripción de la tabla parameter..... | 38 |
| Tabla 30: Descripción de la tabla web_source..... | 39 |
| Tabla 31: Descripción de la tabla transmission..... | 39 |
| Tabla 32: Descripción de la tabla property..... | 39 |
| Tabla 33: Descripción de la tabla effect..... | 39 |
| Tabla 34: Descripción de la tabla property_effect..... | 39 |
| Tabla 35: Descripción de la tabla screen_property..... | 39 |
| Tabla 36: Descripción de la tabla service..... | 39 |
| Tabla 37: Descripción de la tabla element_information..... | 40 |
| Tabla 38: Descripción de la tabla method..... | 40 |
| Tabla 39: Descripción de la tabla archived_default_news..... | 40 |
| Tabla 40: Descripción de la tabla archived_configurable_news..... | 40 |
| Tabla 41: Descripción de la tabla published_default_news..... | 40 |
| Tabla 42: Descripción de la tabla published_configurable_news..... | 40 |
| Tabla 43: Descripción de la tabla public.sf_guard_user..... | 40 |
| Tabla 43: Descripción de la tabla public.sf_guard_group..... | 41 |
| Tabla 44: Descripción de la tabla public.sf_guard_permission..... | 41 |
| Tabla 45: Descripción de la tabla public.sf_guard_group_permission..... | 41 |
| Tabla 46: Descripción de la tabla public.sf_guard_user_permission..... | 41 |
| Tabla 47: Descripción de la tabla public.sf_guard_forgot_password..... | 41 |
| Tabla 48: Descripción de la tabla public.sf_guard_remember_key..... | 41 |
| Tabla 49: Descripción de la tabla public.sf_guard_user_group..... | 41 |
| Tabla 51: Comparación entre versiones de base de datos de PRIMICIA..... | 55 |

Índice de Contenidos

| | |
|--|----------|
| Introducción | 1 |
| Capítulo 1: Fundamentación teórica. | 4 |
| 1.1 Introducción..... | 6 |
| 1.2 Conceptos asociados al dominio del problema..... | 6 |
| 1.2.1 Gestión..... | 6 |
| 1.2.2 Dato..... | 6 |
| 1.2.3 Información..... | 7 |
| 1.2.4 Gestión de la Información..... | 7 |
| 1.2.5 Plataforma..... | 7 |
| 1.2.6 Televisión..... | 7 |
| 1.3 Descripción actual de los procesos del negocio. | 8 |
| 1.3.1 Procesos de gestión de la plataforma de televisión informativa PRIMICIA..... | 8 |
| 1.4 Otras soluciones existentes en el mundo..... | 9 |
| 1.5 Introducción a las Bases de Datos (BD)..... | 9 |
| 1.5.1 Organización de las base de datos. | 10 |
| 1.5.2 Clasificación de las bases de datos..... | 10 |
| 1.5.3 Modelos de Bases de Datos..... | 12 |
| 1.6 Sistemas Gestores de Bases de Datos (SGBD)..... | 14 |
| 1.6.1 Objetivos de los SGBD. | 14 |
| 1.6.2 Arquitectura de los SGBD..... | 15 |
| 1.6.3 Principales gestores de bases de datos..... | 16 |
| 1.6.4 Sistema gestor de base de datos seleccionado..... | 17 |
| 1.7 Herramienta de administración y desarrollo para PostgreSQL..... | 19 |
| 1.8 UML como soporte para el modelado de la base de datos..... | 19 |
| 1.9 Visual Paradigm como Herramienta CASE utilizada..... | 19 |
| 1.10 RUP como metodología para el desarrollo de la solución propuesta..... | 20 |

| | | |
|---|--|-----------|
| 1.11 | Replicación de datos..... | 22 |
| 1.11.1 | Tipos de entornos de réplicas..... | 22 |
| 1.11.2 | Modelo de distribución de datos..... | 23 |
| 1.11.3 | Técnicas de replicación | 23 |
| 1.12 | Herramienta de replicación de la solución propuesta..... | 24 |
| 1.13 | Conclusiones parciales..... | 25 |
| Capítulo 2: Descripción y análisis de la solución propuesta..... | | 26 |
| 2.1 | Introducción | 26 |
| 2.2 | Integración de la solución con los módulos existentes..... | 26 |
| 2.3 | Requisitos del sistema..... | 26 |
| 2.3.1 | Requisitos funcionales..... | 26 |
| 2.3.2 | Requisitos no funcionales..... | 28 |
| 2.6 | Diagrama de clases persistentes..... | 29 |
| 2.6.1 | Descripción de las clases del diagrama de clases persistentes..... | 32 |
| 2.7 | Diseño de la base de datos..... | 32 |
| 2.7.1 | Descripción de las tablas del Modelo Entidad- Relación..... | 34 |
| 2.8 | Optimización de consultas..... | 38 |
| 2.9 | Propuesta de replicación de la base de datos..... | 42 |
| 2.10 | Conclusiones parciales..... | 44 |
| Capítulo 3: Validación del diseño realizado..... | | 45 |
| 3.1 | Introducción | 45 |
| 3.2 | Validación teórica del diseño de la base de datos..... | 45 |
| 3.2.1 | Integridad..... | 45 |
| 3.2.2 | Normalización de la base de datos..... | 46 |
| 3.2.3 | Análisis de la redundancia de la información..... | 48 |
| 3.2.4 | Análisis de la seguridad de la base de datos..... | 48 |
| 3.3 | Validación funcional del diseño de la base de datos..... | 49 |
| 3.3.1 | Pruebas de rendimiento..... | 50 |

| | |
|--|-----------|
| 3.3.2 Procedimientos almacenados..... | 53 |
| 3.4 Comparación con la versión 1.0 de la BD de PRIMICIA..... | 54 |
| 3.5 Conclusiones parciales..... | 56 |
| Conclusiones | 57 |
| Recomendaciones | 58 |
| Referencias Bibliográficas..... | 59 |
| Bibliografía consultada..... | 63 |
| Glosario de términos..... | 67 |

Introducción

Las Tecnologías de la Información y las Comunicaciones a lo largo de la historia han facilitado llevar el carácter innovador y creativo al ámbito de la comunicación, permitiendo la conexión entre las personas a nivel mundial, eliminando así barreras como el tiempo y el espacio.

En la sociedad de hoy, más que ninguna otra anterior, se está viviendo una etapa donde la información juega un papel primordial, lo que la ha llevado a convertirse en una fuente de vital importancia. La cantidad de información a la que se expone la humanidad es inmensamente enorme y aumenta cada día más. Hoy se necesita conocer mucho más que en etapas anteriores, puesto que el campo del conocimiento ha progresado notablemente. Por este motivo a este tiempo se le ha llamado "la era de la información".

Disímiles organizaciones han tomado conciencia de la gran necesidad de controlar toda esta información, utilizando como el principal soporte de almacenamiento masivo las microcomputadoras. Estas han permitido que la información sea almacenada por tiempo infinito y sin algún riesgo de deterioro o desgaste, como suele suceder con otros formatos de almacenamiento como los libros y revistas.

Inmerso en este proceso de la información se encuentra la informática, la cual ha traído para distintos ámbitos de la vida humana una calidad superior, ya que ha facilitado la realización de labores y tareas, las cuales necesitan de cierta información para ser llevadas a cabo, así como el procesamiento de estas. Por esta razón en la actualidad la informática es usada en la mayoría de los trabajos y en casi todos los ambientes por los que se mueve la humanidad. Sin duda alguna, la mayor ventaja que esta ha reportado es el poder compartir toda la información que se quiera, facilitando de esta manera la comunicación.

En el mundo de hoy el intercambio de información continúa constituyendo una necesidad indispensable. A raíz de esto se ha valorado la posibilidad de perfeccionar las formas de comunicación y como objetivo final los medios de comunicación como la radio y la televisión, los cuales son de relevante necesidad para mantener informado a multitudes. Sin embargo, en muchas partes del mundo, incluyendo a nuestro país, existen grandes dificultades para transmitir la información por dichos medios de comunicación a los llamados entornos cerrados donde existe un gran flujo de personas que necesitan información a la medida. Ejemplo de esto son los hoteles, terminales de transportación, universidades, entre otros.

Actualmente muchas de las empresas, instituciones y centros educacionales, que cuentan con una infraestructura de redes, desearían transmitir a través de ella contenidos informativos. Para poder lograr tal objetivo se utilizan aplicaciones informáticas capaces de administrar y gestionar la información que se quiere transmitir.

La Universidad de las Ciencias Informáticas (UCI), y específicamente el proyecto PRIMICIA del Centro de Desarrollo GEySED (Geoinformática y Señales Digitales), consecuente con esta necesidad de información trabaja en el desarrollo de sistemas que están orientados a la transmisión de televisión informativa. Como consecuencia de esto se crea la plataforma PRIMICIA, la cual está encargada de emitir informaciones en disímiles formatos.

Actualmente existen varias versiones que son sustentadas precisamente en la plataforma de televisión informativa PRIMICIA, como son Señal 3, un espacio informativo dirigido a toda la comunidad Universitaria, Señal ACN, espacio donde se informa noticiosamente a todos los colaboradores que actualmente están prestando servicios en otros países, entre otros.

Esto, a su vez, ha permitido que “PRIMICIA” se considere un producto informático como respuesta a la automatización de la transmisión de televisión informativa, pero actualmente no es tan configurable como se desea. Lo que se pretende es tratar un producto que sea lo más genérico posible, que no dependa del entorno donde se proyecte y que se pueda acomodar a cualquier petición de los clientes. Para lograr este objetivo se ha decidido migrar hacia la segunda versión, donde PRIMICIA, sin lugar a dudas, tendrá mayor robustez, eficiencia y genericidad que versiones anteriores.

Debido a estos cambios que se quieren realizar, la estructura de almacenamiento con la que se cuenta no es capaz de satisfacer las nuevas funcionalidades que se desarrollarán para lograr dicha configuración. Esta aplicación que se pretende realizar dependerá para su correcto funcionamiento de la lectura de determinados datos que se encuentran almacenados en una estructura denominada base de datos. La manipulación dentro de este tipo de aplicaciones que necesitan interactuar con mucha información, constituye un factor primordial y de gran importancia para el desarrollo satisfactorio de diversas funcionalidades como pueden ser publicación de noticias, emisión de reportes, entre otras. La pérdida de alguna información puede traer consigo que esta no llegue de la manera eficaz y constante que se requiere, lo cual traería inestabilidad en el producto y mal funcionamiento del mismo.

De esta manera queda evidenciado como **problema a resolver**: “La inexistencia de una estructura para el almacenamiento de la información que garantice el correcto funcionamiento de la plataforma de televisión informativa PRIMICIA en su versión 2.0.”

Se tiene como **objeto de estudio**: El proceso de diseño de bases de datos relacionales.

El **campo de acción** está centrado en el diseño lógico y físico de la base de datos para PRIMICIA en su versión 2.0.

Como **objetivo general** de la investigación se plantea: Desarrollar la base de datos para la versión 2.0 de PRIMICIA y como **objetivos específicos** se tienen:

- ✓ Modelar la base de datos.
- ✓ Implementar la base de datos.
- ✓ Garantizar la seguridad de los datos.

Como **idea a defender** se plantea: Con el diseño de la base de datos para la plataforma de televisión informativa PRIMICIA en su versión 2.0 se almacenará correctamente la información referente a las noticias, asegurándose el adecuado intercambio de información dentro de la plataforma y el desarrollo exitoso del proyecto.

Para darle cumplimiento a los objetivos del presente trabajo se definieron varias tareas de investigación, estas son:

1. Identificar requisitos del sistema y herramientas para el correcto diseño de la base de datos.
2. Realizar la integración y comunicación con los módulos existentes del sistema.
3. Elaborar réplicas de la base de datos y datos persistentes.
4. Elaborar la propuesta de diseño de la base de datos a utilizar en el desarrollo y despliegue del sistema.
5. Validar teóricamente la propuesta de diseño de la base de datos.
6. Validar funcionalmente la propuesta de diseño de la base de datos.

Después de haber cumplido con cada una de estas tareas se espera obtener como **posible resultado** la base de datos para la plataforma de televisión informativa PRIMICIA en su versión 2.0, donde se almacene correctamente la información referente al canal.

Para el desarrollo de esta investigación se hace necesario emplear varios métodos científicos, dentro de los cuales se puede encontrar los teóricos y empíricos. Dentro de los teóricos se tiene:

✚ **Análisis Histórico – Lógico:** Se emplea para poder realizar un estudio de las principales tendencias de las herramientas a utilizar para el modelado de la base de datos, así como, los Sistemas Gestores de Bases de Datos (SGBD). Además se puede analizar la evolución, las particularidades y ventajas de varias herramientas para seleccionar la que permita un óptimo nivel de aprovechamiento y seguridad.

✚ **Analítico – Sintético:** este es empleado para poder realizar un bosquejo del estado del arte del tema que se está investigando, cuáles han sido los principales avances de las bases de datos hasta la actualidad y el impacto que ha sido provocado por las últimas tecnologías de los SGBD.

✚ **Modelación:** este es un método científico que permite abstraerse para poder expresar gran parte de la realidad. Lo primordial de la modelación es la estrecha relación que existe entre el modelo y el objeto que se modela. Se usa este método para poder lograr una modelación de la realidad mediante el modelo de datos, posteriormente mediante el diseño lógico de la base de datos, y llevarlo finalmente al modelo físico de la base de datos.

Como método empírico que se utiliza está:

✚ **La Observación:** se aplica este método con el objetivo de lograr un acercamiento y familiarización con los procesos de negocios existentes en el campo de acción. Con esto se obtienen las entidades para modelar la presente situación del mundo real.

El presente documento está estructurado en tres capítulos donde se describe toda la investigación realizada, así como los resultados obtenidos. Además se compone de Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía consultada, Glosario de términos y Anexos.

Capítulo 1: Fundamentación teórica.

En este capítulo se realiza un acercamiento al proceso de diseño de las base de datos relacionales, las herramientas que se utilizan en el mundo para facilitar su diseño e implementación, así como un estudio de las principales tendencias de las tecnologías que son utilizadas para guiar todo el proceso de la base de datos, además de los gestores para el desarrollo de las mismas.

Capítulo 2: Descripción y análisis de la solución propuesta.

En este capítulo se abordan los procesos principales que se llevan a cabo para diseñar e implementar la base de datos de la nueva versión de la plataforma PRIMICIA, además de la justificación de la arquitectura que se empleará durante el diseño de la base de datos, así como los principales requisitos funcionales y no funcionales del sistema.

Capítulo 3: Validación del diseño realizado.

En este capítulo se realiza la validación del diseño de la base de datos propuesta, la cual se divide en dos partes fundamentales, validación teórica y validación funcional. Además se mostrarán las reglas de integridad y normalización que se tuvieron en cuenta para la calidad del diseño realizado.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

El presente capítulo presenta una reseña del tema de las Bases de Datos (BD), además de abordar las tendencias, tecnologías y metodologías utilizadas en el mundo para el desarrollo de las mismas. Otros de los aspectos que conforman este capítulo son la explicación de todo el proceso llevado a cabo por la plataforma de televisión informativa PRIMICIA, la descripción de los conceptos fundamentales asociados al dominio del problema, el análisis del gestor de base de datos a utilizar, así como el uso de herramientas de desarrollo.

1.2 Conceptos asociados al dominio del problema.

1.2.1 Gestión

Este término proviene del latín *gestio*. La Real Academia Española define gestión como acción y efecto de gestionar. (Española, 2009)

Pablo Murray expresa que la gestión no es más que la herramienta que engloba fases de la administración siendo la gestión misma un elemento con desarrollo de la administración. Por otra parte es la actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de una organización. (Murray, 2003)

En general, se puede decir que la gestión implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto.

1.2.2 Dato

La palabra *dato* se define como los hechos y cifras que se pueden utilizar como base para el razonamiento, la discusión o el cálculo. Además Sergio D'Ambrosio esclarece que “por sí mismos los datos no tienen capacidad de comunicar un significado y por tanto no pueden afectar el comportamiento de quien los recibe. Para ser útiles, los datos deben convertirse en información para ofrecer un significado, conocimiento, ideas o conclusiones.” (Heber, 2009)

Según las investigaciones de Sanguino se conceptualiza la palabra dato como “elemento o cantidad conocida que sirve de base para la resolución de un problema”. Principio en el que se asienta una discusión. Elemento o base para la formulación de un juicio. (Heber, 2009)

Los datos por sí solos no tienen significado alguno, es decir, deben tener una referencia de espacio y tiempo. Son algo fuera de contexto. Solamente se les asigna significado cuando se les asocia a algo.

1.2.3 Información

En un estudio realizado por Pablo Murray este dice que la información es la agrupación de datos organizados y presentados en un contexto. (Murray, 2003)

Para Sergio D' Ambrosio la información "es un conjunto de datos significativos y pertinentes que describen sucesos o entidades." (Heber, 2009)

El término información, según Verónica Sorberamurina, "alude a un conjunto de datos organizados de manera tal que portan o arrojan un significado." (Heber, 2009)

La información es un conjunto de datos asociados a un contexto, además de ser un elemento fundamental en el proceso de comunicación ya que va a suministrar o arrojar algún significado.

1.2.4 Gestión de la Información

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. (Bartle, 2009)

Gloria Donjuán Dante define gestión de la información al proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económico, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. (Dante, 2010)

Gilberto Sotolongo expresa que "la gestión de la información se ocupa de los resultados finales, no sólo de citas y localizaciones". Fernández-Molina lo corrobora cuando afirma que los profesionales de la información y sus instituciones son un factor indispensable para la permanencia de la organización en el mercado. (Dante, 2010)

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades.

1.2.5 Plataforma

Plataforma proviene del francés *plate-forme*. Entre sus diversos significados, el más usual hace referencia a un suelo superior o tablero horizontal que funciona como soporte de personas o cosas. En informática, una plataforma es el hardware sobre el cual puede ejecutarse o desarrollarse un software.

1.2.6 Televisión

La televisión es un sistema para la transmisión y recepción de imágenes en movimiento y sonido a distancia. Dicha transmisión puede ser efectuada mediante ondas de radio o por redes especializadas de televisión por cable. El receptor de las señales es el televisor. (Pérez, 2007)

Las primeras transmisiones de televisión no incluían el sonido. La idea de introducir en una misma emisión el vídeo y el audio se fue incorporando con posterioridad, planteándose dos posibilidades. En la primera, la televisión estaba destinada a complementar los programas de radio, transmitiendo imágenes de forma auxiliar. La otra entendía la transmisión de imágenes como aspecto principal junto con el sonido y la relacionaba con la transmisión de películas. (Pérez, 2007)

1.3 Descripción actual de los procesos del negocio.

1.3.1 Procesos de gestión de la plataforma de televisión informativa PRIMICIA.

PRIMICIA es una plataforma de televisión informativa, creada en la Universidad de las Ciencias Informáticas. Se ha conceptualizado, desarrollado y soportado en software libre contribuyendo así a la soberanía tecnológica del país. Dicha plataforma no interactúa con ningún otro software externo. Esta integral solución provee un canal para la transmisión constante de disímiles formatos tales como: imagen, video, audio y texto, los cuales se integran y conforman las noticias con una información editada de manera atractiva.

En esta nueva versión la plataforma se estructura en 3 subsistemas que tienen una estrecha relación entre sí brindando de esta manera un resultado final acorde a las necesidades de los usuarios.

El subsistema de Administración es el encargado de administrar todo el canal, así como la gestión de las informaciones y recursos multimedia que son transmitidos, mientras que en el subsistema de Transmisión se trabaja en la visualización de las noticias además de otros materiales publicados. (Benítez, 2009)

El tercer subsistema es el de Configuración, el cual se encarga fundamentalmente de gestionar varios procesos dentro de la plataforma de manera que se ajusten a las peticiones requeridas por los usuarios.

El canal de información soportado por la plataforma PRIMICIA, en general, está diseñado para transmitir automáticamente las noticias integrando textos combinados con imágenes y videos. Durante las emisiones se permite la reproducción de un fondo musical, el cual puede ser predeterminado en dependencia de la noticia que se estará mostrando, además, durante la transmisión de las noticias se hace posible utilizar las infocintas o cintillos informativos, los cuales permiten que se adelanten temas breves, información promocional o relevante. A continuación se realiza una breve descripción de varios canales que fueron desarrollados en la versión 1.0 de la plataforma PRIMICIA.

Señal ACN

Señal ACN es un canal de televisión informativa soportado en la plataforma PRIMICIA y que transmite la ACN (Agencia Cubana de Noticias). Este software cuenta con gran aceptación por parte de los colaboradores cubanos en el extranjero y juega un papel fundamental en la transmisión de la verdad cubana hacia el exterior. Además ha logrado ser un apoyo fundamental a la transmisión de la prensa escrita cubana a los llamados puntos ciegos de la geografía del país.

Señal 3

Este es otro canal que ha realizado la UCI destinado a toda la comunidad universitaria. Su principal objetivo es mantener informados a estudiantes, trabajadores y profesores de dicho centro escolar. Este canal fue el precursor de lo que es hoy PRIMICIA. Actualmente es de mucha utilidad para quienes lo disfrutaban día a día, por la razón de permitir mantenerse informados del acontecer nacional e internacional.

1.4 Otras soluciones existentes en el mundo.

Actualmente en el mundo existen en el campo de la televisión varias soluciones que garantizan la automatización de los procesos de producción y transmisión, entre las más destacadas se pueden encontrar: VSNMatic, VEAS, TVMaker, DV Lince y Videomat. Todos estos sistemas automatizan el proceso de transmisión, siendo posible tener el control total de las líneas de exterior y entradas de satélites, con grabación automática, titulación y gráficos, control de grabadoras de cintas de video, reproducción de videos desde disco duro en formatos como MPEG-2, AVI, entre otros. (Albuquerque, 2007)

Todos estos sistemas, al igual que PRIMICIA, se integran con otros módulos para añadir prestaciones inteligentes de una manera modular y escalable, permiten además publicar bloques noticiosos, los usuarios pueden poner materiales en él si tiene permisos para ello, además de permitir la emisión de reportes. También tiene integración con bases de datos. Como se puede ver cada uno de estos sistemas tienen características muy similares que van encaminadas a automatizar emisoras de televisión convencionales mediante la utilización e integración de varios sistemas que se encargan de los diferentes procesos de negocio, olvidando de esta manera entornos cerrados de televisión que necesitan de funcionalidades más específicas.

1.5 Introducción a las Bases de Datos (BD).

En la sociedad actual de la información, los datos constituyen la parte más esencial de cualquier sistema que se desarrolle, estos son procesados, organizados y gestionados en el computador y la

herramienta que se encarga precisamente de estos procesos es la base de datos, la cual se ha convertido actualmente en uno de los principales instrumentos más difundidos y utilizados en el mundo por tener la capacidad de almacenamiento y recuperación de la información.

El término base de datos es bastante utilizado por los desarrolladores de software y los diseñadores de bases de datos y para poder abarcar dicho término se hace imprescindible referenciar algunos de sus principales conceptos:

- ✚ “Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, una BD puede considerarse una colección de datos variables en el tiempo.” (Mato, 2005)
- ✚ “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.” (Date, 2003)
- ✚ Según Date¹ un sistema de base de datos es:

“Básicamente un sistema computarizado para llevar registros. Se considera a la propia base de datos como una especie de armario electrónico para archivar, es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.”

De todo esto se puede deducir que una base de datos no es más que un conjunto de datos coleccionados con características similares y bien estructuradas, los cuales son almacenados y explotados por los sistemas de información de cualquier empresa o negocio.

1.5.1 Organización de las base de datos.

En cuanto a la organización de las bases de datos se puede decir que se estas se organizan por registros, campos, y archivos. Un *registro* no es más que un sistema completo de campos; los *campos* son la pieza única de información; y un *archivo* es una colección de registros. (García, 2010)

Las bases de datos pueden clasificarse de varias formas, de acuerdo al criterio elegido para su clasificación, que puede ser según la variabilidad de los datos almacenados o según el contenido.

1.5.2 Clasificación de las bases de datos.

¹ C.J. Date es autor, conferencista, investigador y consultor independiente, especializado en la tecnología de bases de datos. Reside en Healdsburg, California.

1.5.2.1 Según la variabilidad de los datos almacenados.

Bases de datos estáticas:

Presentan la característica de ser relativamente invariables, además este tipo de bases de datos son de solo lectura, las cuales se utilizan fundamentalmente para almacenar datos históricos que pueden ser utilizados para realizar proyecciones y tomar decisiones. (Santos, 2009)

Son aquellas bases de datos en las que no se puede modificar la información que está guardada, es decir, solo se puede consultar.

Bases de datos dinámicas:

Estas bases de datos tienen la característica que la información almacenada se modifica con el tiempo, lo cual permite varias operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. (Santos, 2009)

En sentido general, son aquellas bases de datos en las que la información almacenada está en constante cambio.

1.5.2.2 Según el contenido que almacenan.

Bases de datos bibliográficas:

Este tipo de bases de datos contienen un "representante" de la fuente primaria, que permite localizarla. Un típico registro de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no se estaría en presencia de una BD a texto completo. (Santos, 2009)

En general, son utilizadas para guardar las referencias a un contenido dado.

Bases de datos de texto completo:

Estas bases de datos almacenan toda la información de las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas, nunca fragmentos ni referencias. (Santos, 2009)

Directorios:

Este es el caso de directorios digitales donde se especifican los números de los usuarios, ya sean estatales o privados donde se puede encontrar el nombre de dicho usuario y la dirección del mismo. Un ejemplo son las guías telefónicas en formato digital.

Por las características que presenta la concepción del sistema PRIMICIA y teniendo en cuenta la actualización de la información que ofrecen las bases de datos dinámicas se muestran como la opción más viable para la solución de la presente base de datos a desarrollar, ya que se hace necesaria la interactividad de forma dinámica de la información.

1.5.3 Modelos de Bases de Datos.

Además de la clasificación por la función de la BD, también se le puede clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una descripción de lo que se conoce como contenedor de datos en donde se guarda la información, así como los métodos para almacenar y recuperar información de esos contenedores.

Algunos de los modelos utilizados con frecuencia son:

Base de datos jerárquica.

Entre los primeros modelos de datos que surgieron en los Sistemas Gestores de Base de Datos comerciales se encuentran las estructuras en árbol, propias de los productos jerárquicos. Los modelos de datos jerárquicos surgieron alrededor de los años 60. Como su nombre lo indican estas bases de datos almacenan la información en una estructura jerárquica. (Ruiz, 2003)

El modelo jerárquico fue desarrollado para permitir la representación de aquellas situaciones de la vida real en las que predominan las relaciones de tipo 1: N (de 1 a muchos). Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación PADRE/HIJO, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo. (Ruiz, 2003)

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Base de datos en red.

Este modelo se diferencia del jerárquico en la modificación del concepto de nodo, donde se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una eficiente solución al problema de redundancia de datos; pero aún así, la dificultad que significa administrar la información en una BD de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. (Santos, 2009)

Base de datos relacional.

El modelo relacional fue creado por Edgar Frank Codd² a principios de los años 70. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".

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, por lo cual tiene la ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. (Hansen, 2005)

Cada relación tiene un atributo o un conjunto de estos, conocidos como clave primaria que identifican a la fila. Las referencias de interrelaciones son manipuladas a través de llaves foráneas, que son punteros lógicos de una fila de una relación a una fila de otra relación. Para estar seguro de la validación de los datos en una base de datos relacional, Codd formuló las reglas de integridad de la entidad y las reglas de integridad referencial. Estas reglas establecen que el valor de una clave primaria no puede ser nulo, y que el valor de una clave externa debe corresponder con el valor actual de la clave primaria de la otra relación a la que se hace referencia. (Hansen, 2005)

Los sistemas relacionales son de gran importancia ya que ofrecen varios tipos de procesos de datos, como: simplicidad, facilidad de uso para el usuario final, generalidad y consultas de información especificadas de una forma muy sencilla.

Base de datos orientado a objetos.

Este modelo trata de almacenar en la BD los objetos completos (estado y comportamiento). Una BD orientada a objetos no es más que una base de datos que incorpora todos los conceptos importantes del paradigma de objetos como son:

- Encapsulación (propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos).

² **E.F. Codd:** Gran matemático y científico de la computación. Es conocido por sus aportes en las teorías de las bases de datos relacionales.

- 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).

En las bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. (Santos, 2009)

Después de analizar detenidamente cada uno de los modelos de datos existentes se decide utilizar para la base de datos de PRIMICIA el modelo de base de datos relacional ya que este permite evitar la duplicidad de registros, además de favorecer la normalización.

1.6 Sistemas Gestores de Bases de Datos (SGBD)

Los Sistemas Gestores de Bases de Datos, en lo adelante SGBD, aparecen como una capa intermedia capaz de establecer la comunicación entre los programas que utiliza el usuario final y el sistema operativo. El principal objetivo de estos sistemas gestores es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera eficiente. (Zambrano, 2008)

Por tanto, el SGBD es una aplicación que permite a los usuarios crear y mantener la BD, proporcionando un acceso controlado a la misma. Dentro de los servicios que estos ofrecen se encuentran: (Zambrano, 2008)

- ✚ Definición y creación de la base de datos.
- ✚ Manipulación de los datos realizando consultas, inserciones y actualizaciones.
- ✚ Acceso controlado a los datos mediante mecanismos de seguridad de acceso a los usuarios.
- ✚ Mantener la integridad de los datos.
- ✚ Mecanismos de copias de respaldo y recuperación para establecer la información en caso de algún fallo en el sistema.

1.6.1 Objetivos de los SGBD.

✚ Independencia de los datos:

Consiste en hacer que los programas no dependan de la estructura de los datos que debe utilizar, es decir, que se pueda cambiar esa estructura sin tener que cambiar los programas relacionados con ella.

Se han definido dos tipos de independencia:

- **Independencia lógica:** consiste en poder cambiar la definición conceptual del sistema de información sin que ello suponga una modificación de los programas existentes.
- **Independencia física:** consiste en poder modificar parámetros de cómo está almacenada físicamente la información, sin que ello suponga una modificación de los programas existentes.

Seguridad e Integridad:

Consiste en lograr que los usuarios no puedan acceder a datos sin autorización. Si se junta toda la información de la empresa en un solo sitio, el SGBD debe tener mecanismos para que cualquier usuario pueda tener acceso únicamente a la información que necesita de cara a la privacidad de esa información, incluso si tiene acceso a una información que se pueda decidir si además de visualizarla puede modificarla (Zambrano, 2008)

Redundancia mínima:

La redundancia consiste en que existan datos idénticos repetidos en varios lugares. Por ejemplo si se guarda la dirección del cliente en la factura y en los datos generales del cliente, se tendrá redundancia, es decir, el mismo dato repetido en varias tablas.

Facilidad de recuperación de la información:

Otro objetivo muy importante de un SGBD es el proporcionar al usuario (y al programador) herramientas potentes de manejo de datos para que pueda de manera sencilla y rápida, obtener toda la información que desea. (Zambrano, 2008)

1.6.2 Arquitectura de los SGBD.

En el año 1975, el comité del Instituto Americano Nacional de Estándar- Normas de Comité de Planeación y Requerimientos (ANSI-SPARC) propuso una arquitectura basada en tres niveles para los sistemas de bases de datos. Esto ayudó de una manera u otra a conseguir varias características importantes que son inherentes a los sistemas de BD, estas son: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos.

Los 3 niveles a los que se hizo referencia anteriormente son:

Nivel interno:

Este nivel es el más cercano al almacenamiento de los datos, es el de la visión del administrador de la BD, ya que es donde se tiene la definición física de los datos, su organización interna, la definición de las reglas de validación, controles de seguridad, etc.

✚ Nivel conceptual:

En este nivel se tiene una visión global de todos los datos que intervienen en la base de datos, pero olvidándose de la estructura interna. Además en este nivel es donde se tiene la definición de qué datos se guardan en la base de datos y las relaciones que los unen entre sí.

✚ Nivel externo:

Este nivel representa la percepción de los datos por cada uno de los usuarios o programadores, el usuario no ve todos los datos que aparecen en el nivel conceptual, sino solo los que le hacen falta. Cada esquema externo es un subconjunto del esquema conceptual, proporcionando una visión parcial del mismo, lo que permite proteger la información reservada a unos pocos usuarios. En la siguiente figura se muestra una vista general de los niveles.

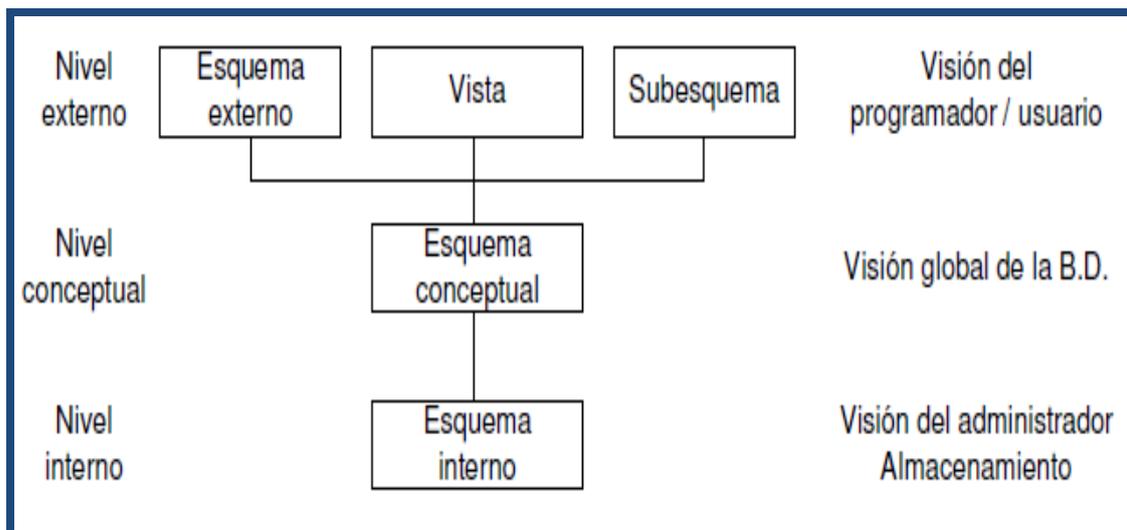


Figura 1: Niveles de una base de datos.

1.6.3 Principales gestores de bases de datos.

Los SGBD, como se había tratado anteriormente, son un software específico dedicado a servir de interfaz entre el usuario, la base de datos y las aplicaciones que la utilizan. El desarrollo que han experimentado los mismos ha sido extraordinario por la facilidad de manejar la información, suministrando a los usuarios las herramientas necesarias que le permitan manipular de forma abstracta los datos con la existencia de múltiples entornos de desarrollo y soluciones informáticas.

Capítulo 1: Fundamentación teórica.

Dentro de los gestores de base de datos más potentes de software propietario se encuentran: Oracle y Microsoft SQL Server; dentro de los de software libre está PostgreSQL y en el caso de licencia dual o doble licenciamiento se encuentra MySQL. En la siguiente tabla se muestra un pequeño cuadro comparativo entre estos gestores:

Tabla 1: Comparación entre sistemas gestores de base de datos. (García, 2010)

| Criterios de comparación | Oracle | MySQL | SQL Server | PostgreSQL |
|--------------------------|------------------------------------|---|----------------|---|
| Sistemas Operativos | Windows, MAC OS X, Linux, GNU/UNIX | Windows, MAC OS X, BSD, Linux, GNU/UNIX | Windows | Windows, MAC OS X, BSD, Linux, GNU/UNIX |
| Consumo de recursos | Gran cantidad de recursos | Pocos recursos | Pocos recursos | Pocos recursos |
| Tipo de licenciamiento | Propietario | GPL o Propietario | Propietario | BSD |
| Integridad referencial | Sí | No | Sí | Sí |

1.6.4 Sistema gestor de base de datos seleccionado.

Luego de realizar una búsqueda de las características de los gestores de BD anteriormente mencionados se conoce acerca de las limitantes que tiene cada uno de ellos. En el caso de Oracle, constituye un producto donde sus costos por soporte técnico son elevadísimos, además de constituir un software propietario y su licencia ser de las más caras en el mundo del software, por lo general solo se ve en empresas muy grandes y multinacionales. Por su parte SQL Server tiene licenciamiento propietario y su licencia de uso tiene valores muy altos. Además solo puede ser utilizado sobre los Sistemas Operativos Windows. Otro de los gestores es MySQL, este presenta como desventaja que su escalabilidad no es buena, por eso no es recomendable usarlo en base de datos muy grandes. También puede provocar problemas de integridad y la información se puede dañar en entornos donde se gestione mucha información simultáneamente y altas concurrencias en las modificaciones de datos. El último gestor que se estudió fue el PostgreSQL, este es el líder entre los gestores de bases de datos de código abierto y cuenta con una comunidad global de miles de contribuyentes, usuarios, docenas de compañías y organizaciones. Además es el más usado en la Universidad de las Ciencias Informáticas, donde cuenta también con una comunidad de desarrollo. Actualmente este se encuentra seleccionado en el documento de Arquitectura del proyecto. A partir de todo lo explicado se puede definir PostgreSQL

como gestor para la base de datos de PRIMICIA versión 2.0, a continuación se presentan algunas de sus características más relevantes.

PostgreSQL.

El proyecto PostgreSQL tiene más de 20 años de ingeniería, desde sus inicios en la Universidad de Berkeley, California, con un ritmo de desarrollo actual sin precedentes. PostgreSQL se distribuye bajo la "Licencia PostgreSQL", esta es similar a la licencia BSD, que solo requiere que el código fuente mantenga información sobre el copyright y la licencia. Esta licencia certificada por OSI es reconocida por ser flexible y amigable para los negocios, puesto que no restringe el uso de PostgreSQL con aplicaciones comerciales y propietarias. (PostgreSQL, 2011)

El conjunto de funcionalidades de PostgreSQL no solo es comparable a los mejores sistemas gestores de datos propietarios, sino que lo superan en características avanzadas, extensibilidad, seguridad y estabilidad.

Como principales características de estos gestores se encuentran: que corre en casi todos los sistemas operativos: Linux, Unix, Mac OS, Beos, Windows, entre otros; cuenta con una documentación muy bien organizada, pública, libre y con comentarios de los propios usuarios; también presenta soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl y Python, así como soporte de protocolo de comunicación encriptado por SSL. Además dispone de extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales y minería de datos. (Quiñones, 2009)

PostgreSQL versión 9.0

Las versiones anteriores de PostgreSQL cuentan con las mismas características anteriormente explicadas. Esta nueva versión de PostgreSQL incorpora avanzadas características en materia de seguridad, soporte de aplicaciones, seguimiento y control, rendimiento y almacenamiento de datos especializados.

PostgreSQL 9.0 incluye más de una docena de cambios importantes que mejorarán muchos aspectos de diseño y rendimiento de aplicaciones de base de datos, entre los que se encuentran: triggers condicionales y por columna, actualización in-situ desde las versiones 8.3 y 8.4, restricciones de unicidad postergables, y paso de mensajes de alto rendimiento. También se mejora el soporte para datos del tipo llave-valor y remoción automática de joins lo cual permitirá optimizar consultas generadas por sistemas ORM. (PostgreSQL, 2011)

La característica más relevante que aparece en esta nueva versión de PostgreSQL es la incorporación de la replicación de la base de datos mediante Streaming Replication y Hot Standby. Esto trae como

ventaja que no sea necesario utilizar una herramienta externa de Replicación, además de ser fácil y cómoda la configuración de esta funcionalidad. En epígrafes posteriores se abordarán las características de este tipo de réplica.

1.7 Herramienta de administración y desarrollo para PostgreSQL.

Para la administración y desarrollo con PostgreSQL se utiliza la herramienta pgAdmin III. Esta es una aplicación gráfica para gestionar PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ y utiliza la librería multiplataforma wxWidgets, lo que permite que se pueda usar en sistemas operativos como Linux, Mac OS X y Windows.

Esta herramienta está diseñada para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La conexión al servidor puede hacerse mediante el protocolo TCP/IP.

1.8 UML como soporte para el modelado de la base de datos.

UML es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representan la arquitectura del proyecto. (Casasola, 2008)

Hoy en día, el Lenguaje para Modelado Unificado (UML) está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante este se puede establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. (Casasola, 2008)

Este lenguaje de modelado es el más conocido y utilizado en la actualidad. Algunas de las características por las que es seleccionado son: el uso de lenguajes visuales facilitando su asimilación y entendimiento por parte del equipo de desarrollo; ofrece un estándar para describir un plano del sistema (modelo); el tiempo invertido en el desarrollo de la arquitectura se minimiza y la detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración.

1.9 Visual Paradigm como Herramienta CASE utilizada.

Las herramientas CASE (Computer Aided Software Engineering) son de gran ayuda para el desarrollo de un software ya que permiten el diseño del mismo antes de su codificación. Esto se realiza mediante diagramas que permiten que todo el equipo de desarrollo comprenda lo que se quiere hacer realmente y que el trabajo que se realice sea de forma homogénea.

Algunas de estas herramientas tienen un valor económico muy alto y requieren altos costos de entrenamiento para el personal. Entre las principales herramientas se encuentran: ERwin, Visual Paradigm y Embarcadero ER / Studio.

ERwin es una herramienta de modelado 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, pero tiene como inconveniente que no soporta el gestor de base de datos seleccionado (PostgreSQL). Otra de las herramientas existentes es Embarcadero ER/ Studio. Con esta, el modelado de datos es fácil para el diseño y construcción de bases de datos a nivel físico y lógico. Además permite la construcción automática de base de datos, así como la ingeniería inversa de base de datos. No es seleccionada porque al igual que ERwin, no permite soporte para el gestor PostgreSQL. A diferencia de esta herramienta, Visual Paradigm sí soporta dicho gestor, se encuentra además descrito en el documento de Arquitectura del proyecto PRIMICIA y es la que más se ajusta a las necesidades por las características que se reflejarán a continuación.

Visual Paradigm

Esta herramienta CASE utiliza UML como lenguaje de modelado y soporta además el diagrama entidad-relación. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal, que permite escoger la más idónea para el usuario según su necesidad.

Dicha herramienta se distribuye bajo licencia gratuita y comercial, entre las principales características que presenta Visual Paradigm están: producto de calidad, multiplataforma, soporta aplicaciones web, varios idiomas, fácil de instalar y actualizar, compatibilidad entre ediciones, soporta a miles de usuarios trabajando sobre el mismo proyecto y permite que cada uno vea los cambios realizados en tiempo real. (Hernandis, 2009)

Posibilita también la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, ingeniería inversa de bases de datos. Este software está disponible para Win98; WinME; WinNT 4.x; Windows2000; WinXP; Windows2003; Mac OS; GNU/Linux. (Hernandis, 2009)

1.10 RUP como metodología para el desarrollo de la solución propuesta.

Capítulo 1: Fundamentación teórica.

La metodología del Proceso Unificado de Software (RUP) es un “amplio proceso de ingeniería de software. Este cuenta con un enfoque disciplinado hacia las industrias probadas para el diseño de software y sistemas dentro de una organización de desarrollo”. Se mantiene un control de la gestión de proyectos eficaz y de alta calidad de la producción de software. (Singha, 2007)

RUP agrupa las actividades en 9 flujos de trabajo principales. Los 6 primeros flujos son conocidos como flujos de ingeniería y los tres últimos como de apoyo. EL ciclo de desarrollo en la vida de un software se divide en 4 fases. En el siguiente gráfico se puede apreciar:

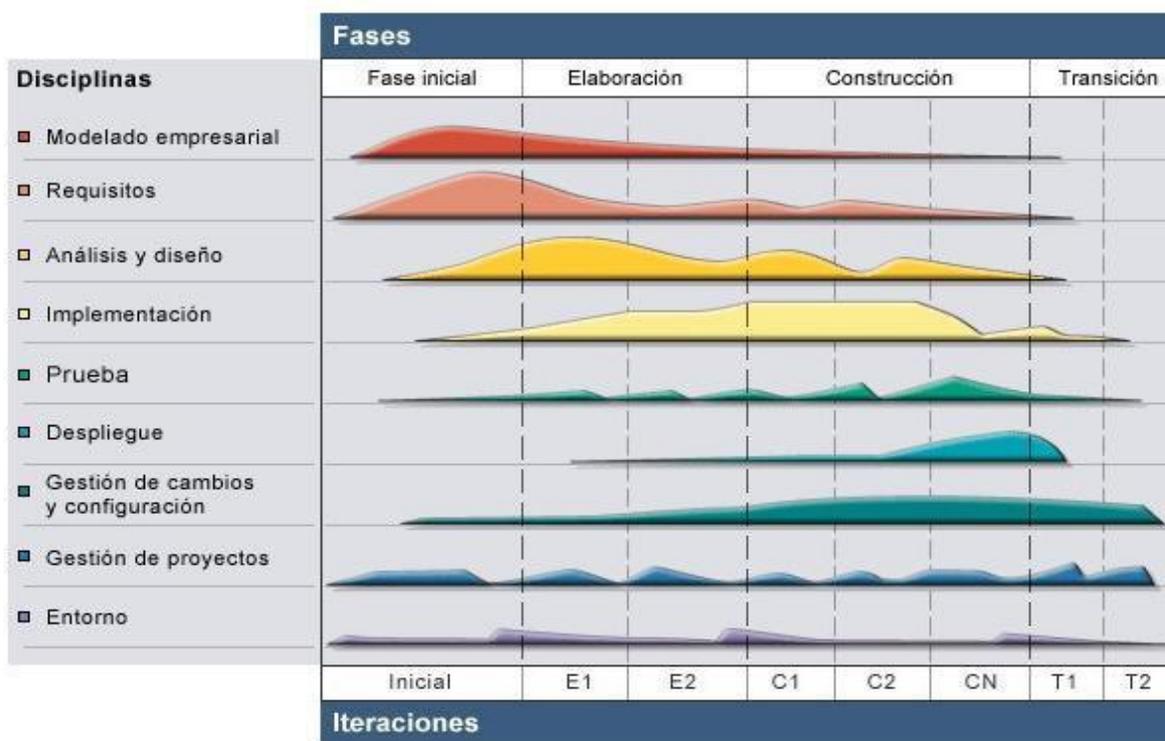


Figura 2: Fases e Iteraciones de la Metodología RUP. (González, 2009)

RUP como metodología presenta varias características. Estas son:

- ✚ **Dirigido por casos de usos:** basándose en ellos se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada caso de uso. (Letelier, 2008)
- ✚ **Iterativo e incremental:** se basa en un proceso en donde el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. (Letelier, 2008)

✚ Centrado en la arquitectura: existe una interacción entre los casos de uso y la arquitectura, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como casos de uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software. (Letelier, 2008)

Según la metodología de RUP, el rol de diseñador de base de datos debe desarrollar una serie de artefactos y actividades para garantizar el cumplimiento satisfactorio de su labor, entre las que se encuentran: identificar las clases persistentes, elaborar el diagrama de clases persistentes, crear clave primaria y restricciones de integridad, definir las reglas de integridad referencial y de la información y por último normalizar el diseño de la BD para su optimización.

El mismo debe tener un gran conocimiento sobre el modelado de datos, el análisis y diseño de técnicas orientada a objetos así como la administración de la BD. El artefacto que se obtiene como resultado de dichas actividades es el Modelo de Datos. Este artefacto describe las representaciones lógicas y físicas de los datos persistentes utilizados por la aplicación. Puede ser creado a partir de un conjunto de clases del diseño, del modelo de diseño, a partir de un modelo entidad-relación creado por el diseñador de BD o mediante ingeniería inversa a partir de una BD existente.

1.11 Replicación de datos.

La réplica de datos es una técnica que permite copiar y distribuir idénticamente las tablas de una base de datos en múltiples bases de datos ubicadas en distintos nodos de la red. (Castro, 2008)

La réplica garantiza además que en caso de fallo del servidor de BD por alguna razón, la aplicación en vez de caerse, obtenga de un servidor alternativo la misma información. Actualmente en la plataforma esta técnica no estaba implementada y considerando lo dicho anteriormente es que la nueva versión contará con la replicación.

1.11.1 Tipos de entornos de réplicas

Existen fundamentalmente 2 tipos de entornos de réplicas. A continuación se describen las características de cada uno:

✚ Maestro-Esclavo (master-slave):

Conocido también como de solo lectura, el cual permite a un solo maestro recibir consultas de lectura/escritura, mientras los esclavos solo pueden aceptar consultas de lectura. (Fajardo, 2007)

✚ Multi-Maestro (multi-master):

Llamada también par-par. Cada sitio es un entorno de la réplica multi-maestro y cada uno de ellos se comunica con otros sitios maestros. Esta capacidad tiene también un severo impacto en el desempeño debido a la necesidad de sincronizar los cambios entre los servidores. (Fajardo, 2007)

1.11.2 Modelo de distribución de datos.

Los modelos de distribución de datos son fundamentalmente aplicados a cada uno de los entornos antes vistos y estos se clasifican en:

✚ Sincrónico:

También conocida como la réplica en tiempo real, aplica cualquier cambio o ejecuta cualquier procedimiento reproducido en todos los sitios que participan en el ambiente de réplica como parte de una sola transacción. Si el procedimiento falla en cualquier sitio, entonces la transacción entera se anula. La réplica sincrónica asegura la consistencia de datos en todos los sitios en tiempo real. (Fajardo, 2007)

✚ Asíncrono:

A menudo alguna funcionalidad de almacenamiento envía, captura cualquier cambio local, lo almacena en una cola, y en intervalos regulares propaga y aplica cambios en sitios remotos. En este modelo, hay un lapso de tiempo para que los datos alcancen la analogía de los datos. (Fajardo, 2007)

1.11.3 Técnicas de replicación

✚ **Sincrónica Confirmación en dos fases (Two Phase Commit):** Surgió cerca del año 1985. Esta técnica permite la sincronización de los datos distribuidos. Cada transacción es realizada solamente si todos los nodos replicados están conectados entre sí y listos para recibir la información, en caso contrario la misma es anulada, es decir no se realiza. (Fajardo, 2007)

✚ **Asincrónica Descarga y Recarga (Dump and Reload):** Consiste en hacer un volcado de los datos, copiar la salva para un dispositivo de almacenamiento, para luego distribuirlo hacia los demás servidores. Presenta como desventaja que en muchos casos pueden existir la desactualización de la información y además este proceso se realiza de forma manual. (Fajardo, 2007)

A continuación se describen 2 técnicas para lograr la replicación asincrónica:

- Instantánea (Snapshots): Consiste en realizar una instantánea de una tabla dada en un momento determinado y distribuir la misma por todas las réplicas. A pesar de que es mucho más avanzada que la técnica de Descarga y Recarga tiene como inconveniente de que las tablas esclavas son de solo lectura, por lo tanto se recomienda utilizar cuando la mayoría de los datos no cambian con frecuencia; se replican pequeñas cantidades de datos; los sitios con frecuencia están desconectados y es aceptable un período de latencia largo.
- Disparadores (Triggers): son programas a los que se les da un nombre de objeto en la base de datos, se asocia con una tabla determinada, y se activa cuando ocurre un cierto tipo de evento en esa tabla, como una inserción, actualización o borrado. Son al igual que las instantáneas, mecanismos asíncronos que proporcionan a la base de datos una forma de replicación de los datos. (Fajardo, 2007)

1.12 Herramienta de replicación de la solución propuesta.

En la actualidad existen varias herramientas destinadas a realizar la réplica de datos, enfocadas al sistema gestor de base de datos seleccionado: PostgreSQL. Estas aplicaciones se escogen de acuerdo al entorno de réplica donde pueden ser aplicadas.

Para el entorno “Maestro-Eslavo” son usadas disímiles herramientas. Estas son: Slony I, PyReplica, Londiste, Streaming Replication y Hot Standby, entre otras.

Slony I constituye un replicador asíncrono donde una única base de datos Maestra, replica a múltiples esclavos, soportando la réplica en cascada y permitiendo a un esclavo ser a su vez maestro para otro servidor. Presenta un esquema avanzado de configuración. Este permite seleccionar las tablas a replicar, así como replicar en múltiples direcciones. Tiene como principales inconvenientes que los datos no se replican instantáneamente al esclavo, sino que se almacenan en una tabla mediante triggers o disparadores y desde ahí los copia a la réplica e intenta ejecutarlos. Este proceso puede llegar a tardar de 8 a 9 segundos. Otras de las desventajas es que soporta pocos nodos de replicación, no detecta errores ni es capaz de recuperarse de ellos y además requiere de una red de comunicaciones altamente confiable y de alta tasa de transferencia. Por su parte Londiste es muy similar a Slony I, aunque más moderno, modular y menos susceptible a fallas. Como desventajas presenta que no permite la replicación en cascada, ni la ejecución de scripts. En el caso de PyReplica es un replicador asíncrono implementado en Python y multiplataforma (Windows y Linux). Al igual que Slony I y otros replicadores basados en disparadores, no soporta la replicación de DDL automática pero puede usarse para propagar órdenes DDL a varios servidores. Además no permite la replicación de

secuencias. El tiempo que demora en ejecutarse el disparador para permitir la replicación es de 15,1 segundos. Por último tenemos a Streaming Replication y Hot Standby, ambas son funcionalidades que incluye la nueva versión de PostgreSQL, la 9.0. Esta a diferencia de las demás herramientas es una replicación integrada donde se permite realizar consultas en las réplicas, así como detenerlas o cancelarlas por retraso. Transmite la replicación instantáneamente y también replica DDL, esta es una de las mejoras que posee.

Después de analizar cada una de las posibles herramientas se muestra como vía más factible, utilizar las funcionalidades Streaming Replication y Hot Standby que trae implementadas la nueva versión del propio gestor que se utiliza PostgreSQL debido a todas las características que posee y que a continuación se describen.

Streaming Replication y Hot Standby

Como se había mencionado anteriormente dos de las características más importantes incluidas en la versión 9.0 de PostgreSQL son *Hot Standby (HS)* y *Streaming Replication (SR)*. Ambas características implementan en el núcleo de PostgreSQL lo necesario para instalar un sistema de replicación asincrónica maestro-esclavo (master-slave) en el que los nodos esclavos se pueden utilizar para realizar consultas de solo lectura. Un sistema de replicación de estas características se podrá usar tanto para añadir redundancia a nuestras bases de datos, como para descargar de trabajo a nuestro servidor principal en lo referente a consultas de solo lectura. (PostgreSQL, 2011)

En general, utilizando SR el sistema de base de datos guardará continuamente datos (cuando sea posible y sin llevar un registro). Esta característica será especialmente útil en entornos en los que la disponibilidad de la información sea crítica.

1.13 Conclusiones parciales

Después de haber realizado un primer acercamiento al objeto de estudio y estudiar las características y elementos con que debe cumplir la base de datos para el tipo de producto PRIMICIA, se ha conocido detalladamente las ventajas y desventajas que brindan las herramientas y modelos seleccionados, permitiendo esto que se haga un buen uso de los mismos para todo el proceso de diseño e implementación de la base de datos. Además con todo el estudio previamente realizado se está en condiciones de comenzar a diseñar la base de datos.

Capítulo 2: Descripción y análisis de la solución propuesta.

2.1 Introducción

En el presente capítulo se realiza una descripción acerca de los principales procesos que se llevan a cabo para el diseño de la base de datos. Durante éste se muestra una breve explicación acerca de la integración de los módulos de la plataforma. Se exponen además los principales requisitos funcionales y no funcionales del sistema en cuestión, el diagrama de clases persistentes y el diagrama entidad-relación.

2.2 Integración de la solución con los módulos existentes.

La versión 2.0 de la plataforma PRIMICIA se estructura en 3 subsistemas y estos a su vez se encuentran divididos en varios módulos: Gestión de Medias, Seguridad, Editorial, Redacción, Reporte, Errores y Gestión de la Señal.

Todos estos módulos se encuentran relacionados entre sí. Acceden a la base de datos física a través de clases de acceso a datos generadas automáticamente por Symfony, framework de desarrollo seleccionado por el arquitecto del proyecto y utilizado en el mismo, el cual posibilita separar el acceso a la base de datos de la lógica de la aplicación en sí.

2.3 Requisitos del sistema.

La definición de las necesidades del software es un proceso complicado, ya que hay que identificar todos los requisitos que el sistema debe cumplir para lograr satisfacer las necesidades de los usuarios finales. El tratamiento de requisitos es aquel proceso donde se especifican y validan las funcionalidades que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar.

2.3.1 Requisitos funcionales.

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. (Sommerville, 2005)

Todos estos requisitos impactan de una manera u otra en las tablas de la base de datos, pues esta es la que se encarga de ofrecer el almacenamiento de los datos que se necesitan para poder realizar los requisitos. A continuación se muestran los requisitos funcionales de mayor impacto en la base de datos.

Capítulo 2: Descripción y análisis de la solución propuesta

Estos se encuentran descritos en el documento Especificación de Requisitos que está ubicado en el expediente de proyecto de PRIMICIA.

RF 1: Autenticar usuarios: el sistema debe permitir que el usuario para su acceso al sistema deberá autenticarse, acción que permitirá según el rol las credenciales correspondientes para el trabajo con el sistema.

RF 2: Insertar usuarios: el sistema debe permitir que se inserten nuevos usuarios.

RF 3: Modificar usuarios: el sistema debe permitir que se realicen cambios en los permisos de los usuarios.

RF 4: Eliminar usuarios: el sistema debe permitir que se eliminen usuarios.

RF 5: Ordenar secciones temáticas: el sistema debe permitir que se ordenen las secciones temáticas para su visualización, horario en que serán mostradas las informaciones en el canal, entre otros criterios.

RF 6: Visualizar sección temática: el sistema debe permitir que se habilite o deshabilite la visualización de la sección temática.

RF 7: Insertar sección temática: el sistema debe permitir que se inserten nuevas secciones temáticas.

RF 8: Modificar sección temática: el sistema debe permitir que se modifiquen las secciones temáticas existentes.

RF 9: Eliminar sección temática: el sistema debe permitir que se eliminen secciones temáticas.

RF 10: Publicar noticia: el sistema debe permitir que se muestre un listado de las noticias redactadas organizadas por secciones temáticas. La publicación se debe realizar teniendo en cuenta fecha de inicio y fin de publicación.

RF 11: Mostrar las noticias publicadas: el sistema debe permitir que se muestre un listado de las noticias publicadas organizadas por secciones temáticas y estado de transmisión (espera, transmisión, vencida), para cada noticia se mostrarán las opciones mostrar la noticia, modificar, archivar y eliminar

RF 12: Redactar noticias: el sistema debe permitir que se redacten las noticias teniendo en cuenta los parámetros establecidos: título, sección temática, música de fondo y pantallas por las que se compondrá la noticia.

Capítulo 2: Descripción y análisis de la solución propuesta

RF 13: Mostrar las noticias redactadas: el sistema debe permitir que se listen las noticias redactadas por el usuario autenticado en caso que tenga permisos de redacción y que la noticia no haya sido publicada.

RF 14: Almacenar recursos multimedia: el sistema debe permitir que se almacenen imágenes, videos y música. Para la acción anterior se deben tener en cuenta los siguientes parámetros para cada tipo de recurso:

- Música: país, género y título.
- Videos e imágenes: sección temática, palabras claves, fecha y nombre.

RF 15: Establecer tipo de transmisión: el sistema debe permitir que se establezca si la plataforma va a realizar la transmisión por bloques noticiosos, o secciones temáticas.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares. (Sommerville, 2005)

RNF de Rendimiento:

- El tiempo de respuesta máximo a las peticiones por parte de los usuarios en las consultas más complejas que actualmente se utilizan, no debe excederse de 10 segundos.
- El sistema debe permitir trabajar conectados concurrentemente 100 usuarios como máximo.

RNF de Seguridad:

- La información almacenada o manejada desde la base de datos debe estar protegida de acceso no autorizado y divulgación.
- La base de datos será independiente de la aplicación.
- A los usuarios autorizados se les garantizarán el acceso a la información, los dispositivos o mecanismos utilizados para lograr la seguridad, no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

RNF de Soporte:

- El soporte y/o mantenimiento de la base de datos no debe necesariamente detener el servicio.

RNF de Portabilidad:

Capítulo 2: Descripción y análisis de la solución propuesta

- La base de datos garantizará su correcto funcionamiento en los sistemas operativos GNU/Linux y Windows, por las características que presenta el SGBD utilizado.

RNF Software:

- El servidor de bases de datos debe ser PostgreSQL versión 9.0.
- La aplicación gráfica de desarrollo de PostgreSQL deberá ser pgAdmin III, versión 1.12 o superior.

RNF Hardware:

- El servidor de base de datos debe contener un procesador Pentium 3 (o superior).
- Debe tener como mínimo 1GB de Memoria RAM.
- Debe contener al menos 1 disco duro de 160 GB.

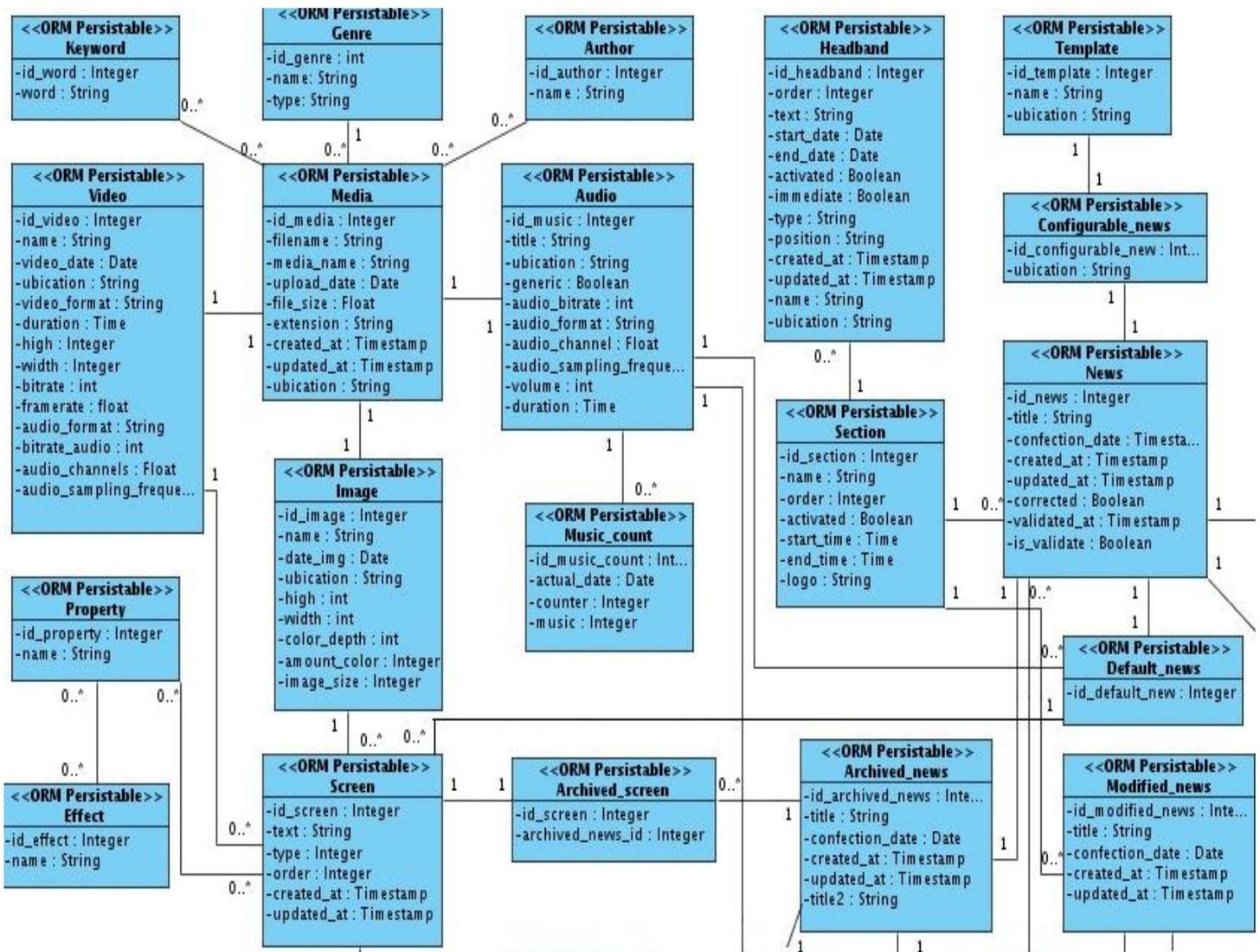
2.6 Diagrama de clases persistentes.

El diagrama del modelo lógico de datos o 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. (Hernández, 2004).

A continuación se muestra el diagrama de clases persistentes de la plataforma de televisión informativa PRIMICIA versión 2.0 donde se representan las clases persistentes en su totalidad.

Nota: Debido al tamaño de la imagen se hace imprescindible dividirla en varias sub-imágenes para una mejor visibilidad. Las relaciones que no se vean se describen debajo de la imagen.

Capítulo 2: Descripción y análisis de la solución propuesta



La tabla **News** tiene una relación (1 a 1) con la tabla **Modified_news**.

La tabla **News** tiene una relación (1 a 1) con la tabla **Published_news**.

La tabla **News** tiene una relación (0..* a 1) con la tabla **Public.sf_guard_user**.

La tabla **Audio** tiene una relación (0..* a 1) con la tabla **Published_default_news**.

La tabla **Screen** tiene una relación (1 a 1) con la tabla **Modified_screen**.

La tabla **Archived_news** tiene una relación (1 a 1) con la tabla **Archived_configurable_news**.

La tabla **Archived_news** tiene una relación (1 a 1) con la tabla **Archived_default_news**.

La tabla **Modified_news** tiene una relación (0..* a 1) con la tabla **Public.sf_guard_user**.

La tabla **Modified_news** tiene una relación (1 a 0..*) con la tabla **Modified_screen**.

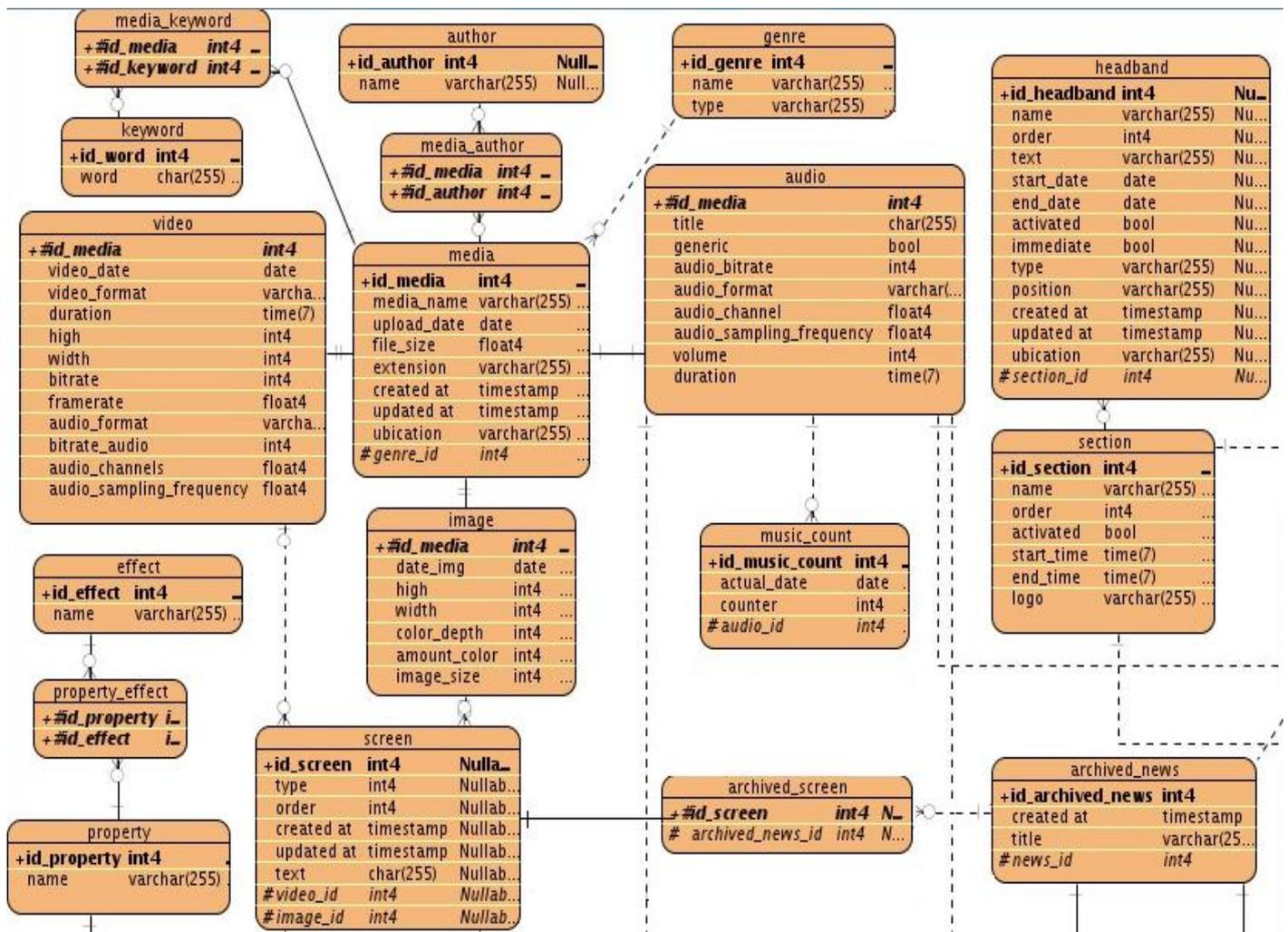
Capítulo 2: Descripción y análisis de la solución propuesta

2.6.1 Descripción de las clases del diagrama de clases persistentes.

Para ver la descripción detallada de las clases que conforman el modelo lógico de la base de datos: [Ver Anexo 1.](#)

2.7 Diseño de la base de datos.

El diseño de bases de datos es el proceso por el que se determina la organización de una base de datos, incluido su estructura, contenido y las aplicaciones que se han de desarrollar. El rendimiento adecuado del sistema se deberá a una correcta selección de las tablas que formarán parte de la base de datos. A continuación se representa el diagrama Entidad- Relación de la base de datos de la versión 2.0 de la plataforma de televisión informativa, en el cual se muestran las tablas existentes, así como las relaciones entre ellas.



Nota: Debido al tamaño de la imagen se hace imprescindible dividirla en varias sub-imágenes para una mejor visibilidad. A continuación se detallan algunas relaciones.

Capítulo 2: Descripción y análisis de la solución propuesta

La tabla **property** tiene relación (1 a 0...*) con la tabla **screen_property**.

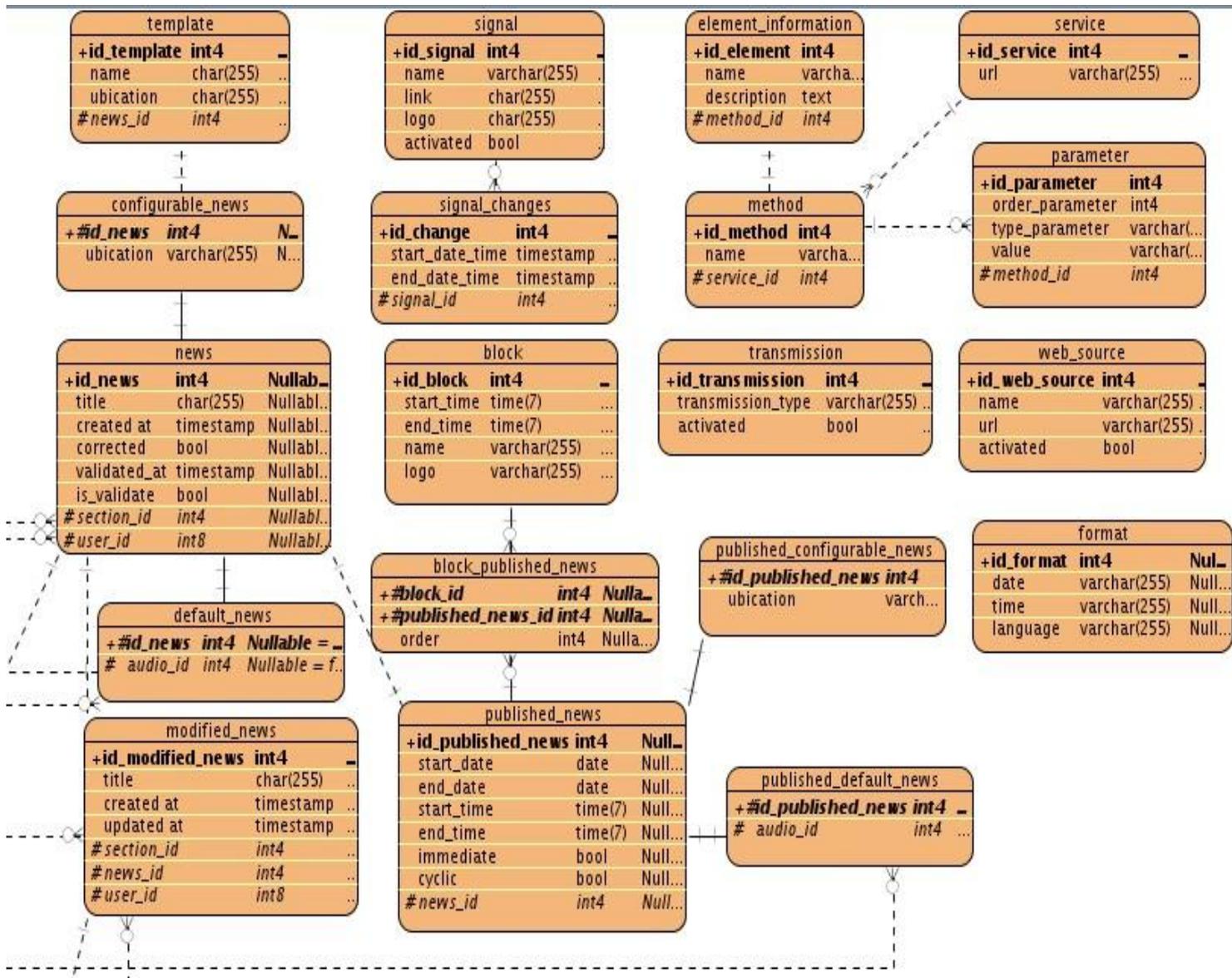
La tabla **screen** tiene relación (1 a 0...*) con la tabla **screen_property**.

La tabla **screen** tiene relación (1 a 1) con la tabla **modified_screen**.

La tabla **audio** tiene relación (1 a 0...*) con las tablas **default_news**, **archived_default_news** y **published_default_news**.

La tabla **section** tiene relación (1 a 0...*) con las tablas **news** y **modified_news**.

La tabla **archived_news** tiene relación (1 a 1) con las tablas **archived_default_news** y **archived_configurable_news**.



Capítulo 2: Descripción y análisis de la solución propuesta

La tabla **modified_news** tiene relación (1 a 0...*) con la tabla **modified_screen**.

La tabla **modified_news** tiene relación (0...* a 1) con las tablas **public.sf_guard_user** y **section**.

La tabla **news** tiene relación (0...* a 1) con la tabla **public.sf_guard_user**.

La tabla **news** tiene relación (1 a 1) con la tabla **archived_news**.

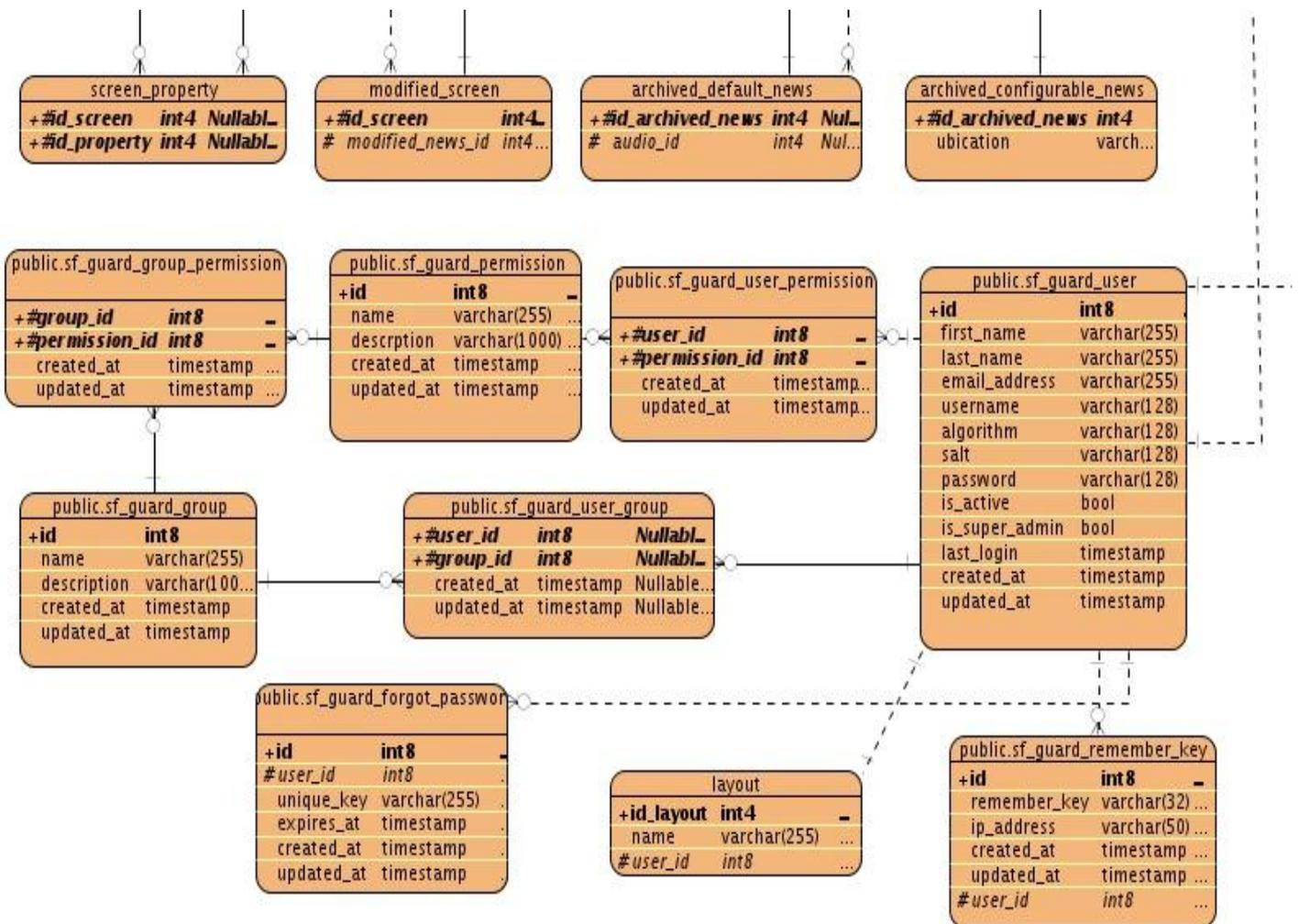


Figura 4: Diagrama Entidad- Relación.

2.7.1 Descripción de las tablas del Modelo Entidad- Relación.

A continuación se muestra la descripción de las tablas que conforman el diseño de la base de datos.

Para ver la descripción detallada con los campos de cada tabla: [Ver Anexo 2](#).

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 2: Descripción de la tabla signal.

| |
|---|
| Nombre: signal |
| Descripción: contiene los datos de las señales que se pueden transmitir. |

Tabla 3: Descripción de la tabla signal_changes.

| |
|---|
| Nombre: signal_changes |
| Descripción: contiene los cambios programados de la señal. |

Tabla 4: Descripción de la tabla section.

| |
|---|
| Nombre: section |
| Descripción: Esta tabla contiene los datos de todas las secciones temáticas del sistema. |

Tabla 5: Descripción de la tabla headband.

| |
|--|
| Nombre: headband |
| Descripción: Esta tabla contiene las infocintas que se mostrarán en cada sección. |

Tabla 6: Descripción de la tabla image.

| |
|--|
| Nombre: image |
| Descripción: Esta tabla contiene todos los datos de las imágenes almacenadas. |

Tabla 7: Descripción de la tabla media.

| |
|--|
| Nombre: media |
| Descripción: Esta tabla contiene la información general de todas las medias(video, image y audio) |

Tabla 8: Descripción de la tabla audio.

| |
|--|
| Nombre: audio |
| Descripción: Esta tabla contiene todos los datos de los audios almacenados. |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 9: Descripción de la tabla music_count.

| |
|--|
| Nombre: music_count |
| Descripción: Esta tabla se utiliza para saber la cantidad de veces que se usó una música determinada. |

Tabla 10: Descripción de la tabla video.

| |
|--|
| Nombre: video |
| Descripción: Esta tabla contiene todos los datos de los videos almacenados. |

Tabla 11: Descripción de la tabla author.

| |
|---|
| Nombre: author |
| Descripción: contiene los autores de cada una de las medias. |

Tabla 12: Descripción de la tabla template.

| |
|---|
| Nombre: template |
| Descripción: contiene las plantillas asociadas a las noticias configurables. |

Tabla 13: Descripción de la tabla news.

| |
|---|
| Nombre: news |
| Descripción: Esta tabla contiene la información general referente a las noticias (default_news y configurable_news). |

Tabla 14: Descripción de la tabla default_news.

| |
|---|
| Nombre: default_news |
| Descripción: contiene los datos de las noticias predeterminadas. |

Tabla 15: Descripción de la tabla configurable_news.

| |
|---|
| Nombre: configurable_news |
| Descripción: contiene los datos de las noticias configurables. |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 16: Descripción de la tabla archived_news.

| |
|---|
| Nombre: archived_news |
| Descripción: Esta tabla contiene todas las noticias que han sido archivadas. |

Tabla 17: Descripción de la tabla modified_news.

| |
|--|
| Nombre: modified_news |
| Descripción: Esta tabla contiene las noticias que han sido modificadas. |

Tabla 18: Descripción de la tabla published_news.

| |
|--|
| Nombre: published_news |
| Descripción: Esta tabla contiene todas las noticias que van a ser publicadas. |

Tabla 19: Descripción de la tabla block.

| |
|---|
| Nombre: block |
| Descripción: Esta tabla contiene la información de los bloques noticiosos. |

Tabla 20: Descripción de la tabla block_published_news.

| |
|--|
| Nombre: block_published_news |
| Descripción: Esta tabla surge a partir de la relación mucho a mucho entre block (bloque) y published_new (noticia). |

Tabla 21: Descripción de la tabla keyword.

| |
|--|
| Nombre: keyword |
| Descripción: Esta tabla contiene las palabras claves del sistema. |

Tabla 22: Descripción de la tabla media_keyword.

| |
|--|
| Nombre: media_keyword |
| Descripción: Esta tabla surge a partir de la relación mucho a mucho entre las tablas media (media) y keyword (palabra clave). |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 23: Descripción de la tabla screen.

| |
|--|
| Nombre: screen |
| Descripción: Esta tabla contiene las pantallas de las noticias predeterminadas. |

Tabla 24: Descripción de la tabla archived_screen.

| |
|---|
| Nombre: archived_screen |
| Descripción: Esta tabla contiene las pantallas de las noticias archivadas. |

Tabla 25: Descripción de la tabla format.

| |
|---|
| Nombre: format |
| Descripción: Esta tabla contiene el formato de fecha y hora del sistema. |

Tabla 26: Descripción de la tabla modified_screen.

| |
|---|
| Nombre: modified_screen |
| Descripción: Esta tabla contiene las pantallas de las noticias que han sido modificadas. |

Tabla 27: Descripción de la tabla genre.

| |
|--|
| Nombre: genre |
| Descripción: Esta tabla contiene los géneros de las medias. |

Tabla 28: Descripción de la tabla layout.

| |
|--|
| Nombre: layout |
| Descripción: Esta tabla contiene los datos de las plantillas. |

Tabla 29: Descripción de la tabla parameter.

| |
|---|
| Nombre: parameter |
| Descripción: En esta tabla se almacenan los parámetros que contiene cada uno de los métodos. |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 30: Descripción de la tabla web_source.

| |
|--|
| Nombre: web_source |
| Descripción: Esta tabla contiene la información referente a las fuentes web que utilice el sistema. |

Tabla 31: Descripción de la tabla transmission.

| |
|---|
| Nombre: transmission |
| Descripción: Esta tabla contiene la información referente a las transmisiones. |

Tabla 32: Descripción de la tabla property.

| |
|---|
| Nombre: property |
| Descripción: Esta tabla contiene las propiedades que puede tener el sistema. |

Tabla 33: Descripción de la tabla effect.

| |
|---|
| Nombre: effect |
| Descripción: Esta tabla contiene los efectos que puede tener el sistema. |

Tabla 34: Descripción de la tabla property_effect.

| |
|---|
| Nombre: property_effect |
| Descripción: Esta tabla surge a partir de la relación mucho a mucho entre las tablas property (propiedad) y effect (efecto). |

Tabla 35: Descripción de la tabla screen_property.

| |
|---|
| Nombre: screen_property |
| Descripción: Esta tabla surge a partir de la relación mucho a mucho entre las tablas property (propiedad) y screen (pantalla). |

Tabla 36: Descripción de la tabla service.

| |
|--|
| Nombre: service |
| Descripción: Esta tabla contiene los servicios que puede utilizar el sistema. |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 37: Descripción de la tabla `element_information`.

| |
|--|
| Nombre: <code>element_information</code> |
| Descripción: Esta tabla contiene los elementos de la información del sistema. |

Tabla 38: Descripción de la tabla `method`.

| |
|--|
| Nombre: <code>method</code> |
| Descripción: Esta tabla contiene los métodos que puede tener los servicios. |

Tabla 39: Descripción de la tabla `archived_default_news`.

| |
|--|
| Nombre: <code>archived_default_news</code> |
| Descripción: Esta tabla contiene archivadas las noticias predeterminadas. |

Tabla 40: Descripción de la tabla `archived_configurable_news`.

| |
|--|
| Nombre: <code>archived_configurable_news</code> |
| Descripción: Esta tabla contiene archivadas las noticias configurables. |

Tabla 41: Descripción de la tabla `published_default_news`.

| |
|---|
| Nombre: <code>published_default_news</code> |
| Descripción: Esta tabla contiene archivadas las noticias predeterminadas a publicar. |

Tabla 42: Descripción de la tabla `published_configurable_news`.

| |
|--|
| Nombre: <code>published_configurable_news</code> |
| Descripción: Esta tabla contiene archivadas las noticias predeterminadas. |

Tabla 43: Descripción de la tabla `public.sf_guard_user`.

| |
|--|
| Nombre: <code>public.sf_guard_user</code> |
| Descripción: Esta tabla contiene los datos de los usuarios. |

Capítulo 2: Descripción y análisis de la solución propuesta.

Tabla 44: Descripción de la tabla public.sf_guard_group.

| |
|--|
| Nombre: public.sf_guard_group |
| Descripción: Esta tabla contiene los datos de los grupos de usuarios. |

Tabla 45: Descripción de la tabla public.sf_guard_permission.

| |
|--|
| Nombre: public.sf_guard_permission. |
| Descripción: Esta tabla contiene los datos de los permisos de los usuarios. |

Tabla 46: Descripción de la tabla public.sf_guard_group_permission.

| |
|---|
| Nombre: public.sf_guard_group_permission. |
| Descripción: Esta tabla surge de la relación de muchos a muchos entre las tablas public.sf_guard_group y public.sf_guard_permission. |

Tabla 47: Descripción de la tabla public.sf_guard_user_permission.

| |
|--|
| Nombre: public.sf_guard_user_permission. |
| Descripción: Esta tabla surge de la relación de muchos a muchos entre las tablas public.sf_guard_user y public.sf_guard_permission. |

Tabla 48: Descripción de la tabla public.sf_guard_forgot_password.

| |
|---|
| Nombre: public.sf_guard_forgot_password. |
| Descripción: Esta tabla contiene el registro de contraseñas olvidadas de cada usuario. |

Tabla 49: Descripción de la tabla public.sf_guard_remember_key

| |
|--|
| Nombre: public.sf_guard_remember_key. |
| Descripción: Esta tabla contiene el registro de contraseñas recordadas de cada usuario. |

Tabla 50: Descripción de la tabla public.sf_guard_user_group

| |
|---|
| Nombre: public.sf_guard_user_group. |
| Descripción: Esta tabla surge de la relación de muchos a muchos entre las tablas public.sf_guard_user y public.sf_guard_group. |

Capítulo 2: Descripción y análisis de la solución propuesta.

2.8 Optimización de consultas.

La optimización de consultas es un proceso que pretende mejorar los tiempos de respuesta en un SGBD relacional. Se debe tener en cuenta que los lenguajes de consulta relacionales son no procedimentales, en los cuales el usuario solo indica cual es el resultado que desea obtener y no el camino de acceso en la base de datos para llegar a dicho resultado. En bases de datos relacionales el lenguaje de consultas SQL es el más utilizado por los desarrolladores para obtener información desde la base de datos. La complejidad que pueden alcanzar algunas consultas puede ser tal, que el diseño de la misma puede tomar un tiempo considerable, obteniendo no siempre una respuesta óptima. (Domínguez, 2004)

Por tales motivos se debe tener en cuenta a la hora de diseñar la base de datos el ajuste máximo del tamaño de los campos, con el objetivo de evitar ocupar mayor espacio, así como agilizar las consultas relacionadas con los mismos. También el trabajo con índices de las llaves primarias y foráneas debe valorarse, además de atributos mediante los cuales se realizarán búsquedas o campos por los cuales se va a ordenar el resultado obtenido. Los índices son usados para encontrar rápidamente los registros que tengan un determinado valor en alguna de sus columnas pero tienen la desventaja que al aumentar el tamaño del campo se hace más engorroso realizar operaciones como inserción, actualización y modificación. Cuando las tablas involucradas tienen muchos registros las consultas que tienen JOIN incluidos se vuelven más lentas y la respuesta de ejecución tiende a demorar bastante. Por tales razones se debe tomar la decisión de aumentar o no el tamaño de las tablas con el uso de índices produciendo una mayor velocidad de respuesta en la obtención de tuplas, a costa de un mayor tiempo de ejecución en consultas sobre las tablas de inserción y modificación.

2.9 Propuesta de replicación de la base de datos.

Streaming Replication (SR) es una nueva funcionalidad que permite transferir asincrónicamente registros WAL sobre la marcha (record-based log shipping) entre un servidor maestro y uno/varios esclavos. Una replicación basada en la transferencia de registros WAL significa que se replicarán absolutamente todas las bases de datos y cambios que se realicen en el servidor maestro. Si lo que se necesita es replicar solamente algunas de las bases de datos existentes en el maestro o algunas tablas, se tendrá que usar otro tipo de replicación.

SR se configura mediante los parámetros *primary_conninfo* en el fichero *recovery.conf* y *max_wal_senders*, *wal_sender_delay* y *wal_keep_segments* en *postgresql.conf*.

Capítulo 2: Descripción y análisis de la solución propuesta.

Nota: En el archivo `postgresql.conf` del servidor maestro se configuran los siguientes parámetros:

`wal_level = hot_standby`, `archive_mode = on`, `archive_command = 'exit 0'`, `max_wal_senders = 10`,
`wal_keep_senders = 5000`, `hot_standby = on`

En la práctica un proceso denominado *receptor WAL* (*WAL receiver*) en el servidor esclavo, se conecta mediante una conexión TCP/IP al servidor maestro. En el servidor maestro existe otro proceso denominado *remitente WAL* (*WAL sender*) que es el encargado de mandar los registros WAL sobre la marcha al servidor esclavo. A continuación se muestra un gráfico sobre el funcionamiento de SR.

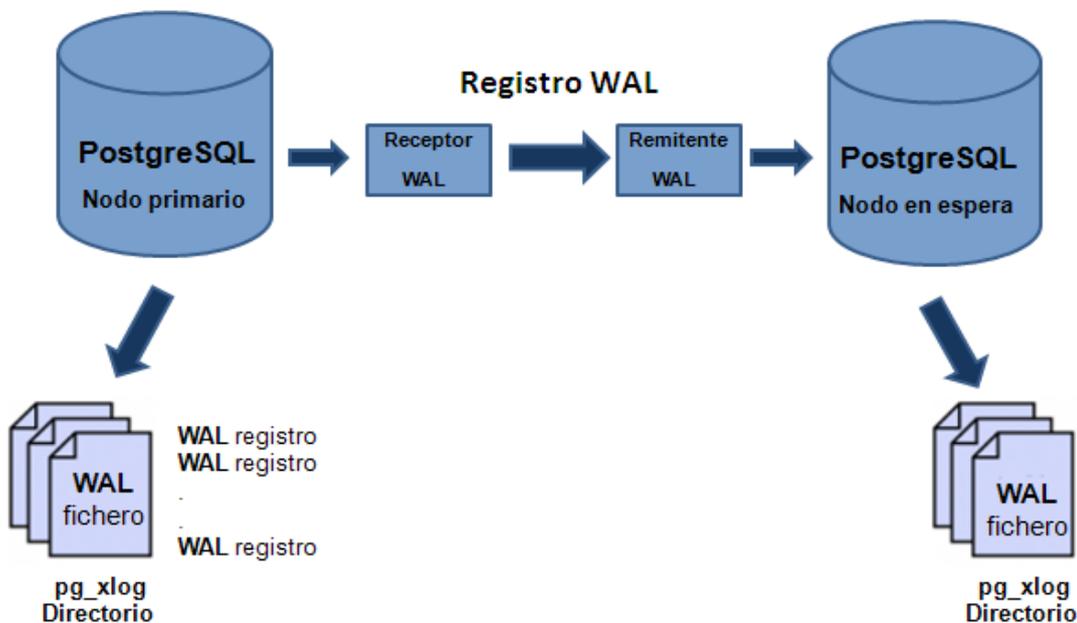


Figura 5: Funcionamiento del Streaming Replication.

Por su parte, Hot Standby (HS) permite acceder en modo de solo-lectura a todos los datos disponibles en el servidor esclavo a donde se está replicando la base de datos. HS se configura mediante los parámetros `hot_standby` y `max_standby_delay` en `postgresql.conf`. El siguiente gráfico explica como HS funciona usando SR y transferencia de ficheros WAL.

Capítulo 2: Descripción y análisis de la solución propuesta.

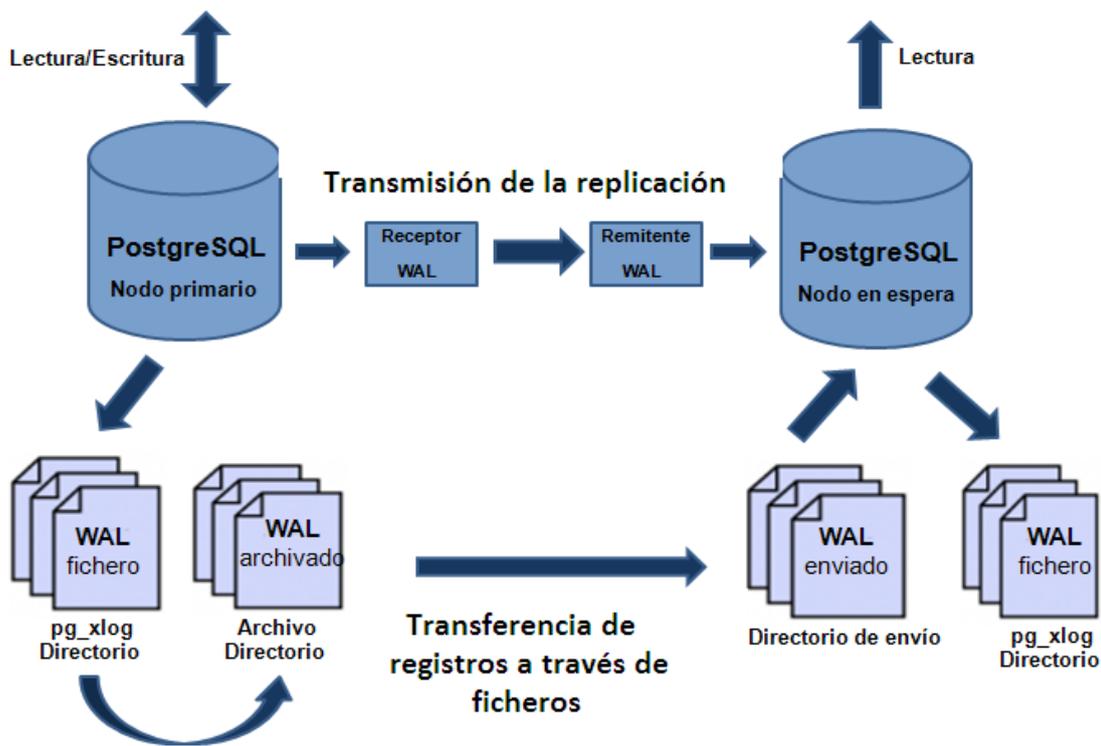


Figura 6: Funcionamiento del Hot Standby.

2.10 Conclusiones parciales.

En este capítulo primeramente se hizo un estudio detallado de los nuevos requisitos funcionales descritos por el analista del proyecto PRIMICIA con el objetivo de analizar el impacto que estos iban a tener en la base de datos que se modelaría. Este proceso permitió que se obtuviera el Modelo Entidad-Relación o Modelo físico de la BD, el cual cuenta con un total de 50 tablas satisfaciendo así la necesidad planteada por dichos requisitos funcionales. Otro de los resultados que se obtuvo en el desarrollo de este capítulo fue la implementación de la replicación la cual ha permitido que la base de datos de la nueva versión de la plataforma PRIMICIA haya quedado configurada para poder realizar todo el proceso de réplicas, garantizando así una alta disponibilidad de la base de datos.

Capítulo 3: Validación del diseño realizado.

3.1 Introducción

En este capítulo se realiza la validación teórica y funcional de la base de datos diseñada. Primeramente se analizan varios aspectos tales como: la integridad, la normalización, el análisis de la redundancia de la información y la seguridad de la base de datos. También se realizan pruebas de volumen y carga a la misma para comprobar su funcionamiento.

3.2 Validación teórica del diseño de la base de datos.

Todo el diseño de la base de datos constituye un proceso primordial, pero luego de este, se deben garantizar aspectos como la consistencia, integridad y seguridad de los datos, como detalles importante en el proceso de desarrollo de una base de datos. Este último representa entre los aspectos mencionados el más importante debido a que garantiza el control de acceso a los datos por las personas autorizadas y además que estos datos no se dañen debido a acciones que no hayan sido previamente controladas.

3.2.1 Integridad

El concepto de integridad de la base de datos hace referencia a la validez, corrección, consistencia, completitud y coherencia de los datos almacenados en la BD. Su principal función se basa en proteger la base de datos contra operaciones que introduzcan inconsistencia en los datos. Esta integridad se expresa normalmente a través de restricciones o reglas, las cuales la base de datos no puede quebrantar. El sistema gestor de base de datos es el encargado de mantener dichas reglas.

Cuando se realizan acciones tales como: insertar datos incorrectos o no válidos, modificar la información existente tomando un valor equivocado o eliminando datos que produzcan la violación de alguna restricción en la base de datos, se está ante una situación en la que la integridad de los datos se encuentra afectada.

La integridad de los datos se identifica en varias categorías tales como:

Integridad de entidad

Las restricciones de entidades aseguran la integridad de las entidades que son modeladas por el sistema. Un ejemplo de este tipo de restricción es el de las llaves primarias para cada tabla. En el caso de la base de datos para PRIMICIA versión 2.0 cada entidad tiene definida la llave primaria, que no puede ser nula ni repetida. Definir correctamente el identificador principal para cada registro, es un

Capítulo 3: Validación del diseño realizado

proceso importante, debido a que esto representa la principal herramienta que utiliza el servidor de base de datos para seleccionar cualquier información que se necesite. (PostgreSQL, 2011)

Integridad de dominio

La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Algunas tareas que aseguran la integridad de dominio son: definir los tipos de datos correctamente para cada campo y 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, etc.

Una definición adecuada de las restricciones de los dominios no solo permite verificar los valores introducidos en la base de datos, sino también examinar las consultas para asegurarse de que tengan sentido las comparaciones que hagan.

Para lograr la integridad de dominio se hace uso de los tipos de datos predefinidos por el gestor de BD utilizado, específicamente integer, varchar, date, time, timestamp y boolean. Además se revisan los valores que no pueden clasificarse de nulos haciendo uso de la restricción NOT NULL. (PostgreSQL, 2011)

Integridad referencial

La integridad referencial es un concepto de base de datos que garantiza que las relaciones entre las tablas siguen siendo coherentes. Cuando una tabla tiene una clave externa a otra tabla, el concepto de integridad referencial establece que no se puede agregar un registro a la tabla que contiene la clave externa a menos que exista un registro correspondiente en la tabla vinculada. También incluye las técnicas conocidas como actualización en cascada y eliminar en cascada, que aseguran que los cambios realizados a la tabla vinculada se reflejan en la tabla principal. (Chapple, 2007)

Para mantener dicha integridad en la base de datos se utilizan: 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 referenciada, así como permitir realizar acciones en caso de actualización o eliminación de los valores a los que se hacen referencia, para esto, se define que al realizar la actualización o eliminación de datos, las mismas se realizarían en cascada.

3.2.2 Normalización de la base de datos.

Capítulo 3: Validación del diseño realizado

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Para realizar la normalización se fue comprobando que cada relación o tabla cumpliera con varias reglas basadas en las dependencias funcionales y las claves primarias. Cada paso a ejecutar responde a una forma normal.

3.2.2.1 Primera Forma Normal (1FN).

La Primera Forma Normal (1FN) constituye el primer paso en el proceso de normalización de la base de datos y representa el más importante de todos. Durante este proceso de normalización, se puede comprobar que satisface la (1FN), si se demuestra que no hay presencia de atributos compuestos o multivaluados, o la combinación de estos. Cada uno de los atributos de la relación debe tener un valor atómico. Además cada una de las tablas deben contener una clave primaria y esta no contiene atributos nulos. (Vicuña, 2009)

La base de datos para la plataforma PRIMICIA cumple con la 1FN, ya que los datos de las relaciones involucradas son atómicos, es decir, no son ni multivaluados, ni compuestos y cada una de las tablas obtenidas cuenta con una clave primaria.

3.2.2.2 Segunda Forma Normal (2FN).

Una relación está en Segunda Forma Normal si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal, es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal). (Vicuña, 2009)

Para eliminar las dependencias parciales de las claves primarias, se eliminan los atributos que son funcionalmente dependientes de cada relación y se colocan en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen). La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos.

Después de haber alcanzado el nivel de la 2FN, se controla la mayoría de los problemas de lógica en la base de datos para la plataforma PRIMICIA.

3.2.2.3 Tercera Forma Normal (3FN).

Capítulo 3: Validación del diseño realizado

La Tercera Forma Normal (3FN) plantea que la relación debe estar en 2FN 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.

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. (Vicuña, 2009)

Para lograr esto en la base de datos diseñada, se eliminan los atributos de cada relación que dependían transitivamente y se ponen en nuevas relaciones con una copia del atributo o los atributos no clave de los que dependen.

Después de realizar todo este proceso, la BD de la plataforma de televisión informativa PRIMICIA, ha quedado normalizada hasta la 3FN, garantizándose de este modo que no existan datos repetidos en la BD diseñada.

3.2.3 Análisis de la redundancia de la información.

La redundancia de información es un término que se utiliza para hacer referencia a la presencia de datos repetidos en la base de datos. Esto debe evitarse, ya que la redundancia dificulta la tarea de modificación de datos, y es el motivo más frecuente de inconsistencia de datos. 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 cien por ciento, lo que a veces es deseable por cuestiones de rendimiento.

Con un buen diseño de 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. En el caso de la base de datos de PRIMICIA, al normalizarla, se eliminó completamente la presencia de datos repetidos innecesariamente.

3.2.4 Análisis de la seguridad de la base de datos.

La seguridad de los datos es un tema sumamente importante a tratar en este capítulo. Se basa en garantizar la coherencia de los datos, comprobando que sólo los usuarios autorizados puedan efectuar las operaciones correctas sobre la base de datos. Esto se consigue mediante un control sobre los

Capítulo 3: Validación del diseño realizado

usuarios que acceden a la base de datos y los tipos de operaciones que están autorizados a realizar. (Ribagorda, 2005)

En PRIMICIA, 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 la transmisión constante de las noticias, funcionalidad principal de la plataforma.

En la actualidad los SGBD existentes garantizan en cierta medida el control de acceso por parte de los usuarios a la BD de acuerdo a los privilegios que tengan asignados y los lugares donde se encuentren ubicados garantizando así su seguridad. El SGBD utilizado para diseñar las bases de datos de la plataforma PRIMICIA –PostgreSQL– incluye formas de restringir el acceso al sistema. Esta función se denomina control de acceso y se pone en práctica creando cuentas de usuarios y contraseñas para que el SGBD controle el proceso de entrada al sistema.

Primeramente se configuran los permisos de conexión para los host y los usuarios a las BDs, que se recogen en el archivo `pg_hba.conf`. En este mismo fichero 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. Después el gestor de base de datos PostgreSQL permite configurar en el archivo `postgresql.conf` las direcciones TCP / IP a través de las cuales el servidor puede escuchar desde aplicaciones cliente, el máximo número de conexiones simultáneas a la BD y el puerto por el cual se van a conectar. (Drake, 2008)

Como seguridad definida por el sistema, se define dar control total de todas las tablas de la base de datos solamente a los usuarios que respondieran al rol de Jefe del Canal, y este usuario será el único que podrá insertar, modificar o eliminar los nuevos usuarios.

Los roles que se definen para cada usuario, protegen la información almacenada, ya que los usuarios registrados tienen los permisos restringidos, es decir solo podrán acceder a las tablas de la base de datos que respondan al rol que desempeñan.

3.3 Validación funcional del diseño de la base de datos.

Después de haber tratado acerca de la seguridad, redundancia e integridad de la base de datos se procede a la validación funcional de la misma. Este es un punto muy importante, pues se comprueba que la base de datos propuesta cumple con los requisitos funcionales que fueron definidos por el analista del proyecto. Para ello se realizan un conjunto de pruebas con el objetivo de simular una

carga de producción real y posteriormente observar el comportamiento que tiene la base de datos bajo dicha carga.

3.3.1 Pruebas de rendimiento.

La validación funcional de la base de datos se basa en las pruebas de rendimiento realizadas donde se determina la respuesta de un sistema ante una cierta carga, así como validar otros atributos relacionados con la calidad, como pueden ser la escalabilidad o el uso de recursos, entre otros.

Ahora bien, existen distintos tipos de pruebas de rendimiento que ayudan a mejorar las capacidades de la aplicación que se esté desarrollando, maximizándose así el beneficio del negocio.

3.3.1.1 Prueba de Volumen.

Las pruebas de volumen se realizan con el objetivo de analizar el comportamiento de la base de datos, con volúmenes de datos almacenados lo más similar posible a los esperados en la explotación real del sistema. Estas pruebas se realizan con la herramienta EMS Data Generator 2011 para PostgreSQL, la cual permite la generación de datos para una o varias tablas simultáneamente, 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.

Con el objetivo de definir la cantidad de tuplas a generar en cada tabla para realizar esta prueba se analizaron, con el Jefe del proyecto PRIMICIA, las estadísticas de todo el proceso de redacción de la noticia en 1 año aproximadamente, ya que este proceso es el que más cúmulo de datos guarda en la BD. A partir de esto se decide generar 32 100 tuplas a la mayoría de las tablas, poblándose en total de 1 222 200 tuplas en la base de datos, a razón de más de 600 por segundos.

Al introducir los datos no se presentaron problemas de volumen de datos, ni de límite de capacidad. Tampoco se detectaron desbordamientos de columnas, tipos de datos, atributos, ni peticiones excesivas de memoria. Esto garantiza que el diseño de la estructura de datos soporta correctamente el almacenamiento de los niveles de información requeridos para la puesta en marcha de la BD.

3.3.1.2 Prueba de carga intensiva

Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de consultas durante el

Capítulo 3: Validación del diseño realizado

tiempo que dura la carga. Esta prueba muestra los tiempos de respuesta de todas las consultas importantes de la aplicación.

Para la realización de este tipo de pruebas a la BD PRIMICIA se emplea 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 seleccionan de acuerdo a las operaciones que se esperan que sean las más utilizadas una vez implantada la BD.

Consulta 1: Listar todos los datos de las noticias predeterminadas que se estén publicando en un rango de tiempo definido.

SELECT

```
section.name,  
section.logo,  
news.title,  
news.id_news,  
published_news.immediate,  
published_news.cyclic,  
block.name,  
block.logo,  
block_published_news.order
```

FROM

```
public.section, public.news, public.block, public.block_published_news, public.published_news
```

WHERE

```
section.id_section=news.section_id AND  
news.id_news=published_news.news_id AND  
block.id_block=block_published_news.block_id AND  
block_published_news.published_news_id=published_news.id_published_news AND  
'2011-04-11' BETWEEN published_news.start_date AND published_news.end_date AND
```

Capítulo 3: Validación del diseño realizado

```
'2:45:05' BETWEEN published_news.start_time AND published_news.end_time AND '05:45:05'  
BETWEEN block.start_time AND block.end_time AND '22:46:05' BETWEEN section.start_time AND  
section.end_time AND section.activated='TRUE'
```

ORDER BY

```
block_published_news.order ASC;
```

Para un volumen de 96 500 tuplas se generaron 28 580 resultados en un tiempo de 1010 ms.

Consulta 2: Listar datos de las noticias predeterminadas cuyo audio tenga un tamaño de archivo superior a 100.00 Kb.

SELECT

```
public.audio.title,  
public.default_news.id_news,  
public.news.title,  
public.news."created_at",  
public.news.section_id,  
public.news.user_id,  
public.news.corrected,  
public.news.validated_at,  
public.news.is_validate
```

FROM

```
public.audio INNER JOIN public.media ON (public.audio.id_media = public.media.id_media)  
INNER JOIN public.default_news ON (public.audio.id_media = public.default_news.audio_id)  
INNER JOIN public.news ON (public.default_news.id_news = public.news.id_news)
```

WHERE

```
media.file_size > 100.00
```

Para un volumen de 64 300 tuplas se generan 100 resultados en un tiempo de 341 ms.

Consulta 3: Listar datos relacionados con las noticias publicadas que su fecha de inicio esté entre el 10 de abril del 2011 y el 10 de junio del 2011.

SELECT

```
public.published_news.immediate,  
public.published_news.cyclic,  
public.section.name,  
public.section.logo,  
public.news.title,  
public.news.id_news
```

FROM

```
public.section INNER JOIN public.news ON (public.section.id_section = public.news.section_id)  
INNER JOIN public.published_news ON (public.news.id_news = public.published_news.news_id)
```

WHERE

```
published_news.start_date BETWEEN '2011-04-10' AND '2011-06-10'
```

Para un volumen de 96 300 tuplas se generan 14 340 resultados en un tiempo de 958 ms.

Luego de haber comprobado el comportamiento de la base de datos frente a volúmenes elevados de datos se puede decir que la misma los procesa correctamente en un intervalo de tiempo proporcional. Los resultados son alentadores pero no se debe confiar debido a que la mejor prueba que se le puede realizar a un producto informático es la interacción diaria con el usuario.

3.3.2 Procedimientos almacenados.

Un procedimiento almacenado es un programa, procedimiento o función, que está almacenado en la base de datos y listo para ser usado. La ejecución del procedimiento ocurre en el propio servidor de bases de datos. Esto trae consigo que aumente el rendimiento de la aplicación al no tenerse que mandar datos entre el cliente y el servidor, y además no tener que procesar resultados intermedios en el cliente para obtener el resultado final. En PostgreSQL un procedimiento almacenado se puede escribir en múltiples lenguajes de programación, en este caso, se ha usado el PL/pgSQL el cual está disponible automáticamente, además de ser fácil de aprender y potente.

A continuación se muestra un ejemplo de un procedimiento realizado:

Capítulo 3: Validación del diseño realizado

```
CREATE TYPE tipo_screen AS (id_screen int, "type" int, "order" int, "text" varchar(255), video_id int, image_id int, default_news_id int);
```

```
CREATE FUNCTION proc1 (integer) RETURNS tipo_screen AS
```

```
$BODY$
```

```
DECLARE
```

```
id_noticia ALIAS FOR $1;
```

```
SELECT
```

```
screen.id_screen, screen."type", screen."order", screen."text", screen.video_id, screen.image_id,  
screen.default_news_id
```

```
FROM
```

```
public.screen, public.default_news
```

```
WHERE
```

```
public.screen.default_news_id = public.default_news.id_news AND public.default_news.id_news = $1
```

```
IF
```

```
NOT FOUND THEN RAISE EXCEPTION "No se encuentra ese identificador";
```

```
END IF;
```

```
$BODY$
```

```
LANGUAGE 'plpgsql' VOLATILE;
```

```
ALTER FUNCTION proc1 (integer) OWNER TO postgres;
```

3.4 Comparación con la versión 1.0 de la BD de PRIMICIA.

La versión 1.0 de la base de datos de la plataforma de televisión informativa PRIMICIA fue creada hace aproximadamente 2 años. Desde la puesta en marcha de la misma se han ido implementando, por parte de los desarrolladores, nuevas funcionalidades las cuales no son garantizadas con ese diseño propuesto. A partir de ahí y de la necesidad de migrar hacia una segunda versión se ha decidido crear una nueva estructura de almacenamiento, la cual tuviera en cuenta el diseño ya realizado. Con el desarrollo de la versión 2.0 de la base de datos se han ido incrementando nuevas tablas, nuevos atributos y modificaciones en las tablas ya existentes, la incorporación de un plugin que

Capítulo 3: Validación del diseño realizado

garantice la seguridad del sistema, la implementación de la replicación, entre otras. A continuación se realiza una comparación entre ambas versiones.

Tabla 51: Comparación entre versiones de base de datos de PRIMICIA.

| | Versión 1.0 | Versión 2.0 |
|-----------------------------|--|---|
| Gestor de BD | PostgreSQL 8.2 | PostgreSQL 9.0 |
| Cantidad de tablas | 20 | 50 |
| Idioma de la BD | Español | Inglés |
| Replicación | No | Si |
| Garantizar seguridad | Mediante 3 tablas en la BD (usuario, rol, usuario_rol) | Incorporación de un Plugin con 9 tablas en la BD. |
| Cambios en el diseño | Tabla noticia_corregida. | Se eliminó esta tabla y se agregaron los campos <i>is_corrected</i> y <i>validated_at</i> en la tabla <i>news</i> . |
| | Tabla tipo_pantalla. | Se eliminó esta tabla y se agregó el campo <i>type</i> en la tabla <i>screen</i> (pantalla). |
| | Tablas imagen, video y musica. | Se creó una tabla padre <i>media</i> ya que habían datos repetidos en ambas tablas y se le agregaron nuevos campos a cada una de las tablas ya existentes. |
| | Tabla noticia | Se crearon 2 nuevas tablas que heredan de <i>news</i> (<i>default_news</i> y <i>configurable_news</i>), debido a la existencia de 2 tipos de noticias diferentes. |
| Pruebas de volumen | Se generaron 5000 tuplas en cada tabla. | Fueron generadas 32 000 tuplas en cada tabla de la BD. |
| Consulta de prueba | Listar los datos de las noticias redactadas del usuario que tiene identificador 17. | |
| | SELECT noticia.id_noticia, noticia.titulo, noticia.fecha_confeccion, noticia.usuario_id, noticia.musica_id, | SELECT news.id_news, news.title, news.updated_at, news.section_id, news.user_id, news_corrected, |

Capítulo 3: Validación del diseño realizado

| | | |
|------------|---|---|
| | <code>noticia.seccion_id</code> | <code>news.validated_at, news.is_validated</code> |
| | FROM noticia | FROM news |
| | WHERE noticia.usuario_id=17 | WHERE news.user_id = 17 |
| | ORDER BY noticia.id_noticia ASC | ORDER BY news.id_news ASC |
| Resultados | Para un volumen de 20 206 se arrojaron 120 resultados en un tiempo de 407 ms. | Para un volumen de 32 100 se arrojaron 321 resultados en un tiempo de 109 ms. |

En la tabla solo se muestran algunos cambios, pero de esta forma se demuestra que el diseño de la versión 2.0 de la base de datos para PRIMICIA ha sido más profundo y radical que el anterior.

3.5 Conclusiones parciales.

Este capítulo se ha centrado en documentar todo el proceso de validación funcional y teórica de la base de datos. En este se realizaron varias pruebas a la BD lo cual permitió comprobar que la estructura diseñada respondía correctamente ante grandes volúmenes de datos insertados previamente. Por lo tanto, después de haber realizado todas estas pruebas se puede decir que la base de datos para la plataforma de televisión informativa PRIMICIA está lista para su funcionamiento.

Conclusiones

Hoy en día las bases de datos constituyen una de las estructuras más eficientes para guardar información, logrando así que grandes cantidades de esta sean almacenadas sin redundancia, centralizada y bien estructurada. A su vez, los sistemas gestores de bases de datos también han ido marcando grandes pautas ya que han aumentado su potencialidad con el transcurso del tiempo, siendo hoy en día los SGBD relacionales los más utilizados a nivel mundial y nacional.

Con el desarrollo de este trabajo se ha dotado a la plataforma de televisión informativa PRIMICIA de una nueva estructura para almacenar la información correspondiente a toda la gestión del canal, donde se garantiza el funcionamiento adecuado de la misma, dando respuesta así al problema que ha dado lugar a este trabajo de diploma. Esta BD cumple con los requerimientos funcionales que demanda la plataforma. Para todo su desarrollo se han utilizado solamente herramientas de código abierto, evitando así la existencia de restricciones por concepto de licencias de software propietario.

Recomendaciones

Después de haber realizado el diseño de la base de datos de la plataforma de televisión informativa PRIMICIA versión 2.0 se recomienda:

- ✚ El refinamiento del modelo lógico propuesto en próximas iteraciones del desarrollo del software.
- ✚ Actualización periódica de la base de datos, logrando que se mantenga la integridad y consistencia de los datos.

Referencias Bibliográficas

Alburquerque, Amado. 2007. *Sistema Integrado de Gestión Estadística (SIGE). Rol Diseñador de la Base de Datos.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Álvarez, Sara. 2007. Arquitectura de las bases de datos. [En línea] 2007. [Citado el: 20 de diciembre de 2010.] <http://www.desarrolloweb.com/articulos/arquitectura-base-de-datos.html>.

Anzaldo, Juan. 2005. Breve Historia de las Bases de datos. [En línea] 2005. [Citado el: 10 de diciembre de 2010.] <http://janzaldo.wordpress.com/2005/12/06/breve-historia-de-las-bases-de-datos/>.

Ballester, Lidier González. 2009. *Diseño de una base de datos para el control de los RRHH en los polos productivos de la facultad 9.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Bartle, Phil. 2009. INFORMACIÓN PARA LA GESTIÓN y gestión de la información. [En línea] 2009. [Citado el: 10 de diciembre de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

Benítez, Frank. 2009. *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Cardero, Licet. 2007. *Diseño e implementación de la Base de Datos del Sistema Informático de los Tribunales Militares de Región.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Casasola, Oscar. Introducción al UML. [En línea] [Citado el: 12 de diciembre de 2010.] [http://www.programacion.com/articulo/introduccion a uml 181](http://www.programacion.com/articulo/introduccion%20a%20uml%20181).

—. 2008. Introducción al UML. [En línea] 2008. [Citado el: 12 de diciembre de 2010.] [http://www.programacion.com/articulo/introduccion a uml 181](http://www.programacion.com/articulo/introduccion%20a%20uml%20181).

Castro, Daniel. 2008. *Replicación de Datos.* Universidad Central "Marta Abreu" de las Villas. Santa Clara : s.n., 2008.

Chapple, Mike. 2007. Referential Integrity. [En línea] 2007. [Citado el: 8 de febrero de 2011.] <http://databases.about.com/cs/administration/g/refintegrity.htm>.

Dante, Gloria. 2010. Gestión de la Información. [En línea] 2010. [Citado el: 20 de noviembre de 2010.] <http://www.fahce.unlp.edu.ar/academica/Areas/bibliotecologia/Catedras/gestindeinformacin/>.

Date, Cristopher. 2003. *Introducción a los Sistemas de Bases de Datos*. Ciudad de la Habana : Félix Varela, 2003.

Domínguez, Arnoldo Leyva. 2004. *Modelo Lógico y Físico de la Base de Datos correspondiente a los Módulos de Investigación Criminalística del proyecto Cicpc*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2004.

Drake, Jhon. 2008. *Practical PostgreSQL*. California : Editorial O'Reilly Media, 2008.

Española, Real Academia. 2009. Real Academia Española. [En línea] Real Academia Española, 2009. [Citado el: 1 de 12 de 2010.] <http://buscon.rae.es/drae/>.

Fajardo, Norge. 2007. *Sistema de réplica para bases de datos distribuidas en PostgreSQL*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

García, Yanly. 2010. *Diseño de la base de datos para el grupo de calidad de la facultad 9*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2010.

González, Carlos D. 2010. Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] 2010. [Citado el: 1 de diciembre de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>.

—. Curso PostgreSQL, SQL avanzado y PHP. [En línea] [Citado el: 1 de diciembre de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>.

González, Carlos de Jesús Andrés. 2009. *Diseño del subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Gutiérrez, Duniel Miranda. 2007. *Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Hansen, Gary W. 2005. Diseño y Administración de Bases de Datos. [En línea] 2005. [Citado el: 15 de 12 de 2010.] <http://bibliodoc.uci.cu/pdf/reg00071.pdf>.

Heber, Romero. 2009. *Gestión del conocimiento y la información en el Polo de Gestión Universitaria*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

- Hernández, Anaisa. 2004.** Un método para el diseño de la base de datos a partir del modelo orientado a objetos. [En línea] 2004. [Citado el: 24 de febrero de 2011.] <http://redalyc.uaemex.mx/pdf/615/61570402.pdf>.
- Hernandis, José A. 2009.** Visual Paradigm for UML. [En línea] 2009. [Citado el: 11 de diciembre de 2010.] <http://www.versionzero.com/noticia/210/visual-paradigm-for-uml>.
- Letelier, Paulo. 2008.** *Rational Unified Process*. Universidad Politécnica de Valencia. España. 2008.
- Mato, Rosa M. 2005.** *Sistemas de Base de Datos*. Ciudad de la Habana : Pueblo y Educación, 2005.
- Murray, Pablo. 2003.** *GESTIÓN – INFORMACIÓN - CONOCIMIENTO*. Escuela de Bibliotecarios de la Biblioteca Nacional de Buenos Aires. Buenos Aires, Argentina : s.n., 2003.
- Mustain, Elein. 2004.** Introducing Slony. [En línea] 2004. [Citado el: 19 de noviembre de 2010.] <http://onlamp.com/pub/a/onlamp/2004/11/18/slony.html>.
- Pérez, Olga. 2007.** *Recordando la Historia: El concepto de "televisión" en sus orígenes*. 2007. pág. 160.
- PostgreSQL, Sitio oficial de. 2011.** Sitio oficial de PostgreSQL. [En línea] 2011. [Citado el: 20 de febrero de 2011.] <http://www.postgresql-es.org>.
- Quiñones, Ernesto. 2009.** Introducción a PostgreSQL. [En línea] 2009. [Citado el: 27 de enero de 2011.] [http://www.postgresql.or.pe/articles/introduccion a postgresql.pdf](http://www.postgresql.or.pe/articles/introduccion%20a%20postgresql.pdf).
- Ribagorda, Arturo. 2005.** *Seguridad en bases de datos*. Madrid : Fundación Dintel, 2005.
- Ruber Hernández, Yunior Montaner. 2008.** *PRIMICIA, Plataforma de Televisión Informativa*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008.
- Ruiz, Dr. Francisco. 2003.** *El modelo de datos jerárquico*. Universidad de Castilla La Mancha. 2003.
- Sampieri, Roberto Hernández. 2004.** *Metodología de la Investigación*. Ciudad de la Habana, Cuba : s.n., 2004.
- Sánchez, Yaniel Álvarez. 2007.** *Base de Datos para la Residencia de la UCI*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Santos, Juan C. 2009. Bases de datos y sistemas de gestión de bases relacionadas y procedimiento software. [En línea] 2009. [Citado el: 16 de diciembre de 2010.] <http://www.slideshare.net/luisitops/bases-de-datos-1497047>.

Singha, Rajib. 2007. What is Rational Unified Process (RUP). [En línea] 2007. [Citado el: 8 de diciembre de 2010.] <http://www.buzzle.com/articles/what-is-rational-unified-process-rup.html>.

Sommerville, Ian. 2005. *Ingeniería del Software*. Madrid, España : Editorial PEARSON EDUCACIÓN, 2005.

Vicuña, Patricio. 2009. *Normalización de una base de datos*. Buenos Aires : s.n., 2009.

Zambrano, Raquel. 2008. *Sistemas gestores de Bases de Datos*. Universidad de Córdoba. 2008.

Bibliografía consultada

Albuquerque, Amado. 2007. *Sistema Integrado de Gestión Estadística (SIGE). Rol Diseñador de la Base de Datos.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Álvarez, Sara. 2007. *Arquitectura de las bases de datos.* [En línea] 2007. [Citado el: 20 de diciembre de 2010.] <http://www.desarrolloweb.com/articulos/arquitectura-base-de-datos.html>.

Anzaldo, Juan. 2005. *Breve Historia de las Bases de datos.* [En línea] 2005. [Citado el: 10 de diciembre de 2010.] <http://janzaldo.wordpress.com/2005/12/06/breve-historia-de-las-bases-de-datos/>.

Armengot Iborra, Marcelo J. *Sistema de Base de Datos cliente-servidor con interfaz gráfica multiplataforma: Perl.* Valencia : s.n., 2004.

Ballester, Lidier González. 2009. *Diseño de una base de datos para el control de los RRHH en los polos productivos de la facultad 9.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Bartle, Phil. 2009. *INFORMACIÓN PARA LA GESTIÓN y gestión de la información.* [En línea] 2009. [Citado el: 10 de diciembre de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

Benítez, Frank. 2009. *Procedimiento para el seguimiento y tratamiento de los errores de PRIMICIA.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Burbano Proaño, Diego Javier. 2006. *Análisis comparativo de bases de datos de código abierto.* Quito : s.n., 2006.

Cardero, Licet. 2007. *Diseño e implementación de la Base de Datos del Sistema Informático de los Tribunales Militares de Región.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Casasola, Oscar. *Introducción al UML.* [En línea] [Citado el: 12 de diciembre de 2010.] <http://www.programacion.com/articulo/introduccion a uml 181>.

—. **2008.** *Introducción al UML.* [En línea] 2008. [Citado el: 12 de diciembre de 2010.] <http://www.programacion.com/articulo/introduccion a uml 181>.

Calvo, José Carlos. 2009. *Reutilización, Parametrización y Patrones de Diseño.*

Castro, Daniel. 2008. *Replicación de Datos*. Universidad Central "Marta Abreu" de las Villas. Santa Clara : s.n., 2008.

Chapple, Mike. 2007. Referential Integrity. [En línea] 2007. [Citado el: 8 de febrero de 2011.] <http://databases.about.com/cs/administration/g/refintegrity.htm>.

Dante, Gloria. 2010. Gestión de la Información. [En línea] 2010. [Citado el: 20 de noviembre de 2010.] <http://www.fahce.unlp.edu.ar/academica/Areas/bibliotecologia/Catedras/gestindeinformacin/>.

Date, Cristopher. 2003. *Introducción a los Sistemas de Bases de Datos*. Ciudad de la Habana : Félix Varela, 2003.

Domínguez, Arnoldo Leyva. 2004. *Modelo Lógico y Físico de la Base de Datos correspondiente a los Módulos de Investigación Criminalística del proyecto Cicpc*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2004.

Drake, Jhon. 2008. *Practical PostgreSQL*. California : Editorial O'Reilly Media, 2008.

Española, Real Academia. 2009. Real Academia Española. [En línea] Real Academia Española, 2009. [Citado el: 1 de 12 de 2010.] <http://buscon.rae.es/drael/>.

Fajardo, Norge. 2007. *Sistema de réplica para bases de datos distribuidas en PostgreSQL*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

García, Yanly. 2010. *Diseño de la base de datos para el grupo de calidad de la facultad 9*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2010.

González, Carlos D. 2010. Curso Base de Datos PostgreSQL, SQL avanzado y PHP. [En línea] 2010. [Citado el: 1 de diciembre de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>.

—. Curso PostgreSQL, SQL avanzado y PHP. [En línea] [Citado el: 1 de diciembre de 2010.] <http://www.usabilidadweb.com.ar/postgre.php>.

González, Carlos de Jesús Andrés. 2009. *Diseño del subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.

Gutiérrez, Duniel Miranda. 2007. *Sistema de Manejo de Datos de Ensayos Clínicos: Diseño e implementación de la Base de Datos*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

- Hansen, Gary W. 2005.** Diseño y Administración de Bases de Datos. [En línea] 2005. [Citado el: 15 de 12 de 2010.] <http://bibliodoc.uci.cu/pdf/reg00071.pdf>.
- Heber, Romero. 2009.** *Gestión del conocimiento y la información en el Polo de Gestión Universitaria*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.
- Hernández, Anaisa. 2004.** Un método para el diseño de la base de datos a partir del modelo orientado a objetos. [En línea] 2004. [Citado el: 24 de febrero de 2011.] <http://redalyc.uaemex.mx/pdf/615/61570402.pdf>.
- Hernandis, José A. 2009.** Visual Paradigm for UML. [En línea] 2009. [Citado el: 11 de diciembre de 2010.] <http://www.versionzero.com/noticia/210/visual-paradigm-for-uml>.
- Letelier, Paulo. 2008.** *Rational Unified Process*. Universidad Politécnica de Valencia. España. 2008.
- Mato, Rosa M. 2005.** *Sistemas de Base de Datos*. Ciudad de la Habana : Pueblo y Educación, 2005.
- Moratilla, Antonio. 2008.** *Bases de datos avanzadas. Replicación de bases de datos*.
- Murray, Pablo. 2003.** *GESTIÓN – INFORMACIÓN - CONOCIMIENTO*. Escuela de Bibliotecarios de la Biblioteca Nacional de Buenos Aires. Buenos Aires, Argentina : s.n., 2003.
- Mustain, Elein. 2004.** Introducing Slony. [En línea] 2004. [Citado el: 19 de noviembre de 2010.] <http://onlamp.com/pub/a/onlamp/2004/11/18/slony.html>.
- Pérez, Olga. 2007.** *Recordando la Historia: El concepto de "televisión" en sus orígenes*. 2007. pág. 160.
- PostgreSQL, Sitio oficial de. 2011.** Sitio oficial de PostgreSQL. [En línea] 2011. [Citado el: 20 de febrero de 2011.] <http://www.postgresql-es.org>.
- Pupo, Rosell. 2009.** *Implementación del componente réplica de base de datos para Akademos v2.0*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2009.
- Quiñones, Ernesto. 2009.** Introducción a PostgreSQL. [En línea] 2009. [Citado el: 27 de enero de 2011.] http://www.postgresql.or.pe/articles/introduccion_a_postgresql.pdf.
- Ribagorda, Arturo. 2005.** *Seguridad en bases de datos*. Madrid : Fundación Dintel, 2005.

Ruber Hernández, Yunior Montaner. 2008. *PRIMICIA, Plataforma de Televisión Informativa.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2008.

Ruiz, Dr. Francisco. 2003. *El modelo de datos jerárquico.* Universidad de Castilla La Mancha. 2003.

Sampieri, Roberto Hernández. 2004. *Metodología de la Investigación.* Ciudad de la Habana, Cuba : s.n., 2004.

Sánchez, Yaniel Álvarez. 2007. *Base de Datos para la Residencia de la UCI.* Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2007.

Santos, Juan C. 2009. Bases de datos y sistemas de gestión de bases relacionadas y procedimiento software. [En línea] 2009. [Citado el: 16 de diciembre de 2010.] <http://www.slideshare.net/luisitops/bases-de-datos-1497047>.

Singha, Rajib. 2007. What is Rational Unified Process (RUP). [En línea] 2007. [Citado el: 8 de diciembre de 2010.] <http://www.buzzle.com/articles/what-is-rational-unified-process-rup.html>.

Sommerville, Ian. 2005. *Ingeniería del Software.* Madrid, España : Editorial PEARSON EDUCACIÓN, 2005.

Vicuña, Patricio. 2009. *Normalización de una base de datos.* Buenos Aires : s.n., 2009.

Zambrano, Raquel. 2008. *Sistemas gestores de Bases de Datos.* Universidad de Córdoba. 2008.

Zerquera, Karen Mercedes. 2009. *PRIMICIA, Plataforma de Televisión Informativa. Rol de Diseñador de Base de Datos.* Universidad de las Ciencias Informáticas. Ciudad Habana : s.n., 2009

Glosario de términos

ANSI-SPARC (Comité del Instituto Americano Nacional de Estándar- Normas de Planeación y Requerimientos). Arquitectura propuesta por dicho comité, que representa un estándar de diseño abstracto de un Sistema de Gestión de Base de Datos (SGBD).

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.

Clase: Define los atributos y los métodos de una serie de objetos.

Dependencia funcional: Es una relación entre atributos de una misma relación (tabla). Si X e Y son atributos de la relación R, se dice que Y es funcionalmente dependiente de X (se denota por $X \rightarrow Y$) si cada valor de X tiene asociado un solo valor de Y (X e Y pueden constar de uno o varios atributos). A X se le denomina determinante, ya que X determina el valor de Y. Se dice que el atributo Y es completamente dependiente de X si depende funcionalmente de X y no depende de ningún subconjunto de X.

Entidad: Es cualquier concepto del mundo real con una existencia independiente.

Licencia BSD: La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Licencia GPL: La licencia GPL se aplica al software de la FSF (Free Software Foundation) y el proyecto GNU y otorga al usuario la libertad de compartir el software y realizar cambios en él. Dicho de otra forma, el usuario tiene derecho a usar el programa, modificarlo y distribuir las versiones modificadas pero no tiene permiso de realizar restricciones propias con respecto a la utilización de ese programa modificado.

Modelo entidad-relación: Es una técnica para el modelado de datos utilizando diagramas de entidad-relación.

Normalización: Proceso de reducción sobre una estructura de datos que procura aumentar la integridad, disminuir la redundancia y las dependencias funcionales de esa estructura.

Plataforma: Hardware sobre el cual puede ejecutarse o desarrollarse un software.

PostgreSQL: Es un sistema de gestión de bases de datos objeto-relacionales que tiene alto rendimiento y flexibilidad.

Query: Consiste en una cadena de consultas, normalmente se utilizan para: insertar, actualizar o editar valores de la base de datos.

RUP (Proceso unificado de software). Es una metodología bastante apropiada para proyectos que sean grandes, incluso para pequeños también. Actualmente constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

SGBD (Sistema de 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.

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.

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

Tabla: Se refiere al tipo de modelado de datos, donde se guardan los datos recolectados por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.

Tupla: Se define como una función finita que mapea (asocia unívocamente) los nombres con algunos valores. Una tupla es una secuencia ordenada de objetos.

UCI: Universidad de las Ciencias Informáticas.

UML: (por sus siglas en inglés Unified Modeling Language, Lenguaje Unificado de Modelado). Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

WAL: (Write-Ahead Logging) es un registro de escritura anticipada (WAL) es una familia de técnicas para proporcionar atomicidad y durabilidad (dos de las propiedades ACID) en los sistemas de base de datos.

Windows: Es la familia de sistemas operativos desarrollados y comercializados por Microsoft.