

Universidad de las Ciencias Informáticas

Facultad 6

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

**TÍTULO: Capa de acceso a datos y modelo físico de la Base de
Datos del Sistema Videoconferencia de la Facultad 6.**

Autor(a)

Yudirenia Torriente Rodríguez.

Tutor(a)

Ing. Ekaterina Alejovna Ramírez Ramírez.

Cotutor(a)

Ing. Solange Rodríguez Vázquez

Mayo de 2011.

Año 53 de la Revolución

Declaración de Autoría

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

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

Yudirenia Torriente Rodríguez
Autor(a)

Ing. Ekaterina Alejovna Ramírez Ramírez.
Tutor(a)

Dedicatoria

*Dedico este trabajo de diploma a todas las personas que de una forma u otra han aportado su granito de arena; brindándome las enseñanzas, el valor y la confianza que necesitaba para formarme como profesional. Especialmente a mi madre, mi hermano, mis tías y mis abuelos por ser mi razón de ser y sin los cuales no hubiera llegado hasta aquí. Especialmente a ti mamá, que siempre confiaste en mí, que en el momento en que ni yo misma me sentía con fuerzas para continuar, tú me abriste los brazos y no permitiste que me rindiera, todo lo contrario, tus palabras: **“Tú lo vas a lograr, te lo mereces”**, fueron más que suficientes para que yo siguiera adelante. Para ti está dedicado este Trabajo de Diploma, te amo con la vida.*

♥Mamá♥.

También a todas aquellas personas que me dieron su apoyo y han estado a mi lado por estos 5 años y

A todos ustedes...

Agradecimientos.

Quiero agradecer a todas las personas que confiaron en mí y me brindaron su apoyo incondicional a lo largo de mi vida y de estos 5 años; principalmente a mis tres madres por la formación y educación que me dieron, sin ellas no hubiera sido posible estar donde estoy ya que ellas han sido mi guía y mi bastoncito que las he tenido delante para levantarme y coger fuerza para seguir adelante.

Principalmente a mi mamá Lourdes por creer en mí, por su apoyo incondicional y su infinito amor, porque este es su sueño hecho realidad pues lo que soy se lo debo a ella y por ser mi razón de existir.

A mi abuela linda Adela que siempre ha estado ahí para escucharme y darme el consejo que necesito, y que también este es su sueño, por ti mamita he cumplido mi promesa de ser ingeniera.

A mis dos tías Lercia y Liliana que desde chiquita me han mimado y apoyado en todo y han formado parte de este sueño, por su forma agradable de hablarme y de darme fuerzas cuando creo no tenerlas.

A mi padrastro Delfín por haber sido como mi padre durante todo este tiempo, por ayudarnos a mí, a mi hermano y a mi mamá, y por poder contar siempre con su apoyo, muchas gracias por existir. También a mi hermano Yunior que lo quiero mucho y le deseo lo mejor del mundo.

A mis compañeras de apartamento a la Flaca, Yalili, Yiyi, Yudalis, Yuliet y principalmente a mi querida amiga Yane que llevamos juntas estos 5 años por haberme soportado todo este tiempo.

A los profesores Jeobanis, Cesar, Pacheco, Daniel principalmente a los que integran mi tribunal a Claritza, Fernando, Carolay, Olga, y mis tutoras Ekaterina y Solangel. Por estar siempre cuando los necesitaba para aclararme dudas y explicarme algo, por haber hecho posible esta investigación.

A estudiantes que me brindaron su ayuda cuando más lo necesité, Elias, Beni, Javier, Yordan y principalmente a alguien muy importante a quien le debo hoy haber podido graduarme a Yanier por tener esa paciencia enorme para soportarme y para explicarme y por estar ahí cuando más necesitaba de su ayuda y sus palabras agradables y confortables.

Resumen

El departamento de Señales Digitales de la Facultad 6 tiene entre sus propósitos la creación de un nuevo Sistema de Videoconferencia propio de la universidad y que el mismo se ajuste a las características de la misma. Además que permita el intercambio de información y el encuentro de varias personas y centros ubicadas en sitios distantes.

Cada uno de estos sistemas posee una BD que les permite lograr un buen control y la seguridad de cada uno de los datos que el mismo contiene internamente, además de parámetros específicos de una videoconferencia que facilitan su funcionamiento.

La propuesta realizada en dicha investigación se basa en el diseño y desarrollo de una BD que garantice la gestión y almacenamiento de la información que se originan a partir de las funcionalidades del sistema. Dicha propuesta tiene datos detallados de la misma, incluyendo tablas, índices, restricciones y procedimientos almacenados específicos de la base de datos que son necesarios para almacenar, recuperar y suprimir objetos persistentes dentro de la misma. Se realizó la implementación de la capa de acceso a datos para lograr una eficaz conexión de la bases de datos.

Dicha implementación fue realizada utilizando herramientas como: Gestor de BD PostgreSQL, Lenguaje de programación C++ y como IDE de desarrollo Qt Creator.

Palabras Claves.

Base de Datos; Capa de acceso a datos; Sistemas de Gestión de BD; Sistema de Videoconferencia

Índice

Introducción	1
Capítulo 1: Fundamentación teórica.....	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema	5
1.3. Objeto de estudio.....	7
1.3.1 Descripción actual del dominio del problema	8
1.3.2 Situación problemática.....	9
1.4 Estudio de los Sistemas de Videoconferencia.....	9
1.4.1 Elementos básicos de un Sistema de Videoconferencia	10
1.4.2 Modalidades de Videoconferencia	11
1.4.3 Estándares de Videoconferencia	11
1.5 Análisis de soluciones existentes.....	12
1.5.1 Observaciones sobre el análisis de las soluciones existentes.....	13
1.6 Características de las arquitecturas de diseño de BD	14
1.7 Tendencias y tecnologías actuales de BD.....	15
1.7.1 Tipos de BD	15
1.8 Sistemas Gestores de BD Relacionales.....	18
1.9 Herramientas CASE.....	21
1.10 Framework de desarrollo a utilizar	23
1.11 Características del ORM para el mapeo de la BD.....	24
1.12 Conclusiones parciales.	27
Capítulo 2: Diseño de la solución propuesta.....	28
2.1 Introducción	28
2.1.2 Modelo de objetos del negocio.....	28
2.2 Diagrama de clases del diseño	28
2.2.1 Clases persistentes	28

2.3	Diseño de la BD.....	29
2.3.1	Modelo Entidad Relación (MER).....	30
2.4	Normalización de la BD	32
2.4.1	Formas Normales	33
2.5	Diagrama Entidad Relación (DER) de la BD	34
2.6	Descripción de las tablas del diagrama entidad relación de la BD	35
2.7	Diagrama de componente.....	39
2.8	Diagramas de componentes para la implementación de la capa de acceso a datos.	40
2.9	Conclusiones parciales.....	42
Capítulo 3: Validación del diseño realizado.....		43
3.1	Introducción	43
3.2	Validación de la teoría del diseño.....	43
3.2.1	Análisis de la integridad de los datos	43
3.2.1.1	Categorías de integridad de los datos:	44
3.4	Análisis de la seguridad de los datos	47
3.5	Validación funcional y tipos de pruebas	48
3.5.1	Herramienta para un llenado voluminoso y eficiente de la BD	48
3.5.2	Herramienta para la administración de BD.....	51
3.7	Conclusiones parciales.....	58
3.8	Conclusiones generales	59
Recomendaciones		60
Referencias bibliográfica.....		61
Bibliografía consultada.....		65
Glosario de términos.....		68

Tablas

Tabla 1: Descripción de la entidad usuario.....	36
Tabla 2: Descripción de la entidad videoconferencia	36
Tabla 3: Descripción de la entidad usuario_videoconferencia.....	36
Tabla 4: Descripción de la entidad registro.....	37
Tabla 5: Descripción de la entidad usuario_lista_contactos	37
Tabla 6: Descripción de la entidad vc_publicada	38
Tabla 7: Descripción de la entidad accion	38
Tabla 8: Descripción de la entidad rol	38
Tabla 9: Descripción de la entidad tipo_videoconferencia	39
Tabla 10: Descripción de la entidad lista_contacto	39
Tabla 11: Descripción de la entidad modo_grabacion.....	39

Ilustraciones

Ilustración 1: Data Mapper	33
Ilustración 2: Identity Field	33
Ilustración 3: Identity Map	34
Ilustración 4: Concrete Table Inheritance	34
Ilustración 5: Singleton.....	35
Ilustración 6: Diagrama de clases persistentes	38
Ilustración 7: Cardinalidad uno a uno	40
Ilustración 8: Cardinalidad uno a muchos.....	40
Ilustración 9: Cardinalidad muchos a muchos	40
Ilustración 10: Modelo Entidad Relación	41
Ilustración 11: Diagrama Entidad Relación.....	44
Ilustración 12: Diagrama de componentes	49
Ilustración 13: Herramienta para la prueba de volumen de la BD.....	55
Ilustración 14: Ejemplo de la realización de la prueba de volumen	56
Ilustración 15: Eficiencia del llenado de la BD.....	57
Ilustración 16: Herramienta de administración de BD.....	58

Capítulo 1: Fundamentación teórica

Introducción

Históricamente se ha tenido la necesidad de guardar información, ya sea por considerarse una simple actividad o por tener en cuenta la importancia que tiene para el desarrollo. La primera forma de almacenamiento de información usada por el hombre fue sin duda la escritura, pero esta no fue muy eficiente ya que se corría el riesgo de perder la información guardada con el tiempo. Para darle solución a esta deficiencia se creó una herramienta conocida como base de datos (BD), con el fin de almacenar gran cantidad de información de forma segura.

Estas son muy importantes para el manejo de la información, la cual es fundamental para cualquier empresa, y también para cualquier persona o entidad. Con una buena organización de la información se puede lograr un alto nivel competitivo dentro del mercado y obtener mayores niveles de capacidad de desarrollo. El manejo y control de la información permite identificar cuáles son las fortalezas, debilidades y sectores vulnerables con los que se cuenta en una organización.

Gracias al gran desarrollo científico-técnico existente en la actualidad este método de almacenamiento de información se ha convertido en la vía más confiable y menos costosa a la cual acude la mayor parte de la población mundial. El mismo, es utilizado como una fuente secundaria en cuanto a la recuperación y acumulación de los datos a nivel científico, social, económico, político y cultural. Además, un sistema de información de BD permite tener un mejor control sobre todos los elementos y recursos con los que cuenta una institución.

Las aplicaciones de almacenamiento de datos más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objetivo de almacenar la información experimental.

El estado revolucionario cubano, reconocido a nivel mundial como baluarte indiscutible en materia de educación, no se encuentra exento del uso de estas tecnologías de almacenamiento de información. Las mismas son utilizadas en centros como: industrias, bancos locales, compañías manufactureras y otras empresas e instituciones para mejorar su funcionamiento interno.

La Universidad de las Ciencias Informáticas (UCI) desde sus inicios hace ocho años, ha sido buena protagonista de los logros informáticos obtenidos en el país. A través de ella comienzan a girar las estrategias de desarrollo de la Industria Cubana del Software. Esta mantiene convenios no solo con empresas cubanas sino también con extranjeras, que proveen a la universidad tecnologías para facilitar el desarrollo de los productos conveniados.

Capítulo 1: Fundamentación teórica

Un ejemplo de estos convenios es el del Sistema de Videoconferencia entre la empresa Huawei y la UCI. Este sistema solo permite la conexión de pocas cámaras y su única comunicación es con tecnologías de la misma empresa. Esto no le es favorable a la universidad debido al costo de la tecnología por lo que se hizo necesario idear la posibilidad de solucionar esta problemática en la misma institución. Esta labor fue otorgada a la Facultad 6, específicamente al departamento de Señales Digitales del centro “Geoinformática y Señales Digitales” (GEYSED).

Después de realizar un estudio sobre dicho sistema en el proyecto Factoría de Software del departamento antes mencionado, se decidió por los problemas que presentaba realizar un nuevo Sistema de Videoconferencia.

Un Sistema de Videoconferencia no es más que una tecnología que proporciona un sistema de comunicación bidireccional de audio, video y datos, que le permite a las sedes receptoras y emisoras mantener una comunicación simultánea interactiva en tiempo real. (Hidalgo, 2009)

Un Sistema de Videoconferencia necesita contar con una BD para el almacenamiento de la información que maneja o manipula. Las mismas no son más que un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Un elemento importante dentro del entorno de las BD es la capa de acceso a datos que sirve como puente entre la capa lógica del negocio y el proveedor de datos. La misma pretende encapsular las especificidades del proveedor tales como (SQL, Oracle, archivos XML, texto, hojas electrónicas). (Server, 2007)

Actualmente en el proyecto productivo Factoría de Software de la Facultad 6 se está trabajando en el nuevo Sistema de Videoconferencia para lograr una buena comunicación entre los usuarios. El mismo no cuenta con una herramienta que permita lograr la unificación, organización y almacenamiento de todo el extenso volumen de información que se originará con la realización de las funcionalidades para este sistema. El control de la información constituye una necesidad para la dirección de este proyecto productivo y para los futuros clientes que trabajarán en él, ya que brinda a los usuarios una forma más rápida de acceder a la información proporcionada por la aplicación.

Según las situaciones planteadas surge entonces como **problema a resolver**: ¿cómo lograr que el Sistema de Videoconferencia cuente con un manejo especializado de la información?

Una vez identificado el problema a resolver se define como **objeto de estudio**: el proceso de gestión de información en sistemas de Video Conferencias. Definiéndose como **campo de acción**: automatización del procesamiento de los datos del Sistema Video Conferencia.

Capítulo 1: Fundamentación teórica

Para dar solución al problema planteado surge como **objetivo general**: desarrollar la capa de acceso a datos e implementar el modelo físico de la BD del Sistema Video Conferencia de la Facultad 6.

La **idea a defender es**: con el diseño e implementación de la BD y la capa de acceso a datos del Sistema de Videoconferencia de la Facultad 6, se garantizará la unificación y seguridad de la información y una rápida conexión de dicho sistema.

Para darle cumplimiento al objetivo trazado se determinó que las **tareas de la investigación científica** estarían encaminadas a:

- Caracterizar los sistemas de Video Conferencias.
- Caracterizar las diferentes arquitecturas de diseño de BD
- Documentar las tendencias y tecnologías actuales de BD para escoger cuáles serán aplicadas.
- Caracterizar los diferentes patrones de diseño para capas de acceso a datos para escoger el más indicado.
- Diseñar la BD.
- Modelar las clases de la capa de acceso a datos.
- Implementar la BD.
- Implementar la capa de acceso a datos.
- Validar los resultados obtenidos.

Para el desarrollo de la investigación fueron utilizados un conjunto de métodos científicos, dentro de los que se encuentran los métodos teóricos, que permiten estudiar las características que no son observables del modelo de investigación, y los métodos empíricos que facilitan la descripción de las características del objeto.

Dentro de los teóricos se utilizaron los siguientes:

- Histórico–Lógico: se utilizó para el estudio de los antecedentes existentes con respecto a la documentación que se quiere mostrar.
- Modelación: permite crear modelos donde se plasmen los resultados que se deben obtener mediante la investigación.

Capítulo 1: Fundamentación teórica

- Analítico–Sintético: para analizar toda la documentación existente con respecto a la visualización de la documentación referente al Sistema Video Conferencia que se va a realizar.

Los métodos empíricos utilizados:

- Entrevista: se realizaron entrevistas al tutor y jefes que laboran en el proyecto Factoría de Software para la obtención de la documentación apropiada para el desarrollo de la investigación.
- Observación: para obtener la información que se quiere visualizar en la BD del Sistema Video Conferencia.

El Trabajo de Diploma ha sido conformado de la siguiente manera:

Capítulo 1: este capítulo contiene los conceptos asociados al dominio del problema, las metodologías utilizadas en el mundo para su creación, las herramientas empleadas para el diseño de BD y se definen las que se utilizaron en la confección de la BD del Sistema de Videoconferencia del proyecto Factoría de Software del departamento de Señales Digitales de la Facultad 6.

Capítulo 2: en este capítulo se presenta el diagrama de clases persistentes del Sistema de Videoconferencia, se realiza la normalización de las tablas de la BD y se describe el diseño físico de la misma. Además se abarca todo lo referente a la implementación de la capa de acceso a datos del Sistema de Videoconferencia.

Capítulo 3: en este capítulo se brinda información sobre la validación realizada al diseño de la BD propuesta, se muestran las reglas a tener en cuenta para obtener un diseño de óptima calidad, así como la selección de consultas para un llenado voluminoso e inteligente de la BD para validarla funcionalmente.

Capítulo 1: Fundamentación teórica

Capítulo 1: Fundamentación teórica

1.1. Introducción

Este capítulo está dirigido a trazar todos los elementos teóricos que sustentan el objeto de estudio y el objetivo de la investigación científica. Se muestran todos los conceptos que desde el punto de vista teórico permiten un mejor entendimiento de lo que se plantea en la situación problemática y en el marco del problema en sentido general, con el objetivo de proporcionar mayor cantidad de información al respecto y enfatizar sobre la situación problemática donde coexiste el objeto de estudio.

Se aborda también temas relacionados con el origen y desarrollo de las BD y la importancia de las mismas, las herramientas empleadas para el diseño de BD y de capa de acceso a datos, se definen las que se utilizarán en la confección de la BD del Sistema de Video Conferencia de la Facultad 6.

Además se realiza la propuesta de desarrollo de la capa de acceso a datos y la implementación del modelo físico de la BD como objetivo general de la investigación, aclarando las ventajas que ofrecería este procedimiento para el Sistema de Video Conferencia que se va a realizar en el departamento Señales Digitales de la Facultad 6.

Por último se realiza un estudio sobre las soluciones que existen y brindan una solución parcial o total al objeto de estudio planteado. Se realiza un análisis de las mismas con el fin de facilitar la comprensión de la importancia de la investigación, su alcance y su aporte científico.

1.2. Conceptos asociados al dominio del problema

Con el objetivo de proporcionar al lector un mejor entendimiento de los temas que serán abordados en la investigación, se detallan a continuación una serie de conceptos asociados al dominio del problema entre los que se destacan: Capa, Capa de acceso a datos, Modelo, Modelo físico de datos, Diagrama Entidad Relación, Base de datos, Sistema de Información, Videoconferencia, Gestión de Información.

- **Capa:** en el modelo de acceso a datos, una capa es un nivel lógico en el cual residen componentes o aplicaciones lógicas. Las capas pueden residir en uno o más equipos o servidores, el número de capas hace referencia al número de niveles y no al número de equipos en los cuales los servicios son divididos. (Sánchez, 2005)

Las capas que generalmente se incluyen en aplicaciones son:

Capítulo 1: Fundamentación teórica

- **Capa de Cliente:** conocida como capa de Presentación, es la que contiene las interfaces en las que el usuario interactúa con el sistema. (Sánchez, 2005)
- **Capa de la Lógica de Negocios:** contiene la lógica que interactúa con el origen de datos. Esta capa intermedia contiene la parte de la aplicación que interactúa con los datos, por ejemplo: la creación de una cadena de conexión al origen de datos. (Sánchez, 2005)
- **Capa de acceso a datos:** se relaciona directamente con el origen de datos.
- **Modelo:** son una representación de la realidad que contiene las características generales de algo que se va a realizar. En las BD, esta representación se elabora de forma gráfica. (Codd, 2009)

Los modelos o abstracciones son los elementos que en primera instancia permiten tener una aproximación al diseño de las BD. A estos modelos se les llama modelos semánticos y existen diferentes representaciones para ellos. La idea de tener modelos semánticos es contar con un modelo que permita capturar el significado de los datos y como ellos se pueden evolucionar. (Rozic, 2004)

- **Modelo de Entidad-Relación:** fue desarrollado en el año 1976. Este se centra en buscar las entidades básicas del modelo y luego la relación que existe entre ellas. (Rozic, 2004)
- **Modelo de datos:** describen cómo se representan los datos, sea en una empresa, en un sistema de información o en un sistema de gestión de BD.

En la informática, un modelo de datos es un lenguaje utilizado para la descripción de una BD. Por lo general, un modelo de datos permite describir las estructuras de datos de la BD, el tipo de los datos que incluye y la forma en que se relacionan, las restricciones de integridad, las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos). (WP Daily, 2008)

Además estos tipos de modelos permiten describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

- **Modelo físico de datos:** se usan para describir los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de BD. (Angie Pao 29, 2010)

Capítulo 1: Fundamentación teórica

- **Diagrama Entidad Relación (DER):** constituye una técnica para representar la estructura lógica de una BD en forma de gráficos. Proporcionan un medio sencillo y de fácil comprensión para comunicar las características sobresalientes del diseño de cualquier BD. (Date, 2001)
- **Base de Datos:** una BD es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones, sistemas o programas de una organización. (Rozic, 2004)

Una BD es el conjunto de informaciones almacenadas en un soporte legible por ordenador y organizadas internamente por registros. Los mismos están formados por todos los campos referidos a una entidad u objeto almacenado. Los campos no son más que cada uno de los elementos que componen un registro. (Pino, 2004)

- **Sistema de Información:** un sistema de información (SI) representa todos los elementos que forman parte de la administración, el procesamiento, el transporte y la distribución de la información dentro de la compañía. (Jeff, 2008)

Los SI son un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de la organización. (Laudon, 2006)

- **Videoconferencia:** una videoconferencia es la comunicación simultánea bidireccional de audio y vídeo, permitiendo mantener reuniones personales o con grupos de personas situadas en lugares alejados entre sí. Adicionalmente, los equipos pueden tener la capacidad de mostrar o compartir presentaciones, archivos y videos y/o conectarse entre múltiples ubicaciones (multipunto) a la vez, resultando una comunicación total entre ellos. (Barrios, 2009)

La videoconferencia es un sistema de comunicación que permite mantener reuniones colectivas entre varias personas que se encuentran en lugares distantes. Esta comunicación se realiza en tiempo real y se transmite tanto la imagen como el sonido. (Cordero, 2008)

- **Gestión de información:** la gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización. (Marrero, 2003)

1.3. Objeto de estudio

Las BD como un medio informático de gran importancia y utilidad en el desarrollo de muchas esferas han servido para el almacenamiento de grandes volúmenes de información que diariamente se

Capítulo 1: Fundamentación teórica

generan en cada centro de la universidad. A grandes rasgos se puede decir que las BD agrupan todos los datos relevantes de los centros laborales. La expresión datos relevantes hace referencia a aquella información que por su significado será necesaria mantenerla almacenada, ya que sustentará las actividades del negocio y la toma de decisiones correspondientes.

Las BD logran que la información se encuentre en un único lugar determinado, esto implica que los datos se encuentren lógicamente unificados e interrelacionados, constituyendo un todo que debe, por lo tanto, diseñarse, administrarse y usarse desde un punto de vista global.

1.3.1 Descripción actual del dominio del problema

La UCI constituye en la actualidad un centro de estudios universitarios de gran categoría, la misma posee dos objetivos importantes: informatizar el país y desarrollar la industria del Software para contribuir al desarrollo económico del mismo. En dicho centro se combinan de manera directa tres aspectos para la producción de software y para la preparación científica de toda la comunidad en general: la docencia, la producción y la investigación. Por lo que a largo de su desarrollo se ha creado una serie de temáticas que forman parte de la investigación de cada Facultad en particular. Las mismas han favorecido la evolución del software, lo que garantiza el crecimiento económico de cualquier país del mundo.

Actualmente los Centros Productivos que conforman la UCI dirigen toda la investigación y desarrollo a una determinada área temática o tema de investigación de un sector determinado. Dentro de los mismos se encuentran los equipos de desarrollo llamados Proyectos Productivos que son los encargados de desarrollar las soluciones informáticas dirigidas al tema que los ocupa. Estos proyectos están conformados por una serie de profesores y estudiantes que son los autorizados a seleccionar todos los requerimientos que debe tener el software a realizar.

A la UCI, la empresa Huawei le facilitó un Sistema de Videoconferencia, el mismo le fue asignado al proyecto productivo Factoría de Software de la Facultad 6. A este sistema se le tenían que realizar nuevas funcionalidades para adaptarlo a las necesidades de dicha Universidad, pero esto no se pudo efectuar ya que el sistema está bajo licencia propietaria y no se tiene acceso al código fuente.

Debido a esta situación se decidió realizar un nuevo Sistema de Videoconferencia para permitir la interacción visual, auditiva y verbal entre personas distantes, desde cualquier lugar. Esto juega un papel importante en la comunicación entre las Facultades Regionales (Mini UCI) y la UCI, ya que

Capítulo 1: Fundamentación teórica

permitirá la participación en clases, conferencias y eventos realizados en cualquiera de estas sedes universitarias.

1.3.2 Situación problemática

Actualmente la posibilidad de contar con una BD en el Sistema de Videoconferencia que se va a realizar en el proyecto productivo Factoría de Software del departamento de Señales Digitales de la Facultad 6 es de una gran ventaja y ayuda para este local, ya que se contará con una herramienta que servirá de apoyo a los usuarios a la hora de guardar la información con la que se está trabajando, además esta BD contará con una capa de acceso a datos que permitirá una conexión eficiente puesto que muchas conexiones efectuadas a la misma vez no afectarán la velocidad de transferencia de datos.

En los centros donde se manipula cantidades considerables de información las BD tienen gran importancia, ya que permiten de forma rápida y flexible el almacenamiento, la recuperación, la organización, la impresión y distribución de los datos almacenados.

Debido a las ventajas que ofrecen las BD para el aseguramiento y control de todo tipo de documentación referente a cualquier sistema a realizar, surge en el Proyecto Factoría de Software del departamento de Señales Digitales de la Facultad 6 la necesidad de crear un sistema de esta índole para dirigirlo al Sistema de Videoconferencia que se va a realizar.

La necesidad de contar con una herramienta para guardar y gestionar la información referente a las funcionalidades que se harán para la realización del Sistema de Videoconferencia de la Facultad 6, constituye la principal meta a seguir, pues poseer tal producto constituiría una vía importante para los futuros negocios que se puedan realizar dentro del Centro GEySED.

1.4 Estudio de los Sistemas de Videoconferencia

El desarrollo de este epígrafe se centra en el análisis de un conjunto de soluciones que existen vinculadas al desarrollo de la BD para el Sistema de Videoconferencia del proyecto Factoría de Software de la Facultad 6. Además pretendiendo corroborar la necesidad de contar con una BD que permita la centralización y control de la información referente al Sistema de Videoconferencia que se va a realizar. Se analizan también algunos sistemas de este tipo que poseen estas características.

Los negocios cara a cara en la actualidad son una práctica costosa, con alto consumo de tiempo, en lo cual se hace uso del teléfono, el fax o el módem. La videoconferencia ofrece hoy en día una solución

Capítulo 1: Fundamentación teórica

accesible a esta necesidad de comunicación, con sistemas que permiten transmitir y recibir información visual y sonora entre puntos o zonas evitando así los gastos y pérdida de tiempo que implican el traslado físico de la persona. Estas ventajas hacen a la videoconferencia el segmento de mayor crecimiento en el área de las telecomunicaciones. El hecho de poder reunir a personas que se encuentran en diferentes lugares a cualquier hora o momento del día, proporciona múltiples ventajas como: ahorro de tiempo y dinero en viajes, posibilidad de compartir y modificar datos en tiempo real y tener a varias personas conectadas en una misma conferencia.

1.4.1 Elementos básicos de un Sistema de Videoconferencia

Para fines de estudio y de diseño los sistemas de videoconferencia suelen subdividirse en tres elementos básicos que son: la red de comunicaciones, la sala de videoconferencia y el códec.

- **La red de comunicaciones**

Para poder realizar cualquier tipo de comunicación es necesario contar primero con un medio que transporte la información del transmisor al receptor y viceversa o paralelamente (en dos direcciones). En los sistemas de videoconferencia se requiere que este medio proporcione una conexión digital bidireccional y de alta velocidad entre los dos puntos a conectar. (González-Aponcio, 2010)

- **La Sala de Videoconferencia**

La sala de videoconferencia es el área especialmente acondicionada en la cual se alojará el personal de videoconferencia, así como también: el equipo de control, de audio y de video que permitirá capturar y controlar las imágenes y los sonidos que habrán de transmitirse hacia él o los puntos remotos. (González-Aponcio, 2010)

- **El Códec**

Las señales de audio y video que se desean transmitir se encuentran por lo general en forma de señales analógicas, por lo que para poder transmitir esta información a través de una red digital, esta debe de ser transformada mediante algún método a una señal digital, una vez realizado esto se debe de comprimir y multiplexar estas señales para su transmisión. El dispositivo que se encarga de este trabajo es el códec (Codificador/Decodificador) que en el otro extremo de la red realiza el trabajo inverso para poder desplegar y reproducir los datos provenientes desde el punto remoto. (González-Aponcio, 2010)

Capítulo 1: Fundamentación teórica

1.4.2 Modalidades de Videoconferencia

Existen tres modalidades de videoconferencia básicas que se clasifican de acuerdo al tipo de uso que se les da: (Modalidades)

- **Sistemas para PC:** es un sistema en el cual dos personas se comunican transmitiendo señales de audio y vídeo, de acuerdo al software utilizado, también se pueden transmitir archivos.
- **Sistemas de Sobremesa:** son sistemas para grupos medianos o reducidos, se transmiten señales de audio y vídeo, de acuerdo al equipamiento utilizado, también se pueden transmitir archivos.
- **Sistemas de Sala:** es una multi-videoconferencia donde pueden participar un enorme número de personas activamente transmitiendo señales de audio, video y archivos.

1.4.3 Estándares de Videoconferencia

Los estándares utilizados para videoconferencia son establecidos por la Unión Internacional de Telecomunicaciones (ITU). Los cuales definen un conjunto de protocolos para proveer comunicación visual y de audio sobre una red. También permiten obtener una buena calidad en la transmisión de videoconferencia a velocidades iguales o superiores a 384 kbps. Esta exige cumplir con una de las siguientes normas: (González-Aponcio, 2010)

- H320: describe normas para la videoconferencia punto a punto y multipunto en las Redes Digitales de Servicios Integrados ISDN. Este estándar gobierna los conceptos básicos para el intercambio de audio y vídeo en el proceso de comunicación. (González-Aponcio, 2010)
- H323: basado en el protocolo de Internet IP, define la forma cómo los puntos de la red transmiten y reciben llamadas, compartiendo las capacidades de transmisión de audio, vídeo y datos. (González-Aponcio, 2010)

Hoy en día la videoconferencia es una herramienta útil para la comunicación personalizada a larga distancia. En la UCI podría ser utilizada para realizar charlas, conferencias, congresos, dictar clases y realizar reuniones de profesores de forma más eficiente.

Para estudiantes de Ingeniería Informática es de gran importancia, ya que es posible realizar clases con profesores de más experiencia que no se encuentran en la universidad o comunicarse con

Capítulo 1: Fundamentación teórica

alumnos de otras universidades que estudien esta carrera o alguna relacionada con el fin de compartir experiencias.

1.5 Análisis de soluciones existentes

❖ DIMDIM

Es una compañía de software que provee una plataforma web para colaboración y reuniones en tiempo real. Es percibida como una solución de conferencias de la "Web 2.0" gratuita y de código abierto. Dimdim provee el servicio de conferencias web gratuitas donde los usuarios pueden compartir escritorios, mostrar presentaciones, colaborar, chatear, conversar vía audio y transmitir desde una webcam. No es necesario descargar ningún programa para utilizarlo. Solo la persona que es el administrador de la presentación tiene que bajar un plugin.

Algunos de los servicios que ofrece es compartir tres micrófonos y chatear de modo público y privado. Se puede grabar el audio de la videoconferencia, las mismas pueden durar hasta un máximo de 5 horas. (Mendez, 2009)

Algunas de las principales características que posee Dimdim son:

- Es el soporte de comunicaciones multimedia en tiempo real.
- Mediante el tablero blanco que trae incluido se pueden plasmar ideas y soluciones.
- Mediante la pizarra interactiva todos pueden participar, realizando correcciones, preguntas o sugerencias, y siempre en tiempo real.
- Puede ser integrada con las plataformas Moodle, Claroline y la suite colaborativa Zimbra.

❖ RECOL

Ofrece un nuevo servicio muy útil para colegios, asociaciones profesionales y cualquier empresa o entidad. Es un Sistema de Videoconferencia a través de Internet que permite la conexión entre distintos usuarios a través de PC, móvil o H323 (videoconferencia tradicional), sin ninguna instalación previa y con la posibilidad de alquilar licencias de uso (por horas, días o semanas) y que puedan ser compartidas a lo largo del día por tantas personas como se desee. El costo de este servicio, que ahorra desplazamientos y fomenta la conectividad entre empresas y trabajadores es muy inferior al de una videoconferencia tradicional y no exige invertir en un costoso equipo, además de ahorrar en la factura de teléfono al conectarse por Internet. Permite a los conferencistas y usuarios conectados compartir archivos o navegar juntos por Internet.

Capítulo 1: Fundamentación teórica

RECOL también ofrece poder intercalar en la videoconferencia a traductores simultáneos en 49 idiomas. Se trata de un servicio de web de conferencia. Es una aplicación informática que funciona sobre plataforma UNIX y que permite la videoconferencia entre diversos terminales simultáneos, así como compartir aplicaciones entre ellos, enviarse ficheros, navegar conjuntamente por Internet o participar en un chat general o selectivo. Además, la información se transmite con cifrado en tiempo real para todos los servicios a través de http, lo que aporta al servicio una confidencialidad total. (RECOL, 2007)

El sistema de videoconferencia RECOL ofrece una serie de ventajas:

- La videoconferencia se hace a través de Internet, con lo que no hay coste de llamada.
- Pueden participar en la videoconferencia un número ilimitado de usuarios a la vez, incluido el moderador.
- Los usuarios pueden compartir aplicaciones, actualizar o corregir simultáneamente archivos o navegar conjuntamente por Internet.
- El sistema permite intercalar uno o varios traductores simultáneos (Recol ofrece también este servicio de traducción simultánea en 49 idiomas, en caso de necesitarse).
- Las sesiones de videoconferencia pueden grabarse y reproducirse.

❖ **BigBlueButton**

Es una alternativa gratuita y de código abierto, pues con él es posible realizar las famosas conferencias en línea, también conocidas como Webinars. Se puede integrar BigBlueButton con cualquier plataforma sin tener que realizar ningún tipo de pago, por lo que resulta perfecto para un simple uso personal o bien algo más profesional como un uso comercial.

Uno de los grandes aciertos de esta utilidad es que la distribución de sus funciones es bastante acertada, su aspecto es bastante limpio y organizado. Así no habrá ningún problema al momento de realizar chat, compartir el escritorio, realizar presentaciones, enviar videos vía webcam, mantener una comunicación por voz o tener controlada la lista de participantes. Gracias a que su código es totalmente libre, se puede realizar el número de modificaciones que se desee, así cualquiera podrá integrarle más y mejores funciones