

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Modelo lógico y físico de la base de datos para el Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria

Trabajo de diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Ekaterina Alejovna Ramírez Ramírez

Tutor: Ing. Yuset Martínez Carballo

Mayo 2009

*“Quédate quieta, en silencio, y escucha a tu corazón. Y cuando te hable,
levántate y ve donde él te lleve.”*

Susanna Tamaro

Для моей мамы, которая всегда со мной

Спасибо папа

Спасибо сестре

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ekaterina Alejovna Ramírez Ramírez

Estudiante

Yuset Martínez Carballo

Tutor

Datos del contacto

Tutor: Ing. Yuset Martínez Carballo

- Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2007.
- Profesor del Departamento de Técnicas de Programación, Facultad 9.

E-mail: ycarballo@uci.cu

Resumen

Este trabajo contiene la investigación y desarrollo de una base de datos para el almacenamiento de la información de la aplicación Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria. Incluye un estudio del estado del arte, las herramientas utilizadas, la descripción del modelo implementado y la validación de la base de datos obtenida basada en los criterios de integridad, normalización, seguridad, redundancia de datos, llenado de la base de datos y pruebas de carga. También se proponen mejoras para la evolución del sistema de almacenamiento de información en caso de su posible aplicación en otro ambiente.

Palabras claves: base de datos, almacenar información, televisión, media

Abstract

This work includes research and development data on a database system for the storage of information of Process Management System for the Department of Television University. Includes a study of the state of the art, the tools used, the description of the model implemented and the validation of the database obtained based on the criteria of integrity, normalization, security, data redundancy, database dummy-filling and load tests. Improvements are also proposed for the evolution of the storage system if its possible application in another environment.

Key words: database, store information, television, media

Índice de figuras

Figura 1: Modelo de clases persistentes	27
Figura 2: Modelo entidad-relación de la base de datos	44
Figura 3: Esquema Seguridad.....	45
Figura 4: Esquema Media	48
Figura 5: Esquema Planificación.....	60
Figura 6: Esquema Inventario	64
Figura 7: Modelo entidad-relación sin normalizar	72
Figura 8: Modelo entidad-relación después de la normalización	75
Figura 9: EMS Data Generator 2008 para PostgreSQL. Elección de rangos de datos	78
Figura 10: EMS Data Generator 2008 para PostgreSQL. Generación de datos.	79
Figura 11: Prueba #1. Resultados.....	87
Figura 12: Prueba #1. Gráficos de Promedio y Línea 90%.....	88
Figura 13: Prueba #2. Resultados.....	88
Figura 14: Prueba #2. Gráficos de Promedio y Línea 90%.....	89
Figura 15: Prueba #3: Resultados.....	90
Figura 16: Prueba #3: Gráficos de Promedio y Línea 90%.....	91
Figura 17: Base de datos SGPDTU	100

Índice de tablas

Tabla 1: Perfil tecnológico	22
Tabla 2: Descripción de la clase Equipamiento	29
Tabla 3: Descripción de la clase ControlMovimiento	29
Tabla 4: Descripción de la clase NArea	30
Tabla 5: Descripción de la clase Media.....	30
Tabla 6: Descripción de la clase Documental	31
Tabla 7: Descripción de la clase Teleclase	31
Tabla 8: Descripción de la clase Asignatura	31
Tabla 9: Descripción de la clase AudioVisualMiscelanea	32
Tabla 10: Descripción de la clase VideoMusical	33
Tabla 11: Descripción de la clase Foto	33
Tabla 12: Descripción de la clase Pelicula	34
Tabla 13: Descripción de la clase Serie.....	35
Tabla 14: Descripción de la clase Musica.....	35
Tabla 15: Descripción de la clase NTipoFormatoAudioVisual.....	36
Tabla 16: Descripción de la clase SolicitudMedia	36
Tabla 17: Descripción de la clase PlanProduccion	37
Tabla 18: Descripción de la clase Actividad	37
Tabla 19: Descripción de la clase PlanProgramacion	38
Tabla 20: Descripción de la clase Programacion	38
Tabla 21: Descripción de la clase Usuario	39
Tabla 22: Descripción de la clase Rol	39
Tabla 23: Descripción de la clase AudioVisual	40
Tabla 24: Descripción de la clase Permiso.....	40
Tabla 25: Descripción de la clase AudioVisualUCI	40
Tabla 26: Descripción de la clase AudioVisualExterno	41
Tabla 27: Descripción de la clase NTipoFormatoImagen	41
Tabla 28: Descripción de la clase NEstandarVideo	42
Tabla 29: Descripción de la clase NTipoFormatoMusica	42
Tabla 30: Descripción de la clase NTipoAtributo_Equipamiento	42
Tabla 31: Descripción de la clase NTipoAtributo.....	43
Tabla 32: Descripción de la clase NTipoEquipamiento	43
Tabla 33: Descripción de la entidad Usuario.....	45
Tabla 34: Descripción de la entidad Rol	46
Tabla 35: Descripción de la entidad Permiso.....	46

Tabla 36: Descripción de la entidad Rol_Perminos.....	47
Tabla 37: Descripción de la entidad Serie	49
Tabla 38: Descripción de la entidad Pelicula	49
Tabla 39: Descripción de la entidad VideoMusical.....	50
Tabla 40: Descripción de la entidad Documental	51
Tabla 41: Descripción de la entidad NTipoFormatoAudioVisual	51
Tabla 42: Descripción de la entidad AudioVisualExterno.....	51
Tabla 43: Descripción de la entidad AudioVisual	52
Tabla 44: Descripción de la entidad AudioVisualUCI	52
Tabla 45: Descripción de la entidad Teleclase	53
Tabla 46: Descripción de la entidad Asignatura	54
Tabla 47: Descripción de la entidad AudioVisualMiscelanea	54
Tabla 48: Descripción de la entidad NEstandarVideo	55
Tabla 49: Descripción de la entidad NTipoFormatoImagen.....	56
Tabla 50: Descripción de la entidad Foto	56
Tabla 51: Descripción de la entidad Media.....	57
Tabla 52: Descripción de la entidad Musica.....	58
Tabla 53: Descripción de la entidad SolicitudMedia	59
Tabla 54: Descripción de la entidad NTipoFormatoMusica	59
Tabla 55: Descripción de la entidad PlanProgramacion.....	61
Tabla 56: Descripción de la entidad Programacion.....	61
Tabla 57: Descripción de la entidad Actividad	62
Tabla 58: Descripción de la entidad PlanProduccion	63
Tabla 59: Descripción de la entidad Equipamiento.....	64
Tabla 60: Descripción de la entidad ControlMovimiento	65
Tabla 61: Descripción de la entidad NArea	65
Tabla 62: Descripción de la entidad NTipoAtributo_Equipamiento	66
Tabla 63: Descripción de la entidad NTipoAtributo	66
Tabla 64: Descripción de la entidad NTipoEquipamiento	67
Tabla 65: Marco de prueba. Hardware.....	80
Tabla 66: Marco de prueba. Software	81

Índice

Introducción	1
Capítulo 1: Fundamentación teórica.....	5
Introducción	5
Conceptos asociados al dominio del problema	5
Objeto de estudio.....	7
<i>Descripción general</i>	7
<i>Descripción actual del dominio del problema</i>	7
Situación problemática.....	8
Análisis de otras soluciones existentes.....	9
Conclusiones	9
Capítulo 2: Tendencias y tecnologías actuales a desarrollar	10
Introducción	10
Tipos de sistemas de bases de datos	10
<i>Bases de datos jerárquicas</i>	11
<i>Bases de datos de red</i>	11
<i>Bases de datos relacionales</i>	11
<i>Bases de datos orientadas a objetos</i>	12
<i>Bases de datos objeto-relacionales</i>	12
<i>Bases de datos multidimensionales</i>	13
Técnicas para diseñar bases de datos relacionales	13
<i>Diseño descendente, análisis o descomposición</i>	13
<i>Síntesis relacional o procedimiento de síntesis</i>	14
<i>Técnica de diseño de base de datos elegida</i>	14
Gestores para sistemas de base de datos relacionales.....	14
MySQL.....	14
Microsoft SQL Server.....	15
Oracle	16

<i>PostgreSQL</i>	16
<i>Gestor de base de datos elegido</i>	18
Clientes para PostgreSQL.....	18
<i>EMS SQL Manager para PostgreSQL</i>	18
<i>PGAdmin 3</i>	19
Cliente de base de datos elegido	19
Herramientas CASE	20
<i>ERwin</i>	20
<i>Visual Paradigm</i>	20
<i>Rational Rose</i>	21
Herramienta CASE elegida	21
Conclusiones	21
Capítulo 3: Descripción y análisis de la solución propuesta.....	24
Introducción	24
Arquitectura y convenciones de la base de datos	24
<i>Descripción de las clases persistentes</i>	29
Diseño de la base de datos.....	44
<i>Modelo entidad-relación de la base de datos</i>	44
<i>Descripción de la estructura física de la base de datos</i>	45
Conclusiones	67
Capítulo 4: Validación del diseño realizado.....	69
Introducción	69
Validación teórica del diseño	69
<i>Integridad</i>	69
<i>Normalización</i>	72
<i>Análisis de redundancia de información</i>	76
<i>Análisis de la seguridad de la base de datos</i>	76
Validación funcional.....	77

<i>Uso de herramientas para un llenado voluminoso e inteligente de la base de datos</i>	77
<i>Búsqueda o diseño de herramientas para pruebas de carga intensiva</i>	80
Conclusiones: Valoración de resultados y propuesta de iteración	91
Conclusiones	93
Recomendaciones	94
Trabajos citados.....	95
Bibliografía	97
Anexos.....	99

Introducción

Desde que John Logie Baird el 26 de noviembre de 1926 hiciera la primera demostración del funcionamiento de un sistema de televisión a la Royal Institution de Londres, el desarrollo de este portento no se ha detenido, tanto es así que ha llegado a convertirse en uno de los medios líderes de la comunicación masiva, influyendo considerablemente en la formación del carácter de las personas, en su modo de pensar y de actuar.

La televisión llegó a Cuba en el año 1950, convirtiéndonos en uno de los primeros países latinoamericanos en contar con este privilegio, tanto es así que en 1952 de 9 ciudades latinoamericanas que brindaban este servicio 5 eran cubanas. En aquellos años las emisoras pertenecían a las clases económicamente dominantes que utilizaban el influjo de este medio de comunicación para transmitir patrones ideológicos y culturales acordes a sus beneficios, lo que traía consigo una programación altamente consumista plagada de elementos publicitarios.

El triunfo revolucionario dio un vuelco a esta situación implantando una programación más dedicada a aliviar las carencias informativas, educativas, culturales y de entretenimiento que había sufrido la población cubana hasta el momento, fomentando además valores éticos, sociales, políticos e ideológicos. (Castro, 2007)

Acorde a la estrategia asumida por el país, desde el primer año de vida de la Universidad de las Ciencias Informáticas (UCI) se crea la Dirección de Televisión Universitaria (DTU) como un medio de apoyo al proceso docente educativo y de cobertura informática para los estudiantes, lo que unido a la masiva automatización de los servicios y procesos convierten a este centro en una Gran Ciudad Digital.

El procesamiento del extenso volumen de información, que se gestiona diariamente en este centro de estudios, no siempre se efectúa con la calidad necesaria, por lo que no es posible que los servicios estén al nivel del desarrollo informático de la UCI. Por tanto se hace necesaria la automatización de los procesos de gestión de todos los servicios que se brindan en nuestra universidad.

Por su parte, la información que se procesa en la DTU sufre del mismo mal que la información que se procesa en buena parte de los demás sistemas y entidades que operan en la UCI: no está correctamente

gestionada ni usa en toda su magnitud las facilidades de la tecnología de la información en su favor. Cada departamento de esta Dirección gestiona un número de procesos particulares, y cada uno de estos pasa a través de un flujo de trabajo diferente. Estos procesos en la actualidad son manuales o se realizan por medio de aplicaciones que han quedado obsoletas debido a los cambios en la estructura de la organización, por tanto están muy propensos a errores. Dicha **situación problemática** en una entidad con el impacto que tiene la DTU en la población general del centro no puede pasar desapercibida.

Las debilidades mencionadas fueron objeto de estudio en (Alfonso Quesada, 2008), donde, partiendo del conocimiento de toda la información que se procesa en cada uno de los servicios audiovisuales que se prestan en la UCI, se realizó un estudio investigativo al respecto, identificándose los elementos y las soluciones para la implementación de un sistema de gestión de los procesos que se generan en la DTU. La solución definida en (Alfonso Quesada, 2008) estipula una aplicación Web con soporte de datos para resolver el problema de la gestión de los procesos. Dada la importancia de esta solución, para su aplicación práctica se precisa de la elaboración de una serie de instrumentos pertinentes, por lo que se ha decidido dividir su desarrollo en dos partes: la elaboración de la base de datos y el desarrollo de la aplicación Web propiamente dicha. Estas partes concurrirán en una fase posterior del proyecto, presentándose finalmente al cliente un producto integrado y listo para ser utilizado en el ambiente real. La satisfacción de la primera parte, la elaboración de la base de datos, constituye el **fundamento** y la **actualidad** de este trabajo de Diploma. ¿Cómo realizar el modelo lógico y físico de la base de datos para el Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria? constituye el **problema** a resolver.

Tomando en consideración todo lo expresado anteriormente se define como **objeto de estudio** a los procesos para la gestión de la información de los materiales audiovisuales de la UCI, cuyo **campo de acción** abarca la automatización del procesamiento de los datos que surgen como resultado de las necesidades informativas del trabajo de la DTU.

Lo planteado anteriormente lleva a la formulación del siguiente **objetivo**:

Objetivo General: Implementar el modelo físico de la base de datos del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria específicamente en sus cuatro departamentos: Dirección, Producción, Informativo y el de Programación.

Como premisa general para la elaboración del diseño de la base de datos se parte de la **hipótesis** siguiente:

La implementación de una base de datos para la gestión de los procesos de la Dirección de Televisión Universitaria permite una correcta organización y administración de la información existente y previene los problemas presentes en su manejo actualmente.

Este enunciado se convierte en la esencia del presente trabajo de diploma.

La demostración de los presupuestos señalados, el logro de los objetivos planteados y construir un producto con la calidad necesaria determinan la realización de las siguientes **Tareas de la Investigación:**

- Caracterizar el negocio
- Identificar las distintas técnicas de diseño de bases de datos a nivel mundial, nacional y en la Universidad.
- Analizar las tendencias y tecnologías actuales de bases de datos a nivel mundial.
- Realizar el diseño lógico de la base de datos
- Aplicar las técnicas de normalización y de-normalización en la bases de datos diseñada.
- Determinar la redundancia de la información en la base de datos.
- Implementar el modelo físico de la base de datos.
- Comprobar la integridad de los datos en la base de datos.
- Validar el funcionamiento de la base de datos.

Con cuya ejecución se esperan alcanzar los siguientes **resultados:**

- Diseño del modelo lógico.
- Diagrama de clases persistentes refinado.
- Modelo de entidad-relación.

- Implementación del modelo físico de la base de datos.

Los **métodos científicos** utilizados para darle solución al problema a resolver, demostrar la hipótesis y cumplimentar los objetivos son los siguientes:

Métodos Teóricos:

Análisis Histórico-Lógico: Utilizado para evaluar el desarrollo de las tendencias y tecnologías de creación de bases de datos a nivel mundial, nacional y en la UCI de manera que sea posible seleccionar la más adecuada para la resolución de nuestra problemática.

Método Inductivo-Deductivo: Utilizado para valorar los resultados obtenidos en diferentes áreas con la aplicación de distintas metodologías de diseño e implementación de bases de datos de forma que se pueda seleccionar la más adecuada para la creación de la base de datos del Sistema de Gestión para la Dirección de Televisión Universitaria (SGPDTU).

Modelación: Utilizado para obtener el modelo lógico, el modelo de datos y el diagrama de clases persistentes.

Métodos Empíricos:

Entrevista: Utilizado para hacer un levantamiento de las características del negocio con el objetivo de determinar las clases persistentes y las entidades, además de determinar cuáles son las principales tendencias en el desarrollo de bases de datos en la universidad.

Capítulo 1: Fundamentación teórica

Introducción

En este capítulo se abordan los principales conceptos relacionados con el dominio de la solución propuesta en el marco de la DTU. También se analiza en detalle el objeto de estudio del trabajo, la situación problemática existente y un breve análisis de otras soluciones existentes dentro de la gestión de procesos de la DTU.

Conceptos asociados al dominio del problema

Televisión (TV): transmisión instantánea de imágenes, tales como fotos o escenas, fijas o en movimiento, por medios electrónicos a través de líneas de transmisión eléctricas o radiación electromagnética (ondas de radio).

Producción televisiva: técnicas empleadas para crear un programa de televisión.

Subprocesos: son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso.

Sistema: estructura organizativa, procedimientos, procesos y recursos necesarios para implantar una gestión determinada, como por ejemplo la gestión de la calidad, la gestión del medio ambiente o la gestión de la prevención de riesgos laborales. Normalmente están basados en una norma de reconocimiento internacional que tiene como finalidad servir de herramienta de gestión en el aseguramiento de los procesos.

Web: sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (*el web, un web*) se refiere a un sitio web entero, en cambio si se utiliza en femenino (*la web, una web*) se refiere a una página web concreta dentro del sitio web.

Información Audiovisual: contenido de todas las tipologías audiovisuales (teleclases, audiovisuales varios, filmes, series, fotos, música, videos musicales, documentales, programas, efectos de sonido).

Solicitudes de Información Audiovisual: solicitudes que realizan las entidades externas en función de sus actividades, así como las solicitudes que se le hacen a las instituciones afines en correspondencia con las necesidades del Centro de Gestión de la Información Audiovisual (CGIA) para satisfacer las demandas.

Solicitudes de Información Audiovisual Aprobadas: solicitudes aprobadas por los directivos de la DTU, que se envían al CGIA.

Información Audiovisual Solicitada: respuesta de su existencia o no, en el fondo del CGIA.

Información Audiovisual Recuperada: si existe en el fondo del CGIA, se revisa y está en buen estado, se entrega a la entidad que la solicitó.

Información Audiovisual a Editar: si existe en el fondo del CGIA, se revisa y no está en buen estado, se solicita edición o reedición a Producción.

Información Audiovisual Docente: teleclases y materiales complementarios docentes.

Plan de producción: mecanismo de control consistente en una tabla de tiempo donde se planifican las tareas a cumplir semanalmente juntos con los recursos asignados y las personas responsables.

Desiderata: listados de audiovisuales que no existen en el fondo del CGIA, surgen a partir de estudios de necesidades y las solicitudes.

Información Audiovisual Editada: se le ha aplicado todos los procesos de producción y está listo para procesar.

Información Audiovisual Procesada: audiovisuales descargados del servidor, analizados y procesados por los especialistas.

Información Audiovisual en DVD: audiovisuales quemados en DVD y almacenados en el Fondo del CGIA, listos para ser consultados y recuperados para prestar servicios.

Objeto de estudio

Descripción general

Desde el comienzo de la universidad se concibió la televisión como un medio informativo importante en el desarrollo de muchas esferas de la vida de los estudiantes y profesores. Este medio ha servido para difundir, enseñar y entretener a la comunidad universitaria, siempre de acuerdo con los principios que defiende la institución. Unido al desarrollo de la informática y las comunicaciones este medio ha alcanzado grandes logros, llegando a convertirse en un medio de comunicación por excelencia. Debido a la gran matrícula del centro la cual fue creciendo año tras año, y a la necesidad de contar con un sistema propio que se encargara de todo lo relacionado con las teleclases, mantener informada a la comunidad universitaria, recoger y archivar los eventos y las visitas más importantes a la universidad entre otras tareas se crea la Dirección de Televisión Universitaria (DTU). (Alfonso Quesada, 2008)

Descripción actual del dominio del problema

La Dirección de Televisión de la UCI está dividida en 4 grandes departamentos: Producción, Informativo, Programación y la Dirección, entre los cuales existe un gran flujo de actividades que son las que hacen funcionar correctamente la DTU. En la DTU se lleva un control general de todos los medios asignados a la dirección, desde cables, micrófonos, cámaras, hasta las computadoras con las que trabaja el personal de la dirección.

El departamento de Producción es el encargado de filmar y producir los materiales audiovisuales, y además de confeccionar el plan de producción.

El departamento Informativo es el encargado de recibir todas las informaciones provenientes de la dirección de la Universidad, de las facultades, de los miembros del Movimiento Código y Letra y de los estudiantes y trabajadores que de forma general envían avisos, reportajes, artículos y noticias para ser publicadas en la Intranet y Señal 3 fundamentalmente.

El departamento de Programación está dividido en 3 áreas:

- Área de Programación: encargada de conformar semanalmente la programación del canal 7 a partir de las efemérides de la semana, las películas de estreno y los materiales del archivo.

- Centro de Gestión de Información Audiovisual (CGIA): la cual brinda servicios de digitalización y compresión de materiales, catalogación de media por tipología y gestión de recursos, que puede comprender la obtención del mismo fuera de la UCI o a través del satélite. En el CGIA se lleva además el control de las solicitudes externas e internas, y el archivado de los recursos audiovisuales disponible.
- Área de Transmisiones: Confecciona y gestiona la lista de reproducción para los canales internos y además graba los programas de la televisión cubana e internacional.

El departamento de Dirección se encarga de la coordinación y el buen funcionamiento del resto de los departamentos, además de la planificación estratégica de la evolución de la entidad para la mejor sincronización con el resto de la Universidad y sus clientes. Por esto, el departamento de dirección solo se beneficia del sistema proyectado en concepto de reportes obtenidos sobre el trabajo de los demás departamentos.

Al momento de la confección de este trabajo la DTU se encuentra en un proceso de estudio para la incorporación de los elementos de la radio universitaria. Como aun no se ha logrado un consenso en las propiedades de dicha absorción, este tema no es objetivo de automatización por parte del sistema modelado.

Situación problemática

La situación problemática existente está dada por condiciones deficientes de trabajo en la DTU, debido al volumen de trabajo que maneja. Actualmente la DTU cuenta con su propia televisora, donde se producen diferentes tipos de materiales, se prestan servicios de digitalización de materiales, búsqueda y recopilación de información audiovisual, préstamos de materiales y otros servicios dentro y fuera de la Universidad.

El aumento de esta información a medida que pasa el tiempo debido al desarrollo de la institución se une con fallos en la distribución de los planes de trabajo, producción o programación del canal. (Alfonso Quesada, 2008)

Los datos actualmente se gestionan usando varios sistemas no integrados y herramientas tradicionales de ofimática, lo cual no ayuda a la comunicación interdepartamental y además proporciona un ambiente

favorable para la existencia de datos repetidos en los sistemas. La recuperación de información y su posterior análisis se ven entorpecidas precisamente por la no disponibilidad de una sola fuente de donde extraerla, lo cual provoca que los datos no siempre estén accesibles o sean relevantes.

Análisis de otras soluciones existentes

Como resultado del estudio de la bibliografía consultada no se encontró ningún sistema con estas características. Solamente existe uno que surgió como producto de una tesis del curso anterior, el cual es un sistema de gestión de procesos pero solamente de una de las áreas de uno de los 4 departamentos principales de la DTU. Dicho sistema no sustituye el proyectado debido a su limitado alcance, aunque está actualmente en explotación en el CGIA del departamento de Programación.

Conclusiones

Teniendo en cuenta los elementos analizados se puede concluir que la DTU tiene un nivel muy bajo de automatización, que permite la ocurrencia frecuente de errores humanos que minan su eficiencia. La construcción de la aplicación propuesta en (Alfonso Quesada, 2008), soportada por una fuente de datos debidamente diseñada puede ayudar a resolver este problema. Dicha solución debe ser integral, y estar dirigida a todas las áreas de la DTU, logrando así una gestión de procesos orgánica y una centralización total de todos los datos.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar

Introducción

Para la determinación del ambiente de trabajo a confeccionar se han dividido las herramientas necesarias en cuatro categorías, de las cuales se debe seleccionar una tecnología particular:

- Tipos de base de datos
- Gestor de base de datos
- Cliente de base de datos
- Herramientas CASE

Antes de escoger una herramienta particular dentro de cada una de estas categorías se debe primero seleccionar la tecnología de la base de datos, la cual se corresponde a su tipo. Por consiguiente el primer estudio se centra en la tecnología subyacente de la fuente de datos en sí. Para ello y para la selección general de las tecnologías a usar se reutilizará un estudio publicado en (Veliz Díaz, 2008), del cual se da un resumen de sus resultados. Dicho estudio ha sido ajustado debido a que presentaba cierta obsolescencia en las características enunciadas en algunas tecnologías, que han evolucionado desde entonces. Sin embargo se considera que es válido para las decisiones que deben tomarse sobre él.

Tipos de sistemas de bases de datos

Los modelos de bases de datos más extendidos son el modelo relacional y objeto relacional. El modelo orientado a objetos es también muy popular, pero aun está sujeto a mucha discusión su rendimiento frente a sistemas tradicionales de modelo relacional. Con el desarrollo de la tecnología han aparecido los modelos objeto-relacional y multidimensionales, pero estos aun son muy modernos como para tener una cuota de mercado significativa que los haga tener una comunidad influyente para proyectos pequeños o de misión crítica.

Bases de datos jerárquicas

Las Bases de Datos Jerárquicas nacieron a mediados de 1960 y como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol. Son muy útiles para sistemas que administran grandes volúmenes de información y datos muy compartidos lo que permite obtener estructuras estables y de gran rendimiento. (Veliz Díaz, 2008)

Las principales limitaciones de este modelo son su incapacidad de brindar eficientemente una solución a la redundancia de datos y la baja integridad referencial (es posible eliminar un nodo padre sin haber eliminado sus hijos asociados).

Este modelo logró un gran éxito en el mercado, pero fue suplantado por el modelo relacional y por tanto se ha convertido en una tecnología obsoleta.

Bases de datos de red

Es un modelo establecido sobre las bases del modelo jerárquico. La diferencia fundamental radica en la evolución del concepto de nodo, permitiendo que tenga varios padres, algo que resultaba imposible en el modelo jerárquico. Esto brindaba una solución eficaz a la dificultad de la redundancia de datos, pero la dificultad que implicaba gestionar la información en una Base de Datos de Red, provocó que su uso se popularizara más entre programadores que entre usuarios finales. (Wikipedia, Base de datos, 2009)

Actualmente está también superado como tecnología de base de datos debido a la popularidad del modelo relacional.

Bases de datos relacionales

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de entidades (tablas), compuestas por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla). En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de “consultas” que ofrecen una amplia flexibilidad y poder para administrarla. (Hansen, y otros)

La principal dificultad de estos sistemas es lo que se conoce como *object-relational mismatch*, que es un problema concerniente a la diferencia de representación de la información en un modelo de objetos y en un modelo relacional. Esta dificultad ha sido reconocida y solventada por un número considerable de herramientas de desarrollo en todas las tecnologías principales.

Bases de datos orientadas a objetos

A finales de los años 80 del siglo pasado aparecieron las primeras Bases de Datos Orientadas a Objetos (BDOO). Están diseñadas para ser eficaces para el almacenamiento de objetos complejos y métodos. Este modelo es bastante reciente y se basa en los modelos informáticos orientados a objetos. Combina las mejores cualidades de los archivos planos, las Bases de Datos Jerárquicas y Relacionales. Las BDOO representan el siguiente paso en la evolución de las bases de datos para soportar análisis, diseño y programación orientada a objetos. Ofrecen un mejor rendimiento de la máquina que las Bases de Datos Relacionales, para aplicaciones o clases con estructuras complejas de datos. (Veliz Díaz, 2008)

Sin embargo las BDOO, si bien resuelven el problema del *object-relational mismatch* también requieren más hardware que las bases de datos relacionales tradicionales, y sus ventajas solo se aprecian de forma considerable en aplicaciones con modelos de objetos complejos. Aplicación de modelos relativamente sencillos no se benefician de sus ventajas pero si sufren sus desventajas, especialmente debido a que es una tecnología nueva y que no se tiene experiencia suficiente con los errores comunes ni con la tecnología a utilizar.

Bases de datos objeto-relacionales

Las Bases de Datos Objeto Relacionales son base de datos que desde el modelo relacional evolucionan hacia bases de datos más extensas y complejas, incorporando conceptos del modelo orientado a objetos. Se puede afirmar que un Sistema Gestor de Bases de Datos Objeto-Relacional (SGBDOR) contiene dos tecnologías; la tecnología relacional y la tecnología de objetos. (Díaz Villanueva, 2002)

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

Bases de datos multidimensionales

Un enfoque que ha cobrado actualmente fuerza es el procesamiento analítico en línea (en inglés *On-Line Analytical Processing*, OLAP), que ha desarrollado una tecnología de Bases de Datos Multidimensionales. Estas bases de datos fundan los cimientos para el desarrollo de los cálculos y análisis multidimensionales que requiere la inteligencia empresarial. Estos modelos tienen gran aplicación en la bioinformática, debido a la posibilidad de almacenamiento de grandes cantidades de información biológica. Esta tecnología es muy avanzada y costosa, y por tanto solo se utiliza en sistemas empresariales para los cuales sea absolutamente necesario. (Veliz Díaz, 2008)

Técnicas para diseñar bases de datos relacionales

Existen dos escuelas que implican dos grupos de algoritmos a la hora de aplicar la Teoría de Normalización. Una son los métodos de Análisis o Descomposición y la otra los procedimientos de Síntesis que a continuación se detallan.

Diseño descendente, análisis o descomposición

Estos métodos parten de una relación llamada Universal que contiene a todos los atributos del universo del discurso y las dependencias funcionales existentes entre ellos. Por medio de descomposiciones sucesivas y cumpliendo los principios de conservación de la información y de las dependencias resultan esquemas de menor grado en formas normales cada vez más avanzadas, por lo que se puede garantizar que se reducen las anomalías.

Se pueden aplicar estos métodos también a cada una de las relaciones obtenidas a partir de un esquema conceptual en un modelo de datos de alto nivel (Modelo ER) y luego transformar el esquema conceptual a un conjunto de relaciones empleando un procedimiento de transformación. También se puede aplicar el método a cada una de las relaciones que se obtienen de la transformación del Diagrama Entidad Relación (DER) al modelo relacional.

El objetivo inicial que se pretende con estos métodos es separar la información referente a un objeto o entidad diferente en una relación o conjunto de relaciones propio. (Veliz Díaz, 2008)

Síntesis relacional o procedimiento de síntesis

Este es un método de diseño alternativo a la descomposición, resulta ser un enfoque más purista, implica contemplar el diseño de esquema de base de datos relacionales estrictamente en términos de dependencias funcionales y otros tipos especificadas para los atributos de la base de datos. Aquí los esquemas de relación en Tercera Forma Normal (3FN) ó Forma Normal de Boyce-Codd (FNBC) se sintetizan gradualmente agrupando los atributos apropiados. Recorre un camino inverso a la descomposición, es decir, busca agrupar atributos a fin de tener en una relación toda la información correspondiente a un objeto o entidad. (Veliz Díaz, 2008)

Técnica de diseño de base de datos elegida

De entre ambas técnicas analizadas se elige la técnica de diseño descendente, por ser más conocido que la alternativa. Ha sido probado en innumerables proyectos de alcance semejante al presente para producir bases de datos relacionales. Esta elección reduce riesgos por conceptos de tecnología desconocida y por concepto de tiempo de aprendizaje, ya que es la técnica de diseño más usada en el entorno en que se diseña esta base de datos actualmente.

Gestores para sistemas de base de datos relacionales

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU (*General Public License*). Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Aunque MySQL es software libre, se distribuye una versión comercial de MySQL, que solo se diferencia de la versión libre en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida en parte a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (Pecos, 2002)

Ventajas:

- La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.

Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Entre sus ventajas están:

- Soporte de transacciones.
- Gran estabilidad y seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos Lenguaje de Definición de Datos (DDL) y Lenguaje de Manipulación de Datos (DML) gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos. (Microsoft, 2006)

Desventajas:

- Tiene soporte solamente en el sistema operativo Windows.
- Es un software de licencia propietaria.

Oracle

Es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. (Rincón Rincón, 2004)

Ventajas:

- Soporte multiplataforma.
- Oracle soporta bases de datos de todos los tamaños.
- Soporte de transacciones.
- Es mucho más fácil la realización de backups (copias de seguridad).
- Es mucho más eficaz y eficiente.
- Estabilidad.
- Tiene una amplia gama de herramientas para operar con la base de datos tanto como usuario y como Administrador.

Desventajas:

- Elevado precio comercial.
- Exige una gran cantidad de recursos de la máquina donde se encuentre instalado.

PostgreSQL

PostgreSQL es uno de los Sistemas Gestores de Bases de Datos más utilizados por la comunidad de software libre. Está basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (*OpenSource*) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los

conceptos existentes en el Sistema Objeto-Relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. Incluye características de la orientación a objetos, como son herencia, tipos de datos, funciones, restricciones. (Pecos, 2002)

Lleva más de una década de desarrollo, siendo hoy en día un sistema bastante avanzado, que tiene soporte nativo para los lenguajes de programación: C, C++, Java, Python, PHP y muchos más. Se encuentra bajo la licencia BSD (*Berkeley Software Distribution*).

Entre otras características que presenta se encuentran las siguientes:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. Además permite la creación de tipos de datos propios.
- Reúne funciones de disímiles características: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias y triggers.
- Incluye herencia entre tablas.
- Gestión de múltiples usuarios y sus respectivos permisos.

PostgreSQL es el servidor de bases de datos de software libre más poderoso que hay actualmente y constituye por tanto una opción viable ante MySQL cuando se requieren opciones más avanzadas.

Ventajas:

- Implementa el uso de rollback, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Es capaz de validar la integridad referencial además de almacenar procedimientos en la propia base de datos.
- PostgreSQL admite que un proceso escribe en una tabla al tiempo que otros acceden a la misma sin que ocurran bloqueos.
- Es capaz de ajustarse al número de procesadores y al total de memoria que tiene el sistema de forma óptima, siendo capaz de gestionar un gran número de peticiones simultáneas sin errores.

Desventajas:

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento. (Pecos, 2002)
- Alto consumo de recursos de la máquina.
- Baja velocidad.

Gestor de base de datos elegido

Se seleccionó PostgreSQL como gestor de base de datos a pesar de que MySQL era una alternativa igualmente posible, no solo porque las políticas de la institución promueven el uso del mismo sino además por los beneficios que reporta y que son necesarios en el entorno de desarrollo, como por ejemplo su escalabilidad (necesaria dado que existe la posibilidad de un aumento en el número de estaciones de trabajo) y la capacidad de comprobar la integridad referencial.

Clientes para PostgreSQL

EMS SQL Manager para PostgreSQL

El EMS SQL Manager es una herramienta gráfica fácil de utilizar para la administración de PostgreSQL. Trabaja con cualquiera de sus versiones hasta la 8.1 y soporta todas las últimas características de PostgreSQL, incluyendo espacios de tablas, nombres de argumentos en funciones y más. El programa ofrece muchas herramientas poderosas para usuarios experimentados, como un Diseñador Visual de Bases de Datos y un Constructor Visual de Consultas. Es la más completa hasta la actualidad. Permite obtener la documentación sobre el diseño de la base de datos y tiene un grupo de herramientas importantes para la importación y exportación de datos. (Veliz Díaz, 2008)

Características:

- Soporte completo para PostgreSQL hasta la versión 8.1.
- Administración y navegación rápida de bases de datos.
- Administración fácil de todos los objetos PostgreSQL.

- Herramientas de manipulación avanzada de datos.
- Administración efectiva de seguridad.
- Excelentes herramientas visuales y de texto para la construcción de consultas.
- Capacidades de exportación e importación de datos.
- Poderoso diseñador visual de bases de datos.
- Modo guiado para labores de mantenimiento.
- Interfaz atractiva.

PGAdmin 3

Es una interfaz comprensible para el diseño y administración de una base de datos PostgreSQL, diseñada para ejecutarse en la mayoría de los Sistemas Operativos. La aplicación corre bajo GNU/Linux, FreeBSD y Windows 2000/XP. (Espinosa, 2005)

Características:

- Un agente de SQL/shell para tareas programadas.
- Soporte para PostgreSQL
- Roles.
- Procedimientos almacenados.
- Transacciones preparadas.
- Permite registrar un servidor sin establecer conexión.
- Ayuda para usuarios nuevos.

Cliente de base de datos elegido

El EMS PostgreSQL Manager se selecciona a pesar de ser software propietario, su facilidad de uso y riqueza de opciones no tiene rival en ninguna de las alternativas presentes. Las funciones avanzadas de

administración que no posee no serán necesarias para el desarrollo del producto, así que no influyen en la elección.

Herramientas CASE

Para confeccionar una base de datos se necesitan herramientas que posibiliten garantizar la seguridad, consistencia e integridad de la información, así como lograr que todo el proceso de su creación sea lo más sencillo y rápido posible.

Existe una gran variedad de herramientas que se encargan del diseño de las bases de datos. Para el estudio del estado del arte solo se tomaron en cuenta aquellas más conocidas en el entorno universitario: ERwin, Visual Paradigm y Rational Rose, en sus últimas versiones conocidas.

ERwin

Brinda productividad en diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas y cualquier otro objeto de la base de datos necesario. Soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase, DB2, e Informix. (Veliz Díaz, 2008)

ERwin en su versión 7.5.1 incluye además soporte para la generación de modelos físicos de datos para bases de datos en PostgreSQL.

Visual Paradigm

Es una herramienta CASE que ofrece un entorno de creación de diagramas para UML. Esta herramienta utiliza un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. El diseño es centrado en casos de uso y enfocado al negocio, lo que permite generar software de gran calidad. Tiene capacidad para la ingeniería directa e inversa en Java, C++, PHP, entre otros lenguajes y disponibilidad de múltiples versiones para cada necesidad. Es multiplataforma y muy útil para la generación de código fuente en PHP. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la de base de datos a partir del esquema de clases. Incorpora el soporte para trabajo en equipo, proporcionando que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros de equipo.

Esta herramienta soporta bases de datos que incluyen PostgreSQL, SB2, MySQL, DB2, Informix, Microsoft SQL Server, Oracle.

Rational Rose

Es una herramienta CASE que brinda un entorno de creación de diagramas para UML. Usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. El diseño es centrado en casos de uso y enfocado al negocio, lo que permite generar software de gran calidad. Tiene la capacidad de crear el esquema de datos a partir del esquema de clases. Este software es muy conocido en nuestra institución debido a que forma parte del curso estándar de ingeniería de software. Tiene la desventaja de ser un software muy anticuado, difícil de trabajar y poco intuitivo, además de que su función primaria no es la de servir de CASE para procesos de diseño de bases de datos sino que proporciona esta funcionalidad de forma auxiliar. Además de ello el software no soporta las versiones actuales de los gestores de datos más populares cubiertos por el estudio presentado.

Herramienta CASE elegida

De los anteriores se selecciona el Visual Paradigm, por ser de las tres herramientas analizadas la única de la que se posee una licencia y que se conoce existen versiones para sistemas libres. Además es una herramienta bastante intuitiva y flexible, que proporciona las funcionalidades necesarias para un diseño profesional de base de datos como el necesario en este proyecto.

Conclusiones

El perfil tecnológico queda luego del estudio analizado anteriormente:

Tabla 1: Perfil tecnológico

Categoría	Tecnología o Herramienta	Razones
Técnica de diseño de base de datos	Diseño Descendente, análisis o descomposición	Más conocido que la alternativa. Probado en innumerables proyectos de alcance semejante al proyectado para producir bases de datos relacionales. Esta elección reduce riesgos por conceptos de tecnología desconocida y por concepto de tiempo de aprendizaje, ya que es la técnica de diseño más usada en el entorno en que se diseña esta base de datos actualmente.
Herramientas CASE	Visual Paradigm	De las tres herramientas analizadas, Visual Paradigm es la única de la que se posee una licencia y que se conoce existen versiones para sistemas libres. Además es una herramienta bastante intuitiva y flexible, que proporciona las funcionalidades necesarias para un diseño profesional de base de datos como el necesario en este proyecto.
Gestor de base de datos	PostgreSQL	Se seleccionó PostgreSQL como gestor de base de datos a pesar de que MySQL era una alternativa igualmente posible, no solo porque las políticas de la institución promueven el uso del mismo sino además por los beneficios que reporta y que son necesarios en el entorno de desarrollo, como por ejemplo su escalabilidad (necesaria dado que existe la posibilidad de un aumento en el número de estaciones de trabajo) y la

		capacidad de comprobar la integridad referencial.
Cliente de base de datos	EMS PostgreSQL Manager	A pesar de ser software propietario, su facilidad de uso y riqueza de opciones no tiene rival en ninguna de las alternativas presentes. Las funciones avanzadas de administración que no posee no serán necesarias para el desarrollo del producto, así que no influyen en la elección.

Capítulo 3: Descripción y análisis de la solución propuesta

Introducción

En este capítulo se hace un bosquejo de las características de la base de datos en cuanto a diseño y arquitectura. Se presentan los artefactos de diseño utilizados para dar respuesta a los requisitos levantados y a las necesidades de la aplicación asociada a la base de datos diseñada.

Arquitectura y convenciones de la base de datos

La arquitectura en el marco de este trabajo se define como el conjunto de convenciones y normas a respetar para facilitar el trabajo de desarrollo y mantenimiento de los objetos en la base de datos. Dichas normas abarcan desde las puramente tecnológicas hasta las más triviales como pueden ser nomenclaturas de objetos o convenciones de tamaño para cada tipo de datos según la función que realice.

La elección de herramientas para el desarrollo de la base de datos fue expuesta y debidamente justificada en el capítulo anterior por lo cual no volverá a ser mencionada aquí.

Ambiente de producción

El ambiente donde será explotado el producto final del desarrollo tiene unas características no elegibles por los desarrolladores, sino afines a las políticas institucionales, sobre todo en materia de seguridad:

- El sistema operativo a usar en los servidores será Ubuntu.
- El servidor de datos siempre estará protegido por el firewall NetFilter/IPTables propio de cualquier distribución Linux.
- El control de acceso al servidor se realizará mediante los archivos de configuración de postgresql (postgresql.conf y pg_hba.conf). Estos archivos regulan todas las variables necesarias para controlar las IP con permisos para acceder al servidor de datos, los usuarios y la cantidad de conexiones aceptadas. También puede regularse el consumo de memoria y CPU y otras consideraciones de interés en cuanto al impacto del servidor de datos en el hardware donde se despliegue.

- **Concurrencia Máxima:** No fue especificado en los requisitos no funcionales recogidos ningún límite específico de concurrencia. Sin embargo debido al poco volumen de personal existente en la DTU, se puede inferir una concurrencia máxima de 100 usuarios, asumiendo el peor caso posible de que todos los trabajadores se conectaran al unísono al servidor. En este escenario, por tanto, no se prevé la necesidad de modificar las configuraciones por defecto del servidor PostgreSQL, que al menos en su versión 8.2 viene preparado para soportar exactamente 100 conexiones concurrentes por defecto.

Los permisos en la base de datos, esquemas y objetos asociados, se realizarán a través de la creación de roles y grupos de roles, para así darle los permisos de lectura y escritura en dependencia de la responsabilidad o funcionalidad necesaria. Esto es sin perjuicio del acceso a los datos por medio de la aplicación correspondiente.

Organización dentro de la base de datos

Esquemas: Se utilizaron esquemas para aislar las tablas en la base de datos correspondientes a áreas funcionales dentro del negocio levantado, aunque no necesariamente a las áreas administrativas en las que está dividida la DTU. Estas áreas funcionales están compuestas por un conjunto de tablas que representan una actividad específica del negocio o del sistema, y de las que se identificaron 4: Seguridad, Media, Planificación e Inventario. Una explicación más detallada de estas funciones y las tablas que contienen se ofrece más adelante.

Convención de nombres

Base de datos

El nombre de la Base de Datos se escribe con todas las letras en minúsculas. En caso de ser un nombre compuesto las palabras se separan por “_”:

Ej. sgpdtu.

Esquemas

El nombre a emplear para los esquemas se escribe con todas las letras en minúscula, comenzando por el prefijo sch, a continuación el símbolo “_”, y por último el nombre del mismo:

Ej. sch_seguridad.

Tablas

El nombre a emplear para las tablas se escribe con todas las letras en minúscula comenzando por el prefijo correspondiente y las palabras separadas por el símbolo “_”:

Ej. audio_visual_varios.

Nomencladores

Las tablas que representen nomencladores de la aplicación comenzaran su nombre por el prefijo **nom**, seguido por su nombre según la regla de nomenclatura estándar para las tablas.

Ej: nom_asignatura.

Campos

El nombre a emplear para los campos debe ser intuitivo, evitar las abreviaturas y escribirse con todas las letras en minúscula comenzando por el prefijo correspondiente y las palabras separadas por el símbolo “_”: Si el campo es un identificador de llave primaria debe ser **id**: Si es un identificador de llave foránea debe estar constituido por id y el nombre de la tabla de la cual se extrae la llave, separados por “_”.

Ej. id_formato.

Tipos de datos

Se utilizaran los tipos de datos siguientes:

Para los tipos de datos cadena, que constituyen la inmensa mayoría de los campos de la base de datos, se utiliza varchar con diferentes longitudes según la necesidad:

varchar(20): para campos de tipo nombre o contraseña

varchar(30): para todo tipo de campos con funciones disimiles.

varchar(60): para campos que contienen nombre con apellidos o campos con valores que se prevean demasiado extensos para un *varchar(30)*.

varchar(300): descripciones, observaciones y comentario.

integer(10): para cualquier valor de tipo numérico, incluyendo llaves primarias y foráneas. No existen valores de tipo fraccionario así que no es necesario otro tipo de dato numérico. Tampoco existen valores booleanos que deban simularse mediante columnas numéricas.

Date: Para los campos fecha donde el día y el mes son lo único importante.

Timestamp: Para los campos donde sea necesario conocer la hora además de la fecha.

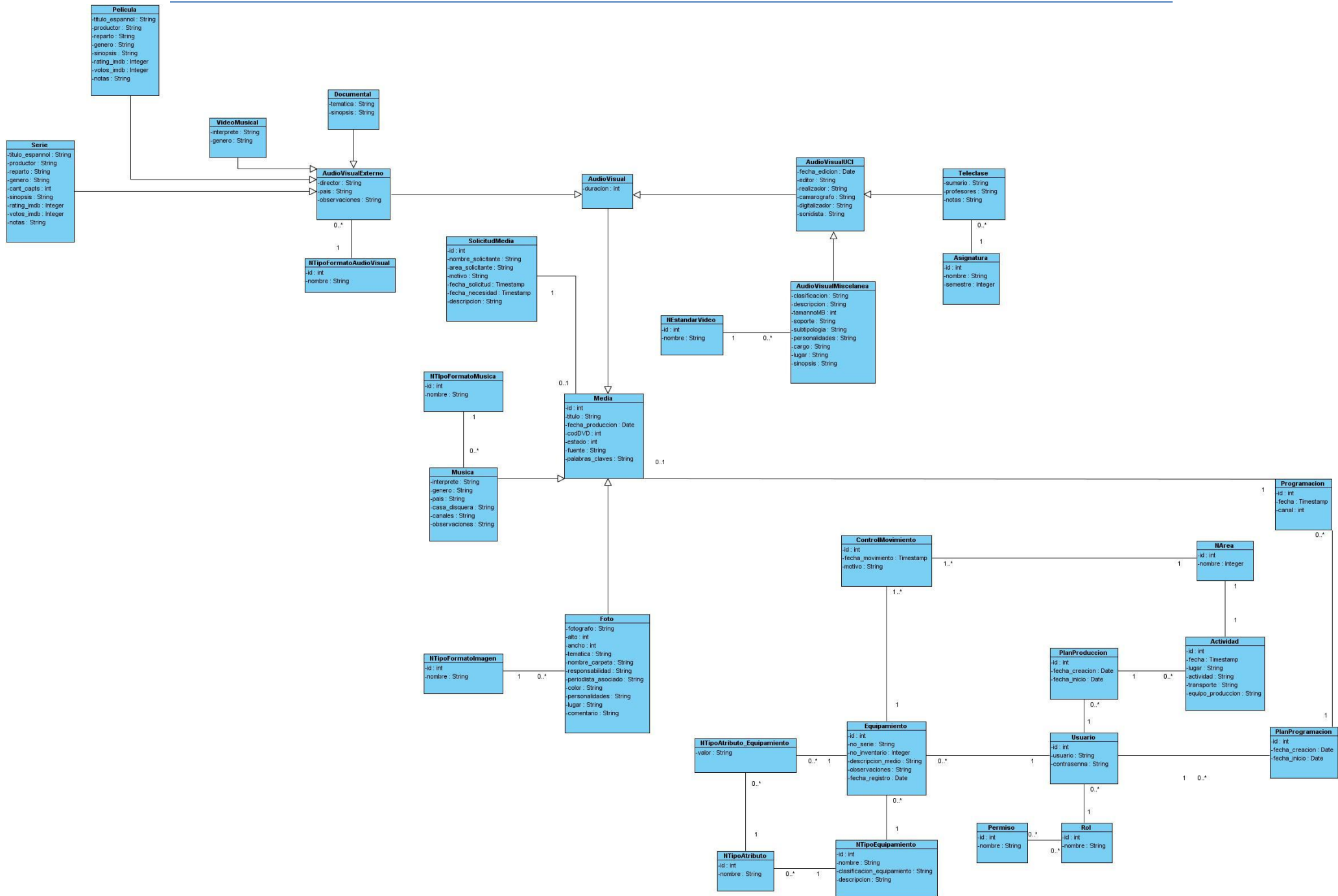
Secuencias autoincrementales

Para todas las llaves primarias se utiliza la secuencia autodefinida por PostgreSQL, debido a que todas las llaves son autoincrementales.

Modelo de clases persistentes

Figura 1: Modelo de clases persistentes

Capítulo 3: Descripción y análisis de la solución propuesta



Descripción de las clases persistentes

Tabla 2: Descripción de la clase Equipamiento

Nombre: Equipamiento	
Descripción: Contiene la información de los equipamientos inventariados de la DTU.	
Atributo	Tipo
id	Int
no_serie	String
no_inventario	Int
descripción_medio	String
observaciones	String
fecha_registro	Date

Tabla 3: Descripción de la clase ControlMovimiento

Nombre: ControlMovimiento	
Descripción: Contiene los movimientos de un medio inventariado por diferentes áreas de la DTU.	
Atributo	Tipo
id	Int
fecha_movimiento	Timestamp
motivo	String

Tabla 4: Descripción de la clase NArea

Nombre: NArea	
Descripción: Nomenclador para representar las áreas de la DTU.	
Atributo	Tipo
id	Int
nombre	String

Tabla 5: Descripción de la clase Media

Nombre: Media	
Descripción: Representa cualquier archivo audiovisual con el que se trabaje en la DTU.	
Atributo	Tipo
id	Int
codDVD	Int
titulo	String
fecha_produccion	Date
fuentes	String
estado	Int
palabras_claves	String

Tabla 6: Descripción de la clase Documental

Nombre: Documental	
Descripción: Contiene la información a guardar de los documentales con los que se trabaja en la DTU.	
Atributo	Tipo
tematica	String
sinopsis	String

Tabla 7: Descripción de la clase Teleclase

Nombre: Teleclase	
Descripción: Contiene la información a guardar de las teleclases con las que se trabaja en la DTU.	
Atributo	Tipo
sumario	String
profesores	String
notas	String

Tabla 8: Descripción de la clase Asignatura

Nombre: Asignatura	
Descripción: Contiene la información a guardar de las asignaturas que tienen teleclases.	
Atributo	Tipo

id	Int
nombre	String
semestre	Int

Tabla 9: Descripción de la clase AudioVisualMiscelanea

Nombre: AudioVisualMiscelanea	
Descripción: Representa cualquier material audiovisual que se produce en la UCI que no clasifique como teleclase.	
Atributo	Tipo
clasificacion	String
descripcion	String
tamannoMB	Int
soporte	String
subtipologia	String
personalidades	String
cargo	String
lugar	String
sinopsis	String

Tabla 10: Descripción de la clase VideoMusical

Nombre: VideoMusical	
Descripción: Contiene la información a guardar de los videos musicales con los que se trabaja en la DTU.	
Atributo	Tipo
interprete	String
genero	String

Tabla 11: Descripción de la clase Foto

Nombre: Foto	
Descripción: Contiene la información a guardar de las imágenes con las que se trabaja en la DTU.	
Atributo	Tipo
fotografo	String
nombre_carpeta	String
responsabilidad	String
periodista_asociado	String
alto	Int
ancho	Int
color	String

tematica	String
personalidades	String
lugar	String
comentario	String

Tabla 12: Descripción de la clase Pelicula

Nombre: Pelicula	
Descripción: Contiene la información a guardar de las películas con las que se trabaja en la DTU.	
Atributo	Tipo
productor	String
titulo_espannol	String
reparto	String
genero	String
sinopsis	String
rating_imdb	Int
votos_imdb	Int
notas	String

Tabla 13: Descripción de la clase Serie

Nombre: Serie	
Descripción: Contiene la información a guardar de las series con las que se trabaja en la DTU.	
Atributo	Tipo
cant_capt	Int
titulo_espanol	String
productor	String
reparto	String
genero	String
sinopsis	String
rating_imdb	Int
votos_imdb	Int
notas	String

Tabla 14: Descripción de la clase Musica

Nombre: Musica	
Descripción: Contiene la información a guardar de la música con la que se trabaja en la DTU.	
Atributo	Tipo
canales	String

genero	String
pais	String
casa_disquera	String
interprete	String
observaciones	String

Tabla 15: Descripción de la clase NTipoFormatoAudioVisual

Nombre: NTipoFormatoAudioVisual	
Descripción: Nomenclador para guardar los tipos de formatos de materiales audiovisuales con los que se trabaja en la DTU.	
Atributo	Tipo
id	Int
nombre	String

Tabla 16: Descripción de la clase SolicitudMedia

Nombre: SolicitudMedia	
Descripción: Representa la solicitud que se le hace al CGIA de una media determinada.	
Atributo	Tipo
id	Int
nombre_solicitante	String

area_solicitante	String
motivo	String
fecha_solicitud	Timestamp
fecha_necesidad	Timestamp
descripcion	String

Tabla 17: Descripción de la clase PlanProduccion

Nombre: PlanProduccion	
Descripción: Representa el cronograma de las actividades a las que se le dará cobertura semanalmente por parte de la DTU.	
Atributo	Tipo
id	Int
fecha_inicio	Date
fecha_creacion	Date

Tabla 18: Descripción de la clase Actividad

Nombre: Actividad	
Descripción: Representa las actividades a las que la DTU les dará cobertura como parte del plan de producción.	
Atributo	Tipo

id	Int
fecha	Timestamp
lugar	String
actividad	String
transporte	String
equipo_produccion	String

Tabla 19: Descripción de la clase PlanProgramacion

Nombre: PlanProgramacion	
Descripción: Representa la cartelera semanal del canal cultural de la UCI.	
Atributo	Tipo
id	Int
fecha_inicio	Date
fecha_creacion	Date

Tabla 20: Descripción de la clase Programacion

Nombre: Programacion	
Descripción: Representa la fecha y hora en que será expuesto un determinado programa por un determinado canal.	
Atributo	Tipo

id	Int
fecha	Timestamp
canal	Int

Tabla 21: Descripción de la clase Usuario

Nombre: Usuario	
Descripción: Representa un usuario con determinados permisos para acceder a la aplicación.	
Atributo	Tipo
usuario	String
contrasenna	String
id	Int

Tabla 22: Descripción de la clase Rol

Nombre: Rol	
Descripción: Representa el papel que juega un usuario dentro de la aplicación.	
Atributo	Tipo
id	Int
nombre	String

Tabla 23: Descripción de la clase AudioVisual

Nombre: AudioVisual	
Descripción: Representa cualquier archivo de video con que se trabaje en la DTU.	
Atributo	Tipo
duracion	Int

Tabla 24: Descripción de la clase Permiso

Nombre: Permiso	
Descripción: Representa los permisos de acceso que tiene un usuario al entrar a la aplicación.	
Atributo	Tipo
nombre	String
id	Int

Tabla 25: Descripción de la clase AudioVisualUCI

Nombre: AudioVisualUCI	
Descripción: Representa cualquier archivo de video producido en la UCI con que se trabaje en la DTU.	
Atributo	Tipo
fecha_edicion	Date

editor	String
realizador	String
camarografo	String
digitalizador	String
sonidista	String

Tabla 26: Descripción de la clase AudioVisualExterno

Nombre: AudioVisualExterno	
Descripción: Representa cualquier archivo de video producido fuera de la UCI con que se trabaje en la DTU.	
Atributo	Tipo
director	String
observaciones	String
pais	String

Tabla 27: Descripción de la clase NTipoFormatoImagen

Nombre: NTipoFormatoImagen	
Descripción: Nomenclador que representa los tipos de formatos de imagen con que se trabaja en la DTU.	
Atributo	Tipo

id	Int
nombre	String

Tabla 28: Descripción de la clase NEstandarVideo

Nombre: NEstandarVideo	
Descripción: Nomenclador que representa los estándares de video con que se trabaja en la DTU.	
Atributo	Tipo
id	Int
nombre	String

Tabla 29: Descripción de la clase NTipoFormatoMusica

Nombre: NTipoFormatoMusica	
Descripción: Nomenclador que representa los tipos de formatos de música con que se trabaja en la DTU.	
Atributo	Tipo
id	int
nombre	String

Tabla 30: Descripción de la clase NTipoAtributo_Equipamiento

Nombre: NTipoAtributo_Equipamiento	
---	--

Descripción: Nomenclador que representa los valores que puede tomar un atributo de un determinado equipamiento.

Atributo	Tipo
valor	String

Tabla 31: Descripción de la clase NTipoAtributo

Nombre: NTipoAtributo

Descripción: Nomenclador que representa los diferentes atributos que puede tener un determinado equipamiento.

Atributo	Tipo
id	Int
nombre	String

Tabla 32: Descripción de la clase NTipoEquipamiento

Nombre: NTipoEquipamiento

Descripción: Nomenclador que representa los diferentes atributos que puede tener un determinado equipamiento.

Atributo	Tipo
id	Int
nombre	String
clasificación_equipamiento	String

Descripción de la estructura física de la base de datos

Esquemas

Seguridad: Mantiene los datos de los trabajadores de la DTU, sus usuarios en la aplicación si es aplicable y los roles y permisos dentro de la misma.

Figura 3: Esquema Seguridad

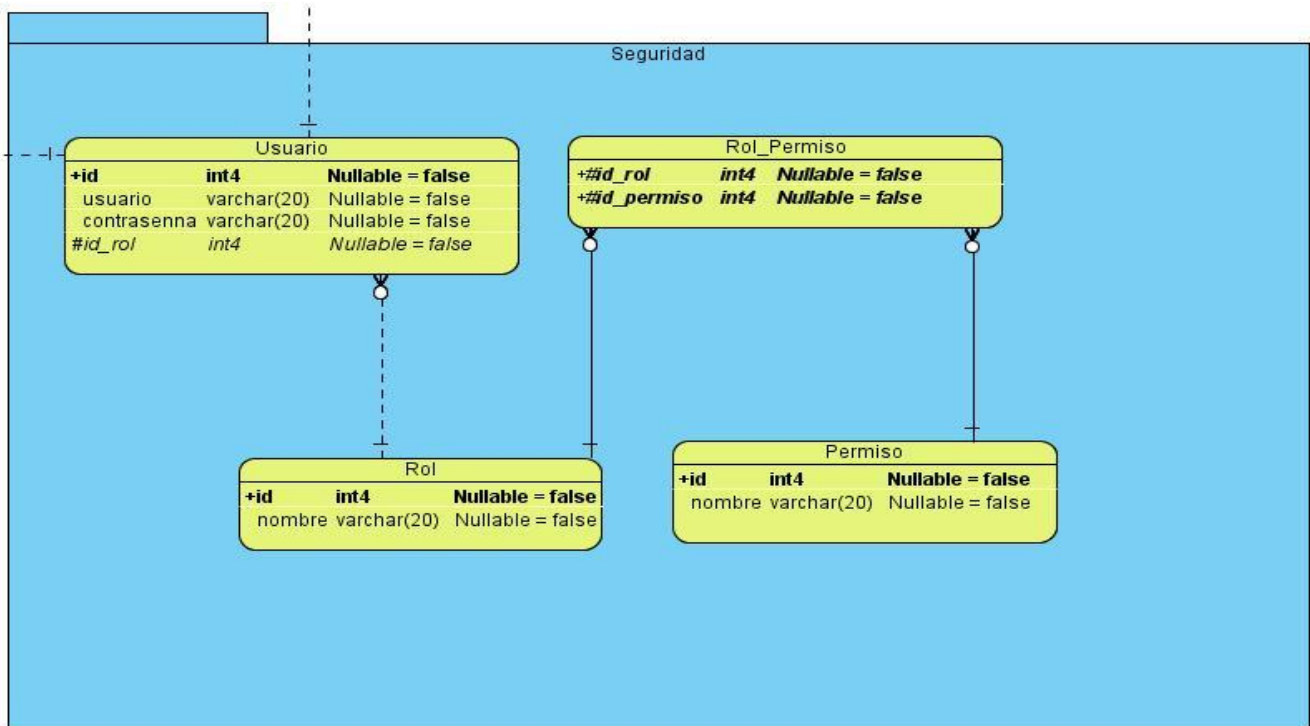


Tabla 33: Descripción de la entidad Usuario

Nombre: Usuario		
Descripción: Representa un usuario con determinados permisos para acceder a la aplicación.		
Atributo	Tipo	Descripción
usuario	Varchar(20)	Usuario con permisos de acceso a la

		aplicación de un trabajador de la DTU.
contrasenna	Varchar(20)	Contraseña asociada a un usuario determinado.
id	Int(10)	Identificador de la entidad.
id_rol	Int(10)	Llave foránea que representa el rol que tiene asociado el usuario.

Tabla 34: Descripción de la entidad Rol

Nombre: Rol		
Descripción: Representa el papel que juega un usuario dentro de la aplicación.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(20)	Nombre del rol.

Tabla 35: Descripción de la entidad Permiso

Nombre: Permiso		
Descripción: Representa los permisos de acceso que tiene un usuario al entrar a la aplicación.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(20)	Nombre del rol.

Tabla 36: Descripción de la entidad Rol_Permisos

Nombre: Rol_Permisos		
Descripción: Representa los permisos que tiene un determinado rol.		
Atributo	Tipo	Descripción
id_rol	Int(10)	Identificador de la entidad rol.
id_permiso	Int(10)	Identificador de la entidad permiso.

Media: Este esquema es el más extenso de todos. Contiene las tablas que representan la jerarquía de todos los elementos audiovisuales manejados por la DTU actualmente, así como los nomencladores con los cuales se relacionan.

Figura 4: Esquema Media

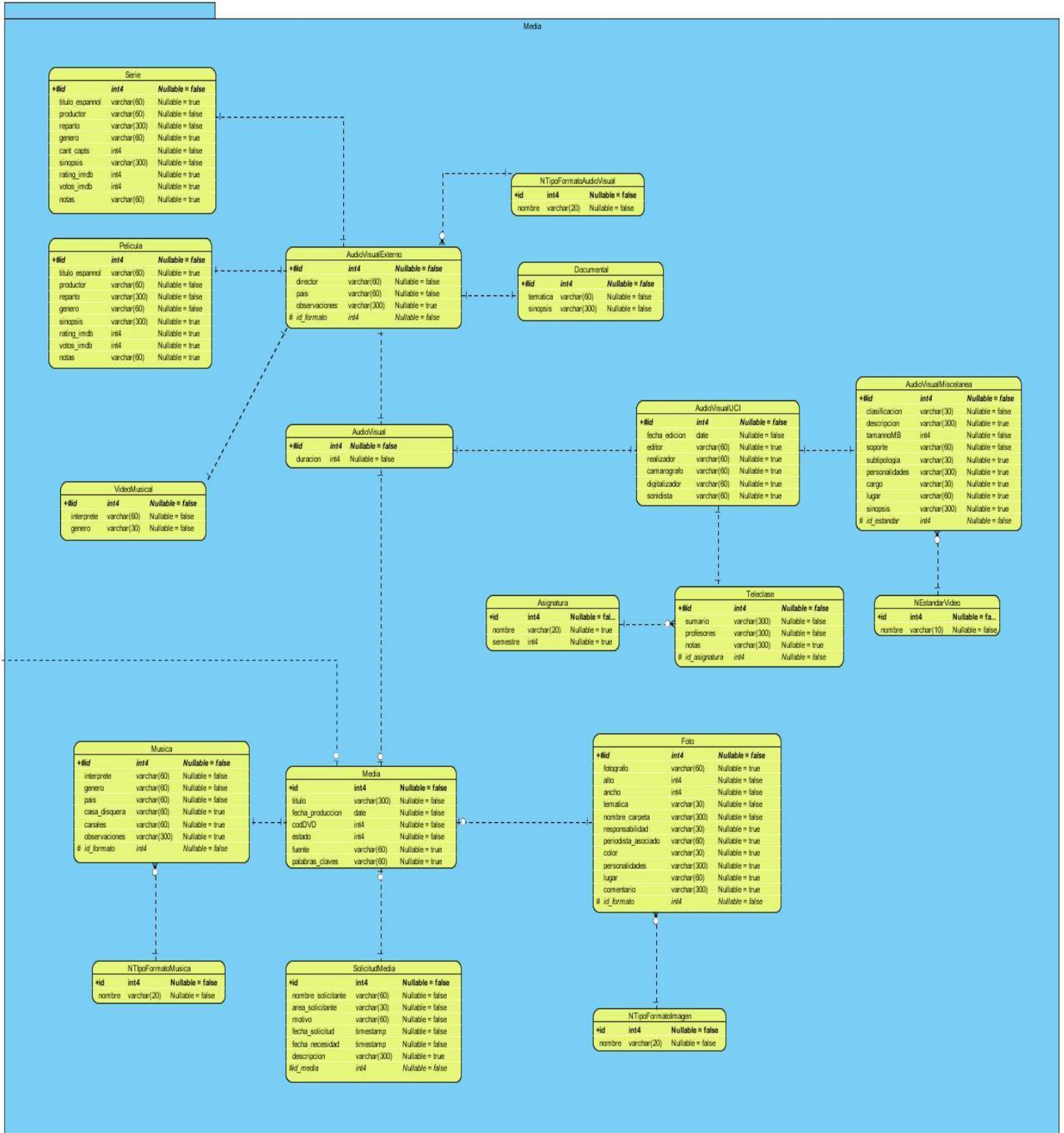


Tabla 37: Descripción de la entidad Serie

Nombre: Serie		
Descripción: Contiene la información a guardar de las series con las que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
cant_capt	Int(10)	Cantidad de capítulos de la serie.
titulo_espanol	Varchar(60)	Título en español de la serie.
productor	Varchar(60)	Nombre y apellidos del productor.
reparto	Varchar(300)	Nombre y apellidos de los miembros del reparto.
genero	Varchar(60)	Genero de la serie.
sinopsis	Varchar(300)	Breve resumen sobre lo que trata la serie.
rating_imdb	Int(10)	
votos_imdb	Int(10)	
notas	Varchar(60)	Observaciones

Tabla 38: Descripción de la entidad Pelicula

Nombre: Pelicula		
Descripción: Contiene la información a guardar de las películas con las que se trabaja en la DTU.		

Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
titulo_espannol	Varchar(60)	Título en español de la película.
productor	Varchar(60)	Nombre y apellidos del productor.
reparto	Varchar(300)	Nombre y apellidos de los miembros del reparto.
genero	Varchar(60)	Genero de la película.
sinopsis	Varchar(300)	Breve resumen sobre lo que trata la película.
rating_imdb	Int(10)	
votos_imdb	Int(10)	
notas	Varchar(60)	Observaciones

Tabla 39: Descripción de la entidad VideoMusical

Nombre: VideoMusical		
Descripción: Contiene la información a guardar de los videos musicales con los que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
interprete	Varchar(60)	Nombre y apellidos del intérprete.
genero	Varchar(30)	Genero musical.

Tabla 40: Descripción de la entidad Documental

Nombre: Documental		
Descripción: Contiene la información a guardar de los documentales con los que se trabaja en la DTU.		
Atributo	Tipo	Descripción
tematica	Varchar(60)	Temática que trata el documental.
sinopsis	Varchar(300)	Breve resumen sobre lo que trata el documental.
id	Int(10)	Identificador de la entidad.

Tabla 41: Descripción de la entidad NTipoFormatoAudioVisual

Nombre: NTipoFormatoAudioVisual		
Descripción: Nomenclador para guardar los tipos de formatos de materiales audiovisuales con los que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(20)	Nombre del formato.

Tabla 42: Descripción de la entidad AudioVisualExterno

Nombre: AudioVisualExterno		
Descripción: Representa cualquier archivo de video producido fuera de la UCI con que se		

trabaje en la DTU.		
Atributo	Tipo	Descripción
director	Varchar(60)	Nombre y apellidos del director.
observaciones	Varchar(300)	Observaciones.
pais	Varchar(60)	País donde se produjo el audiovisual.
id	Int(10)	Identificador de la entidad.
id_formato	Int(10)	Llave foránea que representa el tipo de formato del audiovisual.

Tabla 43: Descripción de la entidad AudioVisual

Nombre: AudioVisual		
Descripción: Representa cualquier archivo de video con que se trabaje en la DTU.		
Atributo	Tipo	Descripción
duracion	Int(10)	Tiempo de duración del audiovisual en minutos.
id	Int(10)	Identificador de la entidad.

Tabla 44: Descripción de la entidad AudioVisualUCI

Nombre: AudioVisualUCI		
Descripción: Representa cualquier archivo de video producido en la UCI con que se trabaje en la DTU.		

Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
fecha_edicion	Date	Fecha de edición del audiovisual.
realizador	Varchar(60)	Representa al trabajador de la DTU que cumplió la función de realizador del audiovisual. Puede ser nulo.
editor	Varchar(60)	Representa al trabajador de la DTU que cumplió la función de editor del audiovisual. Puede ser nulo.
camarografo	Varchar(60)	Representa al trabajador de la DTU que cumplió la función de camarógrafo del audiovisual. Puede ser nulo.
digitalizador	Varchar(60)	Representa al trabajador de la DTU que cumplió la función de digitalizador del audiovisual. Puede ser nulo.
sonidista	Varchar(60)	Representa al trabajador de la DTU que cumplió la función de sonidista del audiovisual. Puede ser nulo.

Tabla 45: Descripción de la entidad Teleclase

Nombre: Teleclase
Descripción: Contiene la información a guardar de las teleclases con las que se trabaja en la DTU.

Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
Id_asignatura	Int(10)	Llave foránea que representa la asignatura a la que pertenece la teleclase.
sumario	Varchar(300)	Resumen de los contenidos que se abordan en la teleclase.
profesores	Varchar(300)	Nombre y apellidos de los profesores que imparten la teleclase.
notas	Varchar(300)	Observaciones.

Tabla 46: Descripción de la entidad Asignatura

Nombre: Asignatura		
Descripción: Contiene la información a guardar de las asignaturas que tienen teleclases.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(20)	Nombre de la asignatura.
semestre	Int(1)	Semestre al que pertenece la asignatura.

Tabla 47: Descripción de la entidad AudioVisualMiscelanea

Nombre: AudiovisualMiscelanea		
Descripción: Representa cualquier material audiovisual que se produce en la UCI que no		

clasifique como teleclase.		
Atributo	Tipo	Descripción
clasificacion	Varchar(30)	Clasificación del audiovisual.
descripcion	Varchar(300)	Descripción de lo que trata el material.
tamannoMB	Int(10)	Tamaño en megabytes del material.
soporte	Varchar(60)	Soporte en el que se grabó.
subtipologia	Varchar(30)	Subtipología del material.
personalidades	Varchar(300)	Nombre y apellidos de las principales personalidades que aparecen en la material.
cargo	Varchar(30)	Cargo de las principales personalidades que aparecen en el material.
lugar	Varchar(60)	Lugar donde se filmó el material.
sinopsis	Varchar(300)	Breve resumen sobre lo que trata el material.
id	Int(10)	Identificador de la entidad.
id_estandar	Int(10)	Llave foránea que representa el estándar de video del audiovisual.

Tabla 48: Descripción de la entidad NEstandarVideo

Nombre: NEstandarVideo
Descripción: Nomenclador que representa los estándares de video con que se trabaja en la DTU.

Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(10)	Nombre del estándar de video.

Tabla 49: Descripción de la entidad NTipoFormatoImagen

Nombre: NTipoFormatoImagen		
Descripción: Nomenclador que representa los tipos de formatos de imagen con los que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre	Varchar(10)	Nombre del tipo de formato de imagen.

Tabla 50: Descripción de la entidad Foto

Nombre: Foto		
Descripción: Contiene la información a guardar de las imágenes con las que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
fotografo	Varchar(60)	Nombre y apellidos del fotógrafo.
nombre_carpeta	Varchar(300)	Dirección de la carpeta en la que está guardada la foto.

responsabilidad	Varchar(30)	
periodista_asociado	Varchar(60)	Periodista asociado al reportaje para el cual se tiró la foto.
alto	Int(10)	Alto de la foto.
ancho	Int(10)	Ancho de la foto.
color	Varchar(30)	Color de la foto.
tematica	Varchar(30)	Temática que trata.
personalidades	Varchar(300)	Principales personalidades que aparecen en la foto.
lugar	Varchar(60)	Lugar donde se tomó la foto.
comentario	Varchar(300)	Observaciones.
id_formato	Int(10)	Llave foránea que representa el tipo de formato de imagen de la foto.

Tabla 51: Descripción de la entidad Media

Nombre: Media		
Descripción: Representa cualquier archivo audiovisual con el que se trabaje en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
codDVD	Int(10)	Código del DVD en que se almacena la media.
titulo	Varchar(300)	Título original de la media.

fecha_produccion	Date	Fecha de producción de la media.
fuelle	Varchar(60)	Fuente de donde se obtuvo la media.
estado	Int(10)	Estado de la media.
palabras_claves	Varchar(60)	Palabras que representan elementos importantes dentro de la media.

Tabla 52: Descripción de la entidad Musica

Nombre: Musica		
Descripción: Contiene la información a guardar de la música con la que se trabaja en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
canales	Varchar(60)	Canales de sonido.
genero	Varchar(60)	Género musical.
pais	Varchar(60)	País del intérprete.
casa_disquera	Varchar(60)	Nombre de la casa disquera.
interprete	Varchar(60)	Nombre y apellidos del intérprete.
observaciones	Varchar(300)	Observaciones.
id_formato	Int(10)	Llave foránea que representa el tipo de formato del archivo.

Tabla 53: Descripción de la entidad SolicitudMedia

Nombre: SolicitudMedia		
Descripción: Representa la solicitud que se le hace al CGIA de una media determinada.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
nombre_solicitante	Varchar(60)	Nombre y apellidos de la persona que solicita la media.
area_solicitante	Varchar(30)	Área de la UCI que solicita la media
motivo	Varchar(60)	Motivo por los que se solicita la media.
fecha_solicitud	Date	Fecha en que se solicita la media.
fecha_necesidad	Date	Fecha para la que se necesita la media.
descripcion	Varchar(300)	Descripción de la media.
id_media	Int(10)	Llave foránea que representa la media solicitada.

Tabla 54: Descripción de la entidad NTipoFormatoMusica

Nombre: NTipoFormatoMusica		
Descripción: Nomenclador que representa los tipos de formatos de música con que se trabaja en la DTU.		
Atributo	Tipo	Descripción

id	Int(10)	Identificador de la entidad.
nombre	Varchar(20)	Nombre del tipo de formato de música.

Planificación: Agrupa las entidades correspondientes a la planificación de la programación y la producción de la DTU.

Figura 5: Esquema Planificación

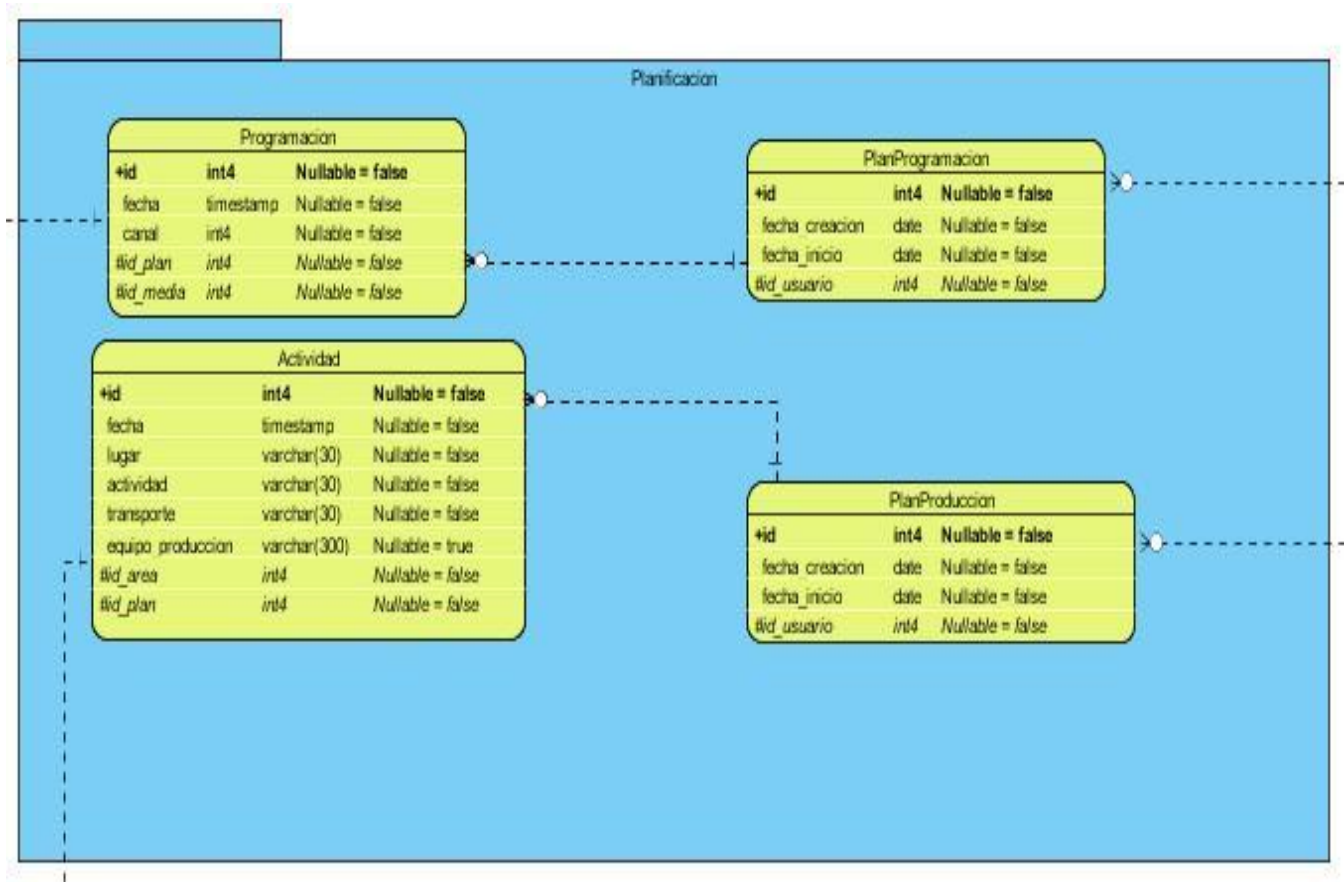


Tabla 55: Descripción de la entidad PlanProgramacion

Nombre: PlanProgramacion		
Descripción: Representa la cartelera semanal del canal cultural de la UCI.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
id_usuario	Int(10)	Llave foránea que representa al usuario que confeccionó el plan de programación.
fecha_inicio	Date	Fecha que marca el inicio de la cartelera.
fecha_creacion	Date	Fecha en que se confeccionó el plan de programación.

Tabla 56: Descripción de la entidad Programacion

Nombre: Programacion		
Descripción: Representa la fecha y hora en que será expuesto un determinado programa por el canal cultural.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
fecha	Timestamp	Fecha y hora en que se será expuesto el programa.
id_plan	Int(10)	Llave foránea que representa el plan al cual pertenece la programación.

id_media	Int(10)	Llave foránea que representa la media que será expuesta.
canal	Int(2)	Número del canal por el que será expuesto el programa.

Tabla 57: Descripción de la entidad Actividad

Nombre: Actividad		
Descripción: Representa las actividades a las que la DTU les dará cobertura como parte del plan de producción.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
fecha	Timestamp	Fecha y hora en que se realizará la actividad a la que la DTU dará cobertura.
lugar	Varchar(30)	Lugar en que se realizará la actividad a la que la DTU dará cobertura.
actividad	Varchar(30)	Descripción de la actividad.
transporte	Varchar(30)	Entidad que asumirá la responsabilidad de la transportación.
Id_area	Varchar(30)	Llave foránea que representa el área a la que se le entrega el material.
id_plan	Int(10)	Llave foránea que representa el plan de producción al cual está vinculada la actividad.

equipo_produccion	Varchar(300)	Nombre y apellidos de las personas que trabajaran en la producción del material.
-------------------	--------------	--

Tabla 58: Descripción de la entidad PlanProduccion

Nombre: PlanProduccion		
Descripción: Representa el cronograma de las actividades a las que se le dará cobertura semanalmente por parte de la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Identificador de la entidad.
id_usuario	Int(10)	Llave foránea que representa al usuario que confeccionó el plan de producción.
fecha_inicio	Date	Fecha que marca el inicio de la semana para la que se planifica la producción.
fecha_creacion	Date	Fecha en que se confeccionó el plan de producción.

Inventario: Este esquema maneja la información relativa a los equipamientos inventariados existentes en la DTU y los movimientos de medios entre áreas.

Figura 6: Esquema Inventario

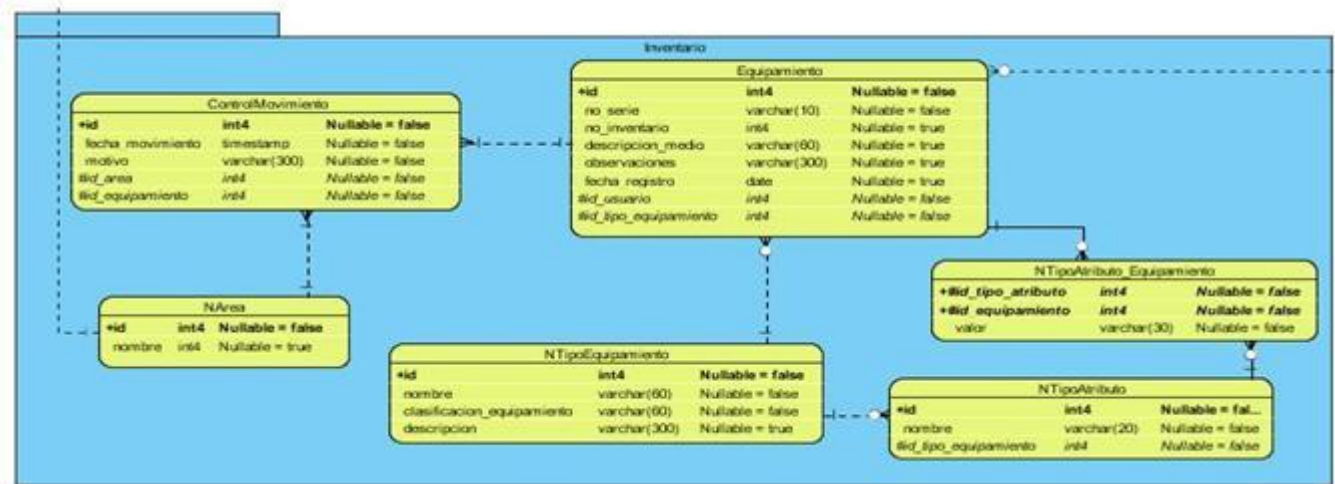


Tabla 59: Descripción de la entidad Equipamiento

Nombre: Equipamiento		
Descripción: Contiene la información sobre los medios existentes en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Llave primaria de la entidad
no_serie	varchar (20)	Número de serie del medio.
no_inventario	Int(10)	Número de inventario.
descripcion_medio	Varchar (60)	Descripción del medio.
observaciones	Varchar (300)	Observaciones
fecha_registro	Date	Fecha en que se registró el medio por primera

		vez.
id_usuario	Int(10)	Llave foránea que representa el usuario al que está asignado un equipamiento determinado.
id_tipo_equipamiento	Int(10)	Llave foránea que representa el tipo de equipamiento.

Tabla 60: Descripción de la entidad ControlMovimiento

Nombre: ControlMovimiento		
Descripción: Representa la reasignación de un medio a un área determinada de la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Llave primaria de la entidad
fecha_movimiento	Timestamp	Fecha en que se reasignó el medio a un área determinada.
motivo	Varchar(300)	Representa el motivo por el cual se hace necesario el movimiento del equipamiento.
id_area	Int(10)	Llave foránea que representa el área a la q se movió el medio.
id_equipamiento	Int(10)	Llave foránea que representa el medio que fue reasignado.

Tabla 61: Descripción de la entidad NArea

Nombre: NArea		
Descripción: Nomenclador que representa las áreas de la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Llave primaria de la entidad.
nombre	Varchar(60)	Nombre del área.

Tabla 62: Descripción de la entidad NTipoAtributo_Equipamiento

Nombre: NTipoAtributo_Equipamiento		
Descripción: Esta tabla constituye la unión de NTipoAtributo y Equipamiento. Representa el valor de un atributo determinado en un equipo dado.		
Atributo	Tipo	Descripción
id_tipo_atributo	Int(10)	Llave primaria de la entidad
id_equipamiento	Int(10)	Llave primaria de la entidad
valor	Varchar(30)	Valor del atributo definido por id_tipo_atributo en el equipo definido por id_equipamiento

Tabla 63: Descripción de la entidad NTipoAtributo

Nombre: NTipoAtributo		
Descripción: Nomenclador que representa el tipo del atributo que posee un determinado equipamiento. Este nomenclador identifica no solo al atributo, sino al tipo de equipo al que pertenece dicho atributo		

Atributo	Tipo	Descripción
id	Int(10)	Llave primaria de la entidad
nombre	Varchar(20)	Nombre del atributo
id_tipo_equipamiento	Int(10)	Llave foránea del tipo de equipamiento al cual pertenece el atributo identificado.

Tabla 64: Descripción de la entidad NTipoEquipamiento

Nombre: NTipoEquipamiento		
Descripción: Nomenclador que representa el tipo de equipamiento disponible en la DTU.		
Atributo	Tipo	Descripción
id	Int(10)	Llave primaria de la entidad
nombre	Varchar(60)	Nombre del tipo de equipamiento
descripcion	Varchar(300)	Descripción del tipo de equipamiento
clasificacion_equipamiento	Varchar(60)	Clasificación del equipamiento que puede ser una de dos, equipo informático o equipo especializado de televisión.

Conclusiones

En este capítulo se ha presentado el diseño de la base de datos para dar solución a la situación problemática. Dicha solución presenta una arquitectura tradicional de bases de datos relacionales para aplicaciones de gestión. Dentro de la misma la información estará organizada en esquemas de acuerdo a sus funciones, de manera que sea sencillo evolucionar cada requisito lo mas separado posible. Se

presentan los artefactos de diseño producto de la aplicación de la técnica de diseño elegida, y la descripción de cada uno de ellos en cada fase del trabajo. Se aprecia claramente la secuencia de trabajo ejecutada desde el modelo conceptual hasta el modelo físico de la base de datos. Como producto de este proceso se ha presentado una base de datos funcional, que contiene todas las entidades significativas del negocio y del sistema y que se encuentra normalizada y lista para ser utilizada por la aplicación.

Capítulo 4: Validación del diseño realizado

Introducción

En el siguiente capítulo se valida, mediante el suministro de evidencia objetiva, que se cumplen los requisitos para la utilización de la base de datos diseñada atendiendo a aspectos teóricos, como la integridad, la normalización, el análisis de la redundancia y la seguridad de la base de datos; y a aspectos funcionales como el llenado de las tablas y las pruebas de carga.

Validación teórica del diseño

Integridad

La exigencia de integridad garantiza la calidad de la información en la base de datos. Dos pasos importantes en el diseño de las tablas son la identificación de valores válidos para una columna y la determinación de cómo forzar la integridad de los datos en la columna. La integridad de datos en el SGPDTU se contempla dividida en 4 categorías:

- **Integridad de entidad:** La integridad de entidad exige que la clave de la entidad sea única y que existan restricciones de tipo PRIMARY KEY o UNIQUE sobre estos campos.
- **Integridad de dominio:** La integridad de dominio es la que se encarga de restringir el conjunto de valores aceptables de un campo determinado. Esto puede lograrse con restricción NOT NULL, restricciones de tipo de datos y restricciones CHECK (que limitan los valores aceptables de un tipo de dato cualquiera), además de cualquier otra forma aplicable según el gestor para limitar los valores que puedan existir en una columna.
- **Integridad referencial:** Contempla la integridad de las relaciones entre las entidades de la base de datos, específicamente cuando se crean, modifican o destruyen filas completas. Esta es la categoría más importante dado que determina la consistencia de los datos existentes en cada tabla. Esto se logra a través de las restricciones de FOREIGN KEY y los modos de propagación de cambios.
- **Integridad definida por el usuario:** En algunas aplicaciones, las reglas de negocio determinan la manera en la cual las entidades del negocio se relacionan entre si y los valores que pueden llegar a poseer. Estas reglas son usualmente escritas por el usuario en la etapa de levantamiento de

requisitos y son manejadas mayormente por la aplicación, pero en casos muy particulares pueden llegar a afectar el diseño de clases persistentes. De ser necesario, estas restricciones pueden reforzarse mediante el uso de desencadenadores o TRIGGERS.

Dentro de cada una de estas categorías se tomaron ciertas acciones que se detallan a continuación.

Integridad de entidad

Los id numéricos correspondientes a las entidades persistentes del sistema son autoincrementales: Esto asegura que nunca existan ids repetidos en la base de datos ni en el sistema. Este efecto se logra utilizando una secuencia para cada entidad, nombrada de la forma <nombre de tabla>_id_seq, por ejemplo: permiso_id_seq para la tabla permiso. Adicionalmente, están protegidos por una restricción de PRIMARY KEY.

Integridad de dominio

Para lograr la integridad de cada dominio se utilizaron los tipos de datos predefinidos por el gestor de PostgreSQL, específicamente INTEGER, VARCHAR, DATE y TIMESTAMP. También se chequean los valores que no pueden ser nulos utilizando la restricción NOT NULL adecuada sobre la columna.

Integridad referencial

Las relaciones entre las entidades tienen distintos modos de propagación. Se pueden dividir en 3 modos diferentes según el tipo de entidad involucrado en la relación:

- **De nomenclador a entidad:** Se asume que los nomencladores son datos inmóviles de la estructura del negocio. Por ejemplo, si se toma el nomenclador NArea, esta no puede ser eliminada sin que todos los trabajadores, equipamientos, actividades y demás relacionadas con ellas sean afectadas. Por tanto, se usa en estas relaciones el modo de propagación DELETE CASCADE y UPDATE CASCADE, que asegura que todas las modificaciones y eliminaciones en un nomenclador sean apropiadamente transmitidas a las entidades nombradas.
- **Dentro de una jerarquía de generalización-especialización:** Dentro de las tablas de la jerarquía se usan igualmente los modos DELETE CASCADE y UPDATE CASCADE. Esto se hace debido a

que un conjunto de estas tablas representan una entidad del negocio y los cambios en una tabla deben ser reflejados al mismo tiempo en las demás.

- **Entre entidades:** El modo a utilizar depende de la relación que tengan las entidades:
 - Si la relación es débil, como por ejemplo, Usuario y Equipamiento, entonces es posible eliminar un Usuario sin que se elimine el Equipamiento que tenía asignado utilizando los modos DELETE SET NULL y UPDATE CASCADE.
 - Si la relación es de dependencia fuerte, como la relación entre Programación y PlanProgramación, entonces cuando se elimina el lado fuerte de la relación se deben eliminar las entidades débiles (modos DELETE CASCADE y UPDATE CASCADE).
 - Finalmente, si la relación es fuerte pero no es identificativa, como es el caso de Usuario y Rol, entonces se deben restringir los cambios de forma que no se pierda la consistencia. Por ejemplo, si un Usuario tiene un id de Rol determinado, dicho Rol no puede ser eliminado hasta que todos los usuarios que lo referencian cambien de rol o sean eliminados a su vez. Esto se logra con DELETE RESTRICT y UPDATE CASCADE.

Integridad definida por el usuario

No se tiene en cuenta mayormente la integridad definida por el usuario para el negocio debido a que estas reglas pueden ser implementadas con más facilidad en la aplicación. El negocio modelado no define ninguna regla particular para las entidades persistentes que pueda ser modelada fácilmente en la base de datos, sino que se concentra más bien en reglas de negocio enfocadas al funcionamiento de los flujos internos de la organización.

Sin embargo, para el sistema se encontró útil el guardar las contraseñas de los usuarios utilizando un hash MD5 que no evita que la contraseña pueda ser usada para la autenticación de los usuarios por parte del sistema, mientras que sí evita que pueda ser leída en texto plano por un atacante cualquiera.

Normalización

Durante la primera iteración de la creación del modelo relacional de la base de datos se obtuvo el diagrama que se muestra a continuación partiendo de los datos que al cliente le interesaba guardar de cada entidad.

Figura 7: Modelo entidad-relación sin normalizar

A este modelo se le aplicaron las reglas de normalización buscando obtener las ventajas que proveen las mismas y que consisten fundamentalmente en lograr un diseño fácil de entender, que minimice los problemas de lógica y que lleve menos espacio en disco duro por la reducción de datos repetidos. Para ello se siguieron fundamentalmente los siguientes pasos:

- Eliminación de atributos multievaluados y calculables. Ejemplo: atributo fecha_fin en las tablas PlanProgramación y PlanProduccion.
- Eliminación de dependencias funcionales parciales mediante la adición de identificadores. Ejemplo: atributo id en las tablas Programación y Actividad.
- Creación de nuevas entidades a partir de atributos dependientes de forma transitiva de la clave primaria. Ejemplo: tabla Asignatura obtenida a partir de los atributos asignatura y semestre ubicados inicialmente en la tabla Teleclase.

Con el procedimiento descrito anteriormente se logró normalizar todas las relaciones solamente hasta la 3ra Forma Normal ya que se considera que no es necesario para la resolución del problema la aplicación de las otras formas normales (Boyce-Codd, 4ta y 5ta Forma Normal) debido a que la base de datos que se implementó no es significativamente grande y describe procesos bastante simples, llevar la normalización más allá de la 3ra Forma Normal solo provocaría dificultades a la hora de acceder a la información debido a que habría que consultar varias tablas, la mayoría de las mismas con tan solo 1 o 2 columnas; aumentaría el tiempo de acceso a los datos por tanto se obtendría una base de datos ineficiente con un esquema demasiado complejo para trabajar.

Ya que todas las relaciones fueron normalizadas hasta la tercera forma normal se concluye que la base de datos quedó normalizada también, obteniéndose finalmente el modelo que se muestra a continuación.

Figura 8: Modelo entidad-relación después de la normalización



Análisis de redundancia de información

La redundancia de datos es un concepto que se refiere a la repetición innecesaria o excesiva de grandes volúmenes de datos en varias tablas. La redundancia de datos es peligrosa debido fundamentalmente a dos factores: favorece el crecimiento desmesurado de los datos almacenados, y dificulta proporcionalmente el mantenimiento de la consistencia de los mismos. Por estas razones la eliminación de información redundante es un paso importante en el diseño de base de datos.

Después de la normalización de todas las relaciones a 3ra Forma Normal, las redundancias en la información fueron erradicadas en su totalidad. Con la introducción de los identificadores numéricos para las entidades persistentes se pudo evitar que los datos que se propagaban como llaves a las demás tablas contribuyeran a aumentar el tamaño de las mismas y a repetir un gran volumen de valores. La eliminación de los valores calculables y la abstracción de información en nomencladores disminuyeron aun más el tamaño de las entidades necesarias para el negocio del SGPDTU.

Análisis de la seguridad de la base de datos

La seguridad del sistema de datos se compone de un pequeño conjunto de acciones, cada una con un propósito específico:

Encriptar contraseñas: Las contraseñas de la aplicación se encriptan utilizando un algoritmo MD5 de resumen. Con esto se garantiza que las contraseñas no puedan ser conocidas ni siquiera por el administrador del sitio. La encriptación de la contraseña se realiza a través de un disparador programado para ejecutarse sobre la inserción o modificación de un usuario en la aplicación.

Creación de usuarios con permisos separados: Para un funcionamiento más seguro del sistema de datos, se decidió crear dos usuarios separados, uno **root**, con permisos totales, para trabajos de administración, con permiso de acceso solo desde el servidor local; y uno **application**, para el trabajo de la aplicación, sin permisos de DDL y con acceso remoto al servidor de datos. De esta forma la aplicación no puede modificar el esquema de los datos aun si su seguridad es comprometida, aunque no evita la pérdida de la información.

Política de respaldo: La política de respaldo consiste simplemente en mantener una copia de seguridad de los datos que se actualice periódicamente. Esto se logra a través de un pequeño script programado en

el servidor que periódicamente extrae el estado actual de la base de datos en una hora determinada de bajo tráfico. Estos respaldos se guardan en una carpeta determinada siguiendo la siguiente convención de nombre: <día><mes><año>_sgpdtu.

Siguiendo estas políticas no se refuerza la seguridad de los datos relativos al dominio, pero sí con respecto al sistema. Los datos no pueden garantizarse con respecto a las reglas del negocio, pero sí se asegura que las contraseñas sean seguras y solo conocidas para el usuario, que los datos solo sean accesibles desde la aplicación sin que sea posible cambiar el esquema de los datos sino es desde el servidor local. Las políticas de respaldo garantizan que la pérdida de datos a causa de un evento imprevisto sea la mínima posible.

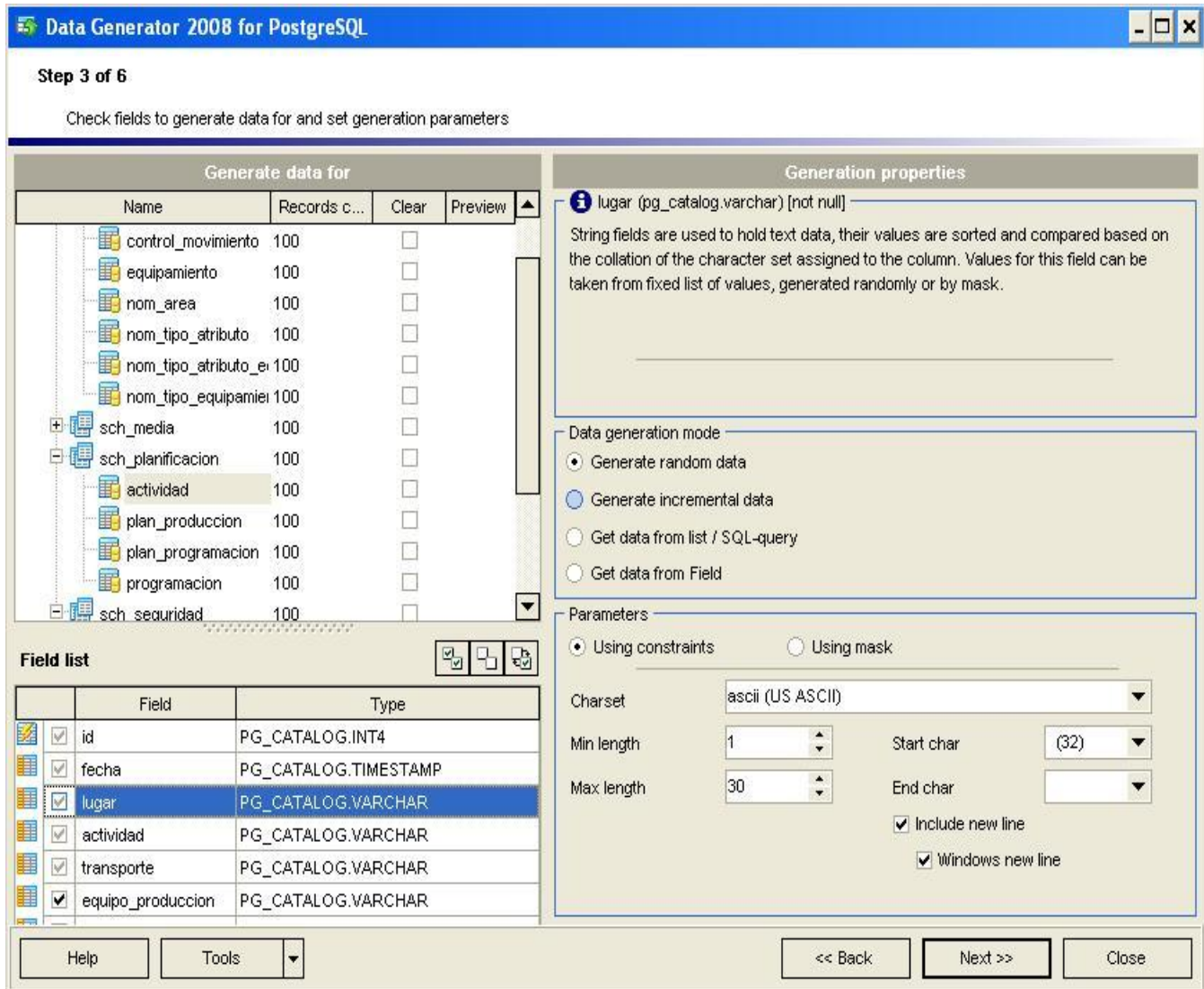
Validación funcional

Uso de herramientas para un llenado voluminoso e inteligente de la base de datos

Para hacer un llenado de pruebas del sistema de datos se eligió como herramienta el EMS Data Generator 2008 para PostgreSQL. Esta herramienta se encarga de proveer facilidades para llenar la base de datos con información que permita realizar pruebas sobre ella sin comprometer la información del negocio de la institución cliente.

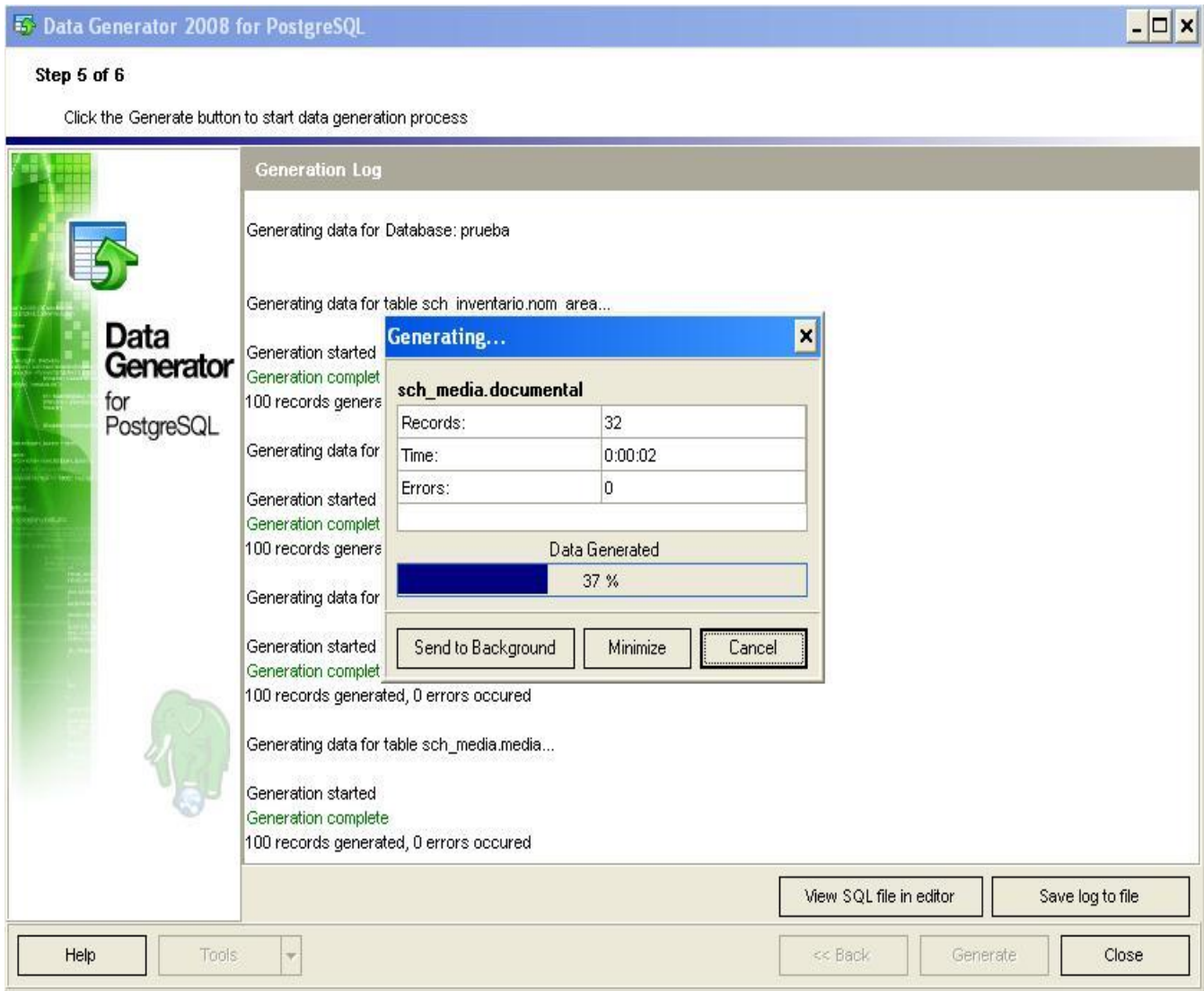
Esta herramienta permite el llenado de la base de datos respetando simultáneamente las restricciones de integridad referencial que hayan sido definidas sobre las tablas y la propiedad auto incremental de los identificadores. Permite el llenado en todas las tablas al mismo tiempo de forma visual y sencilla, dándole al usuario además la posibilidad de elegir el rango de datos que quiere especificar según el tipo de la columna. Esto es especialmente útil si se quiere que la información que se va a usar para las pruebas tenga un parecido estructural con la información real.

Figura 9: EMS Data Generator 2008 para PostgreSQL. Elección de rangos de datos



Se eligió esta herramienta propietaria porque no existen similares libres de fácil acceso que permitan tanta flexibilidad en su uso. Las alternativas libres no están difundidas suficientemente en la Red, y las alternativas propietarias no presentan el mismo grado de libertad de uso o presentan algunas limitaciones para el llenado íntegro de la base de datos o el volumen de datos que pueden generarse de una vez.

Figura 10: EMS Data Generator 2008 para PostgreSQL. Generación de datos.



Utilizando la herramienta se realizó un llenado de la base de datos con 200 filas en cada tabla, conteniendo información aleatoria que luego se usó en las pruebas de carga y concurrencia. Nótese que la información generada por la herramienta no tiene ningún valor desde el punto de vista del usuario por su propio carácter aleatorio.

Búsqueda o diseño de herramientas para pruebas de carga intensiva

Para las pruebas de carga intensiva se eligió JMeter, porque es la herramienta más difundida en la Universidad en su tipo. Actualmente se utiliza tanto en el proyecto CICPC como en el Centro de Aseguramiento de Calidad de Software (Calisoft) de la UCI.

Esta herramienta está implementada en Java por parte de la Apache Software Foundation, y como tal cumple con un número de cualidades deseables: es software libre, multiplataforma y multifuncional. Existe mucha bibliografía para el trabajo con la herramienta y cuenta, a diferencia de otras aplicaciones de software libre, con una ayuda de usuario bien estructurada y explicativa.

Utilizando JMeter se pueden realizar pruebas a base de datos, sitios web, servicios web, y otros. Aunque primariamente el JMeter fue diseñado para pruebas de aplicaciones web, puede usarse para realizar pruebas de carga y concurrencia a bases de datos utilizando JDBC.

Usando JMeter puede configurarse un Plan de Pruebas donde se define la cantidad de usuarios a simular y la cantidad de veces que cada uno de ellos ejecutará la prueba. La prueba en sí se estructura jerárquicamente utilizando elementos de configuración propios de la herramienta. Para una prueba de carga la configuración es relativamente muy sencilla, donde la utilidad de la misma no se ve en su complejidad sino en la forma de recoger los resultados y compararlos.

El marco de pruebas elegido fue el siguiente:

Tabla 65: Marco de prueba. Hardware

Hardware	
Servidor de Base de datos	PC, 1GB DDR2 RAM, 2.20 GHZ CPU
Máquina de pruebas	
Ambas estaciones se conectan a través de una red de Ethernet de 100MB	

Tabla 66: Marco de prueba. Software

Software	
Servidor de Base de datos	Servidor PostgreSQL 8.2
Máquina de pruebas	EMS SQL Manager Pro para PostgreSQL EMS Data Generator 2008 para PostgreSQL JMeter 2.3.2
Ambas estaciones corren Windows XP SP2	

Las pruebas diseñadas utilizando JMeter fueron:

- 15 usuarios, cada uno con 5 ejecuciones de la prueba: La **prueba # 1** fue diseñada con el propósito de probar los tiempos de respuesta teóricos de la base de datos bajo condiciones de poca carga.
- 50 usuarios, cada uno con 5 ejecuciones de la prueba: La **prueba # 2** fue diseñada para probar los tiempos de respuesta esperados de la base de datos en condiciones reales de concurrencia. La cantidad de usuarios esperados en un momento dado se definió como la mitad de los usuarios máximos posibles de la aplicación.
- 100 usuarios, cada uno con 5 ejecuciones de la prueba: La **prueba # 3** fue diseñada para probar la robustez de la base de datos en condiciones de máxima concurrencia esperada. En esta prueba los tiempos de respuesta no son la principal salida, sino la existencia o no de fallos en el gestor cuando se ejecuta la misma.

La configuración de la prueba consistió de 5 consultas SQL a diferentes tablas, generadas utilizando el Query Builder del EMS Manager. Estas tablas fueron previamente pobladas usando el EMS Data Generator:

- **Request Programacion**

```
select * from sch_planificacion.programacion;
```

- **Request Peliculas**

```
SELECT
```

```
sch_media.audiovisual_externo.director,  
sch_media.audiovisual_externo.pais,  
sch_media.audiovisual_externo.observaciones,  
sch_media.audiovisual_externo.id_formato,  
sch_media.media.id,  
sch_media.media.titulo,  
sch_media.media.fecha_produccion,  
sch_media.media.coddvd,  
sch_media.media.estado,  
sch_media.media.fuente,  
sch_media.media.palabras_claves,  
sch_media.pelicula.titulo_espanol,  
sch_media.pelicula.productor,  
sch_media.pelicula.reparto,  
sch_media.pelicula.genero,  
sch_media.pelicula.sinopsis,  
sch_media.pelicula.rating_imdb,
```

sch_media.pelicula.votos_imdb,

sch_media.pelicula.notas,

sch_media.audiovisual.duracion

FROM

sch_media.audiovisual_externo

INNER JOIN sch_media.audiovisual ON
(sch_media.audiovisual_externo.id=sch_media.audiovisual.id)

INNER JOIN sch_media.media ON (sch_media.audiovisual.id=sch_media.media.id)

INNER JOIN sch_media.pelicula ON (sch_media.audiovisual_externo.id=sch_media.pelicula.id)

- **Request Usuarios_Equipamiento**

SELECT

sch_inventario.equipamiento.id,

sch_inventario.equipamiento.no_serie,

sch_inventario.equipamiento.no_inventario,

sch_inventario.equipamiento.descripcion_medio,

sch_inventario.equipamiento.observaciones,

sch_inventario.equipamiento.fecha_registro,

sch_inventario.equipamiento.id_usuario,

sch_inventario.equipamiento.id_tipo_equipamiento,

sch_seguridad.usuario.usuario,


```

sch_seguridad.usuario.contrasenna,
sch_inventario.nom_tipo equipamiento.nombre,
sch_inventario.nom_tipo equipamiento.clasificacion equipamiento,
sch_inventario.nom_tipo equipamiento.descripcion,
sch_inventario.nom_tipo atributo.nombre,
sch_inventario.nom_tipo atributo equipamiento.valor

```

FROM

```
sch_inventario.equipamiento
```

```
INNER JOIN sch_seguridad.usuario ON
```

```
(sch_inventario.equipamiento.id_usuario=sch_seguridad.usuario.id)
```

```
INNER JOIN sch_inventario.nom_tipo equipamiento ON
```

```
(sch_inventario.equipamiento.id_tipo equipamiento=sch_inventario.nom_tipo equipamiento.id)
```

```
INNER JOIN sch_inventario.nom_tipo atributo ON
```

```
(sch_inventario.nom_tipo equipamiento.id=sch_inventario.nom_tipo atributo.id_tipo equipamiento
)
```

```
INNER JOIN sch_inventario.nom_tipo atributo equipamiento ON
```

```
(sch_inventario.nom_tipo atributo equipamiento.id_tipo atributo=sch_inventario.nom_tipo atributo
.id)
```

- **Request Usuario Rol Permiso**

```
SELECT
```

```
sch_seguridad.permiso.nombre,
```

```

sch_seguridad.rol.nombre,
sch_seguridad.usuario.id,
sch_seguridad.usuario.usuario,
sch_seguridad.usuario.contrasenna,
sch_seguridad.rol_permiso.id_rol

```

FROM

```
sch_seguridad.rol
```

```

INNER JOIN sch_seguridad.rol_permiso ON
(sch_seguridad.rol.id=sch_seguridad.rol_permiso.id_rol)

```

```

INNER JOIN sch_seguridad.permiso ON
(sch_seguridad.rol_permiso.id_permiso=sch_seguridad.permiso.id)

```

```

INNER JOIN sch_seguridad.usuario ON
(sch_seguridad.rol_permiso.id_rol=sch_seguridad.usuario.id_rol)

```

- **Request ControlMovimiento_Equipamiento**

SELECT

```

sch_inventario.equipamiento.no_serie,
sch_inventario.equipamiento.no_inventario,
sch_inventario.equipamiento.descripcion_medio,
sch_inventario.equipamiento.observaciones,
sch_inventario.equipamiento.fecha_registro,

```

sch_inventario.control_movimiento.fecha_movimiento,

sch_inventario.control_movimiento.motivo,

sch_inventario.nom_area.nombre

FROM

sch_inventario.equipamiento

INNER JOIN sch_inventario.control_movimiento ON
(sch_inventario.equipamiento.id=sch_inventario.control_movimiento.id equipamiento)

INNER JOIN sch_inventario.nom_area ON
(sch_inventario.control_movimiento.id_area=sch_inventario.nom_area.id)

INNER JOIN sch_inventario.nom_tipo equipamiento ON
(sch_inventario.equipamiento.id_tipo equipamiento=sch_inventario.nom_tipo equipamiento.id)

INNER JOIN sch_inventario.nom_tipo atributo equipamiento ON
(sch_inventario.equipamiento.id=sch_inventario.nom_tipo atributo equipamiento.id equipamiento)

INNER JOIN sch_inventario.nom_tipo atributo ON
(sch_inventario.nom_tipo atributo.id=sch_inventario.nom_tipo atributo equipamiento.id tipo atributo)

AND

(sch_inventario.nom_tipo equipamiento.id=sch_inventario.nom_tipo atributo.id tipo equipamiento)

- **Request Actividad_PlanProduccion**

SELECT * FROM

sch_planificacion.plan_produccion

```
INNER JOIN sch_planificacion.actividad ON
(sch_planificacion.plan_produccion.id=sch_planificacion.actividad.id_plan)
```

Las tablas elegidas para la prueba constituyen el centro de todo el diseño y son por tanto las que se espera que reciban mayor concurrencia de usuarios. Además, la prueba ejercita mediante estas consultas las estructuras principales del diseño, dígase la jerarquía de herencia de la media, la configuración del módulo de seguridad y la configuración de atributos de equipamiento.

Para el análisis de los resultados de las pruebas se definen dos parámetros de especial importancia:

- **Average o promedio:** Especifica el tiempo promedio que se demora la base de datos en responder a una petición del usuario.
- **90% Line o Línea de 90%:** La Línea de 90% representa el mayor tiempo utilizado por el 90% más rápido de las peticiones. O sea, representa el tiempo máximo que puede esperarse que se demore una consulta el 90% de las veces. Esta medida es especialmente útil porque amortiza el error estadístico en las estimaciones.

Los resultados y conclusiones de cada prueba se exhiben a continuación:

Prueba #1

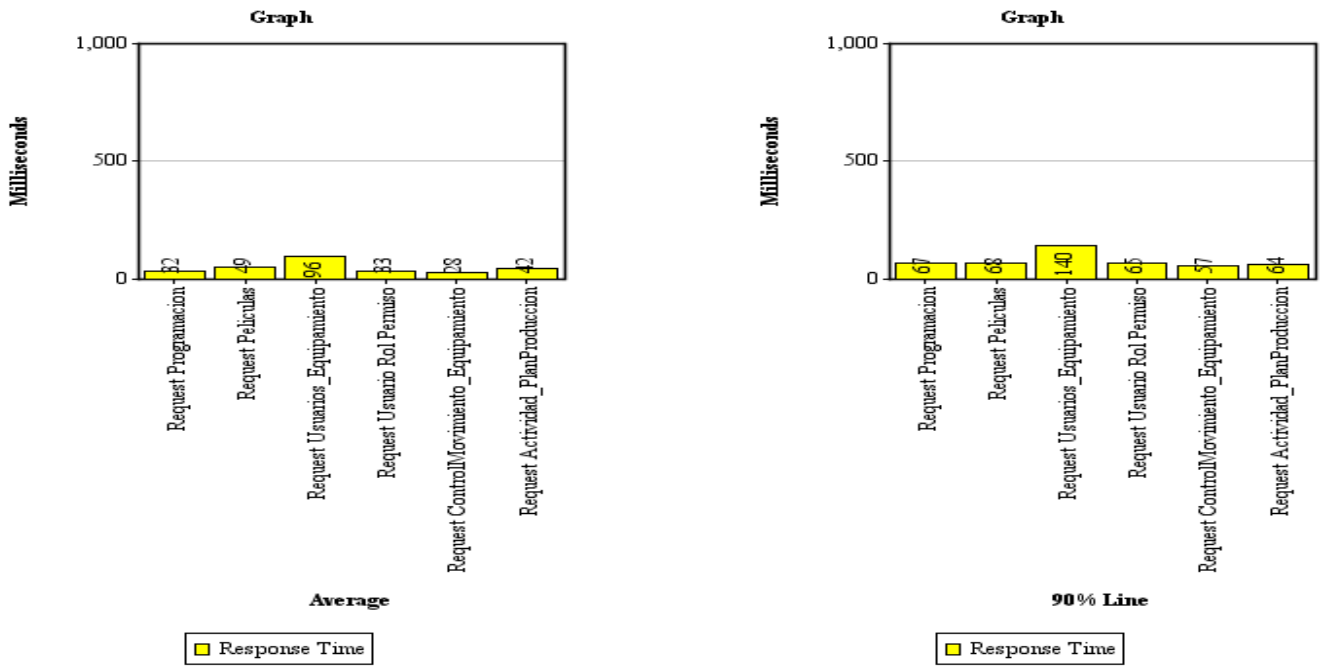
Figura 11: Prueba #1. Resultados

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Request Programacion	75	32	26	67	5	128	0.00%	34.2/sec	254.8
Request Peliculas	75	49	51	68	14	96	0.00%	34.1/sec	1708.4
Request Usuarios_Equipamiento	75	96	93	140	28	200	0.00%	33.8/sec	3658.0
Request Usuario Rol Permiso	75	33	28	65	9	75	0.00%	34.1/sec	724.5
Request ControlMovimiento_Equipamiento	75	28	23	57	7	115	0.00%	34.2/sec	33.5
Request Actividad_PlanProduccion	75	42	44	64	11	104	0.00%	34.5/sec	1080.0
TOTAL	450	47	39	96	5	200	0.00%	198.4/sec	7250.6

En los resultados se advierten la cantidad total de peticiones y los tiempos de cada una de las consultas. Se llega a la conclusión de que la consulta más lenta en condiciones de baja concurrencia es **Request Usuarios_Equipamiento**.

Observando más de cerca los datos de Promedio de tiempo de respuesta y Línea de 90% se confirma el resultado anterior. Nótese que los rendimientos de las demás consultas permanecen bastante cercanos unos de otros, lo cual da a ver que el servidor se comporta establemente frente a distintas consultas.

Figura 12: Prueba #1. Gráficos de Promedio y Línea 90%



Prueba #2

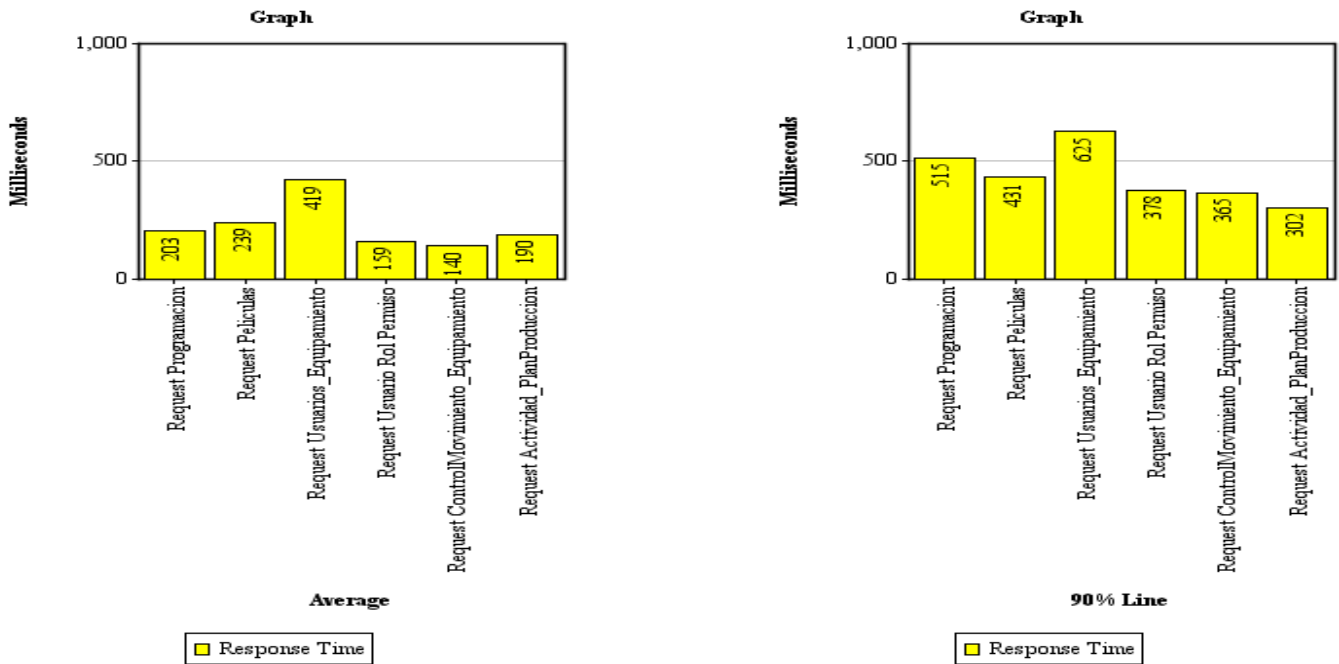
Figura 13: Prueba #2. Resultados

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Request Programacion	250	203	100	515	5	1070	0.00%	32.3/sec	240.2
Request Peliculas	250	239	187	431	13	935	0.00%	32.2/sec	1616.0
Request Usuarios_Equipamiento	250	419	410	625	30	2528	0.00%	32.2/sec	3478.2
Request Usuario Rol Permiso	250	159	103	378	10	926	0.00%	32.6/sec	692.1
Request ControlMovimiento_Equipamiento	250	140	97	365	7	829	0.00%	32.8/sec	32.1
Request Actividad_PlanProduccion	250	190	176	302	10	1069	0.00%	33.2/sec	1039.2
TOTAL	1500	225	170	475	5	2528	0.00%	191.7/sec	7006.8

En los resultados se advierten la cantidad total de peticiones y los tiempos de cada una de las consultas. Se puede afirmar que la consulta más lenta en condiciones de concurrencia esperada es **Request Usuarios_Equipamiento**. Nótese que también un poco separadas del resto destacan **Request Programacion** y **Request Actividad_PlanProduccion** en cuanto a los tiempos máximos de sus peticiones, sin embargo, en promedio destacan más **Request Programacion** y **Request Peliculas**.

Mirando la grafica de la Línea de 90% y comparando con los resultados de la Prueba #1, se ve que existe diferencia relevante entre la segunda petición más lenta y la petición más rápida. Se concluye por tanto que existe una posible degradación en el rendimiento del sistema a mayor concurrencia observada. Para confirmar este resultado se dependerá de la Prueba #3.

Figura 14: Prueba #2. Gráficos de Promedio y Línea 90%



Prueba #3

Figura 15: Prueba #3: Resultados

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Request Programacion	500	654	482	1436	5	2268	0.00%	27.9/sec	207.5
Request Peliculas	500	596	459	1233	17	2198	0.00%	27.6/sec	1393.5
Request Usuarios_Equipamiento	500	689	662	1208	44	2312	0.60%	27.4/sec	2942.3
Request Usuario Rol Permiso	500	539	482	1012	10	2403	0.40%	27.5/sec	582.7
Request ControlMovimiento_Equipamiento	500	450	316	1013	6	2321	0.40%	28.2/sec	27.5
Request Actividad_PlanProduccion	500	487	363	1072	10	2361	0.00%	29.2/sec	913.1
TOTAL	3000	569	464	1223	5	2403	0.23%	163.2/sec	5944.5

En los resultados se advierten la cantidad total de peticiones y los tiempos de cada una de las consultas. Se concluye que en condiciones de concurrencia máxima el gestor alcanza una estabilidad sirviendo los resultados de las consultas, de forma que no hay diferencias significativas en los tiempos de respuesta de ninguna de ellas.

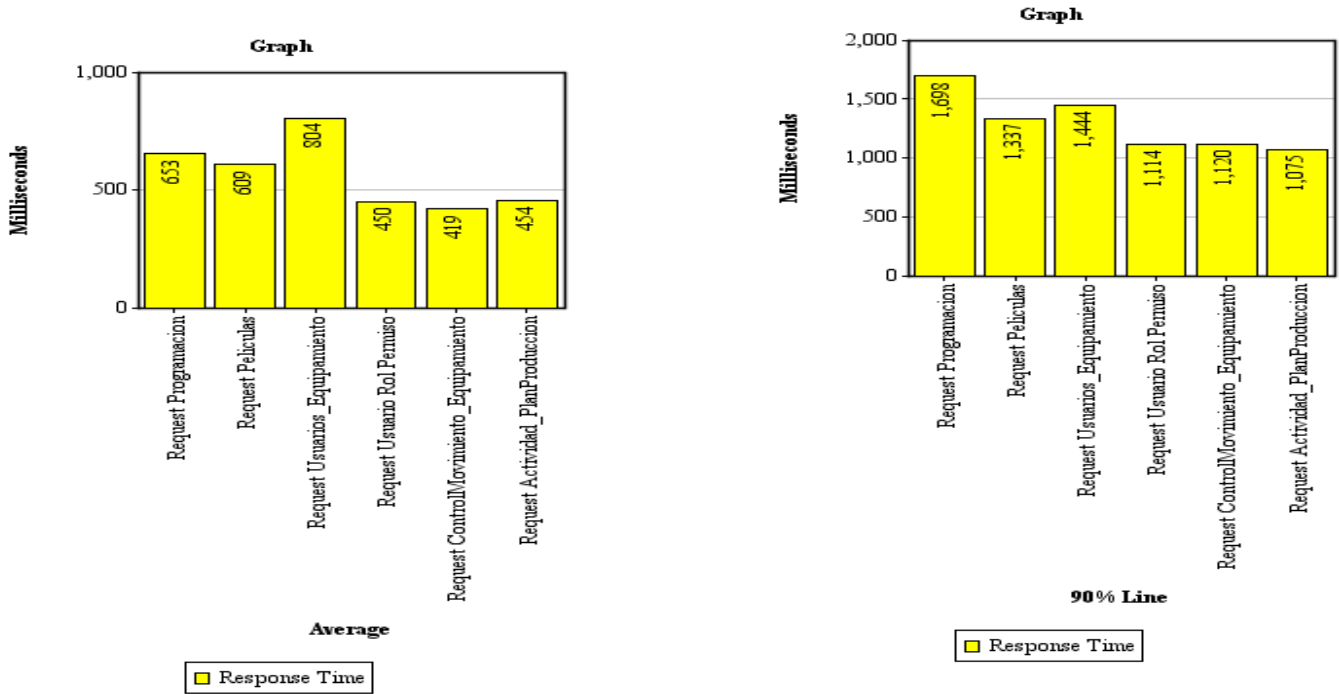
Sin embargo, en esta prueba se incurre en un pequeño porcentaje de errores. Analizando cuidadosamente la causa de estos errores se encontró que están relacionados con la configuración del servidor PostgreSQL, que no acepta suficientes conexiones para dar servicio a tantos usuarios al mismo tiempo. Se afirma entonces que es necesario aumentar la cantidad de conexiones permitidas al gestor a un número ampliamente superior al número máximo de usuarios permitidos, por ejemplo a 200 conexiones.

Resulta significativo que en la gráfica de la Línea de 90% se observa que la **Request Programacion** es más lenta que **Request Usuarios_Equipamiento**. La observación de la gráfica de promedio también muestra que la **Request Programacion** presenta una lentitud pronunciada en sus tiempos de respuesta. Se concluye en este caso que **Request Programacion** no escala tan bien su rendimiento en cuanto a la concurrencia como el resto de las peticiones, dado que su comportamiento y posición relativa entre las demás consultas no se mantiene estable a lo largo de las pruebas.

La diferencia entre la segunda petición más lenta y la petición más rápida es pronunciada, consistente con el resultado de la Prueba #2. Se confirman entonces las conclusiones de dicha prueba de que la base de datos presenta una cierta degradación en su rendimiento a medida que aumenta la concurrencia. No se

considera necesario, sin embargo, tomar acción alguna dirigida a resolver dicha situación dado que los tiempos de respuesta están dentro de los esperados y la degradación del rendimiento se observa solo con grandes cargas concurrentes.

Figura 16: Prueba #3: Gráficos de Promedio y Línea 90%



Conclusiones: Valoración de resultados y propuesta de iteración

Se aseguró apropiadamente la integridad de los datos, tanto referencial como de entidad y demás. La base de datos se presenta por tanto en una condición estable para resistir los cambios y ediciones manuales que puedan ocurrir. Igualmente, se procedió a ejecutar un proceso de normalización y eliminación de redundancias de información dentro de la base de datos, lo cual se llevó a cabo con éxito y determinó la desaparición de información innecesaria y la normalización de todas las tablas a 3ra Forma Normal.

Las políticas de seguridad no son muy estrictas, pero se tomaron las precauciones mínimas necesarias para un sistema de su tipo, incluyendo respaldo y encriptación de contraseñas. Niveles adicionales de seguridad son delegados a la administración de acceso físico del servidor y a la aplicación que interactúe con el sistema gestor de datos.

En cuanto a las pruebas, estas se realizaron a conciencia y arrojaron como resultado que la aplicación puede hacer frente a la necesidad para la cual fue creada, aunque es necesario alterar ligeramente la configuración del servidor para permitir el acceso concurrente máximo definido. Aunque la base de datos presenta cierta pérdida de rendimiento a medida que la concurrencia sube, este comportamiento es esperado y normal y no constituye causa de preocupación.

Se propone como mejora para el diseño la reestructuración de la jerarquía presentada en el esquema media, de forma que las subtipologías manejadas no sean fijas, sino que puedan ser definidas por el usuario en la medida de las necesidades del negocio. Una mejora puede identificarse también en el negocio, debido a concesiones que fue necesario hacer en el diseño para adecuarse a la situación actual de la institución. El caso del equipo de producción es modelo de esta situación, dado que en el negocio actual no está definido un equipo de personas fijo, y por tanto no puede ser representado como una entidad por derecho propio en el diseño, necesitando un campo simple de texto que es propenso a errores. Similarmente el manejo de trabajadores de la DTU puede ser mejorado por parte del negocio para simplificar el diseño del sistema y aumentar su seguridad e integridad.

Conclusiones

Las bases de datos son un importante medio de almacenamiento de información en la actualidad por lo que muchas aplicaciones informáticas utilizan diferentes tipos acorde a sus necesidades. Las más extendidas actualmente son las bases de datos relacionales y justamente este modelo es el que se utilizó para el desarrollo de la aplicación Sistema de Gestión de Procesos de la Dirección de Televisión Universitaria.

El proceso de implementación de la base de datos para este sistema fue debidamente expuesto en este trabajo llegándose a las siguientes conclusiones:

- Se diseñó un modelo de datos acorde a las necesidades del cliente.
- La base de datos está normalizada hasta la 3ra Forma Normal solamente debido a que llevarla más allá implicaría obtener un esquema muy complejo para trabajar, con dificultades a la hora de acceder a la información.
- Se garantizó la integridad y la seguridad de los datos mediante restricciones a diferentes niveles.
- No se realizaron procedimientos almacenados en la base de datos debido a que la aplicación no requiere de consultas complejas y por tanto se aprovecharon las facilidades del framework Symfony, con el cual se implementó el sistema, para el acceso a la información.
- Se demostró con las pruebas de carga que la base de datos mantiene la estabilidad en los tiempos de respuesta de las consultas frente a la conexión simultánea del máximo número posible de usuarios.
- La base de datos diseñada está en condiciones de ser puesta en práctica para darle solución a los problemas de accesibilidad de la información existente en la Dirección de Televisión Universitaria.

Recomendaciones

Debido a la posible utilización de la base de datos modelada en aplicaciones similares se proponen las siguientes mejoras:

- Reestructuración de la jerarquía de generalización- especialización del esquema media de forma que las subtipologías manejadas no sean fijas, sino que puedan ser definidas por el usuario según las necesidades del negocio.
- Modelación de la entidad Equipo de Producción que guardará la información de equipos fijos para la producción de los materiales y la cobertura de las actividades planificadas.
- Modelación de la entidad Trabajadores que guardará la información de todos los trabajadores de la Dirección de Televisión Universitaria aunque los mismos no tengan acceso a la aplicación.
- Modelación de la entidad Inventario donde se lleve el control de los equipos de un inventario determinado, además de la fecha y los datos del usuario que confeccionó el mismo.
- Creación de un documento donde se lleve el control de las dependencias funcionales de la base de datos.
- Modelación de entidades para el control de los eventos de cada usuario que acceda a la base de datos.

Trabajos citados

- **Alfonso Quesada, Belayne. 2008.** Trabajo de Diploma. *Diseño de un Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria*. La Lisa, Ciudad de La Habana, Cuba : s.n., 2 de Julio de 2008.
- **Castro, Esther Cárdenas. 2007.** Portal de Camagüey. *La TV en Cuba: una experiencia camagüeyana*. [En línea] 2007. [Citado el:]
http://www.pprincipio.cult.cu/institucion/ANTENAS/antenas_18_19/secciones/articulos/tv_en_cuba.htm.
- **Díaz Villanueva, Wladimiro. 2002.** Base de datos objeto-relacionales. [En línea] 2002.
http://informatica.uv.es/iiguia/DBD/Practicas/boletin_1.pdf.
- **Espinosa, Humberto. 2005.** Links Global Services. *PostgreSQL: una alternativa de DBMS Open Source*. [En línea] 2005. http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
- **Hansen, James V. y Hansen, Gary W.** *Diseño y Administracion de base de datos*. s.l. : Prentice Hall.
- **Microsoft. 2006.** IslaSoft. *Diferencias entre Microsoft Access y Microsoft SQL Server/MSDE*. [En línea] Septiembre de 2006. http://www.islasoft.com/Producto/Diferencias_SQL_Server_Office.pdf.
- **Pecos, Daniel. 2002.** NetPecos. *PostgreSQL vs. MySQL*. [En línea] 7 de Junio de 2002.
http://www.netpecos.org/docs/mysql_postgres/index.html.
- **Rincón Rincón, Leydi Diana. 2004.** Universidad Nacional de Colombia. *Oracle*. [En línea] 2004.
http://www.fce.unal.edu.co/uifce/newuifce/publicaciones.php?id_publicacion=17
- **Rodriguez, Cesar Flores.** Monografias.com. *Efectos de la televisión*. [En línea]
<http://www.monografias.com/trabajos16/efectos-televison/efectos-televison.shtml>.

- **Veliz Díaz, Osbel. 2008.** Trabajo de Diploma. *Sistema para el Control de los Recursos del Proyecto*. La Lisa, Ciudad de la Habana, Cuba : s.n., Julio de 2008.
- **Wikipedia. 2009.** Wikipedia: La enciclopedia libre. *Base de datos Jerárquica*. [En línea] 12 de Febrero de 2009. http://es.wikipedia.org/wiki/Base_de_datos_jer%C3%A1rquica.
- **Wikipedia, Base de datos. 2009.** Wikipedia: La enciclopedia libre. *Base de datos*. [En línea] 22 de Febrero de 2009. http://es.wikipedia.org/wiki/Base_de_datos.

Bibliografía

- **Alfonso Quesada, Belayne. 2008.** Trabajo de Diploma. *Diseño de un Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria*. La Lisa, Ciudad de La Habana, Cuba : s.n., 2 de Julio de 2008.
- **Castro, Esther Cárdenas. 2007.** Portal de Camagüey. *La TV en Cuba: una experiencia camagüeyana*. [En línea] 2007. [Citado el:] http://www.pprincipo.cult.cu/institucion/ANTENAS/antenas_18_19/secciones/articulos/tv_en_cuba.htm.
- **Díaz Villanueva, Wladimiro. 2002.** Base de datos objeto-relacionales. [En línea] 2002. http://informatica.uv.es/iiquia/DBD/Practicas/boletin_1.pdf.
- **Espinosa, Humberto. 2005.** Links Global Services. *PostgreSQL: una alternativa de DBMS Open Source*. [En línea] 2005. http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.
- **Hansen, James V. y Hansen, Gary W.** *Diseño y Administracion de base de datos*. s.l. : Prentice Hall.
- **Microsoft Corporation. 2009.** MSDN. Centro de desarrollo de SQLServer. *Integridad de los datos*. [En línea] Enero de 2009. <http://msdn.microsoft.com/es-es/library/ms184276.aspx>.
- **Microsoft. 2006.** IslaSoft. *Diferencias entre Microsoft Access y Microsoft SQL Server/MSDE*. [En línea] Septiembre de 2006. http://www.islasoft.com/Producto/Diferencias_SQL_Server_Office.pdf.
- **Pecos, Daniel. 2002.** NetPecos. *PostgreSQL vs. MySQL*. [En línea] 7 de Junio de 2002. http://www.netpecos.org/docs/mysql_postgres/index.html.
- **Rincón Rincón, Leydi Diana. 2004.** Universidad Nacional de Colombia. *Oracle*. [En línea] 2004. http://www.fce.unal.edu.co/uifce/newuifce/publicaciones.php?id_publicacion=17.
- **Rodriguez, Cesar Flores.** Monografias.com. *Efectos de la televisión*. [En línea] <http://www.monografias.com/trabajos16/efectos-television/efectos-television.shtml>.

- **Veliz Díaz, Osbel. 2008.** Trabajo de Diploma. *Sistema para el Control de los Recursos del Proyecto*. La Lisa, Ciudad de la Habana, Cuba : s.n., Julio de 2008.

- **Wikipedia. 2009.** Wikipedia: La enciclopedia libre. *Base de datos Jerárquica*. [En línea] 12 de Febrero de 2009. http://es.wikipedia.org/wiki/Base_de_datos_jer%C3%A1rquica.

- **Wikipedia, Base de datos. 2009.** Wikipedia: La enciclopedia libre. *Base de datos*. [En línea] 22 de Febrero de 2009. http://es.wikipedia.org/wiki/Base_de_datos.

Anexos

Anexo 1: Función encriptar

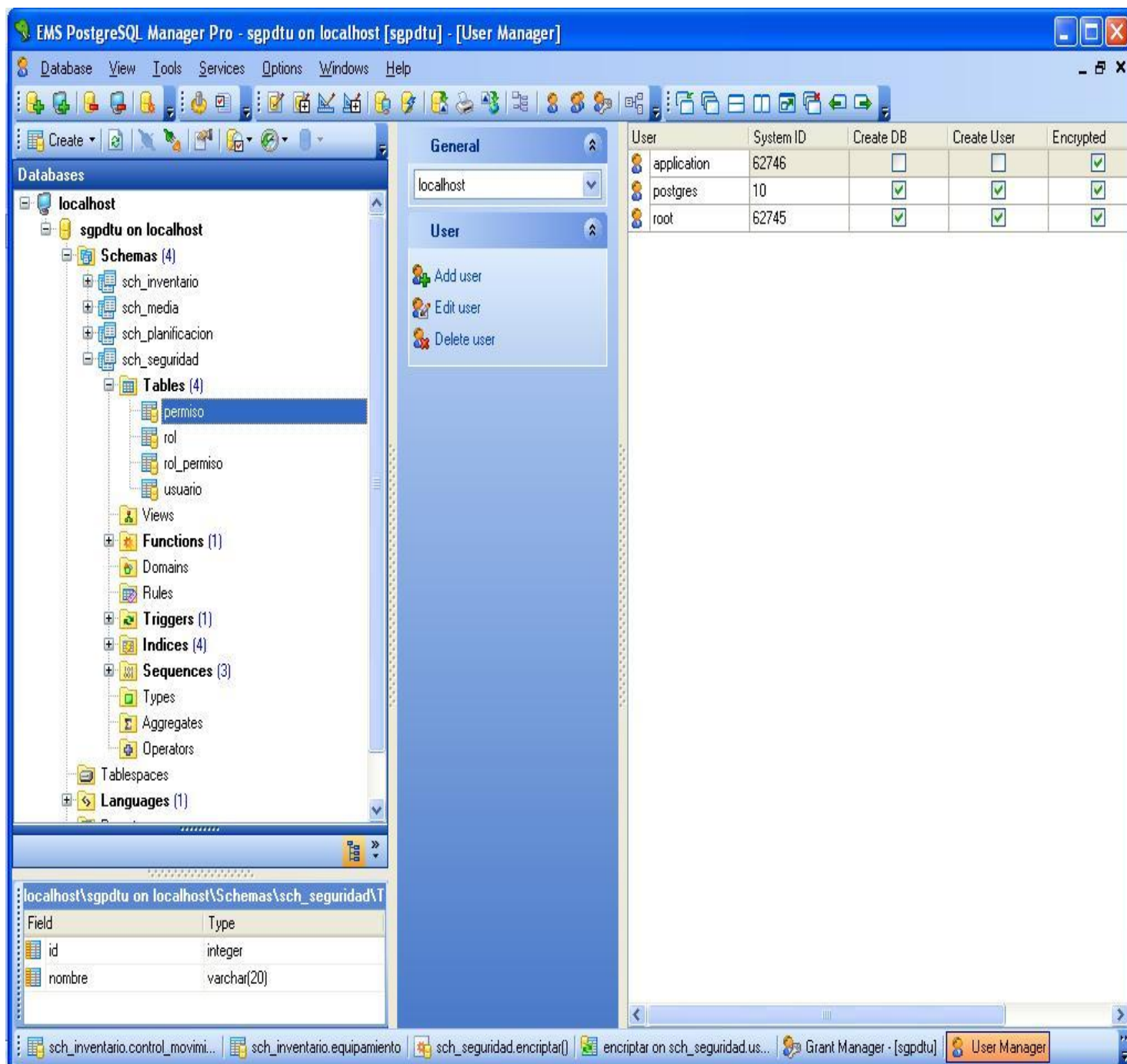
```
CREATE OR REPLACE FUNCTION "sch_seguridad"."encriptar" () RETURNS trigger AS
$body$
BEGIN
    new.contrasenna = md5(new.contrasenna);
    RETURN new;
END;
$body$
LANGUAGE 'plpgsql' IMMUTABLE CALLED ON NULL INPUT SECURITY INVOKER;
```

Anexo 2: Trigger encriptar

```
CREATE TRIGGER "encriptar" BEFORE INSERT
ON "sch_seguridad"."usuario" FOR EACH ROW
EXECUTE PROCEDURE "sch_seguridad"."encriptar"());
```


Anexo 3: Base de datos SGPDTU

Figura 17: Base de datos SGPDTU



The screenshot shows the EMS PostgreSQL Manager Pro interface for the SGPDTU database. The main window is titled "EMS PostgreSQL Manager Pro - sgpdtu on localhost [sgpdtu] - [User Manager]". The interface is divided into several panes:

- Left Pane (Databases):** Shows a tree view of the database structure. Under "localhost", there is a sub-entry "sgpdtu on localhost". Under this, there are "Schemas (4)" (sch_inventario, sch_media, sch_planificacion, sch_seguridad) and "Tables (4)" (permiso, rol, rol_permiso, usuario). Other database objects like Views, Functions, Domains, Rules, Triggers, Indices, Sequences, Types, Aggregates, Operators, Tablespaces, and Languages are also visible.
- Center Pane (User Manager):** Shows the "User" management interface. It includes a "General" tab with a dropdown menu set to "localhost". Below it, there are buttons for "Add user", "Edit user", and "Delete user".
- Right Pane (User List):** Displays a table of users with the following columns: User, System ID, Create DB, Create User, and Encrypted. The data is as follows:

User	System ID	Create DB	Create User	Encrypted
application	62746	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
postgres	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
root	62745	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
- Bottom Pane (Table Structure):** Shows the structure of the "sch_seguridad" table. It has two columns: "id" (integer) and "nombre" (varchar(20)).

The taskbar at the bottom shows several open windows, including "sch_inventario.control_movimi...", "sch_inventario.equipamiento", "sch_seguridad.encryptar()", "encryptar on sch_seguridad.us...", "Grant Manager - [sgpdtu]", and "User Manager".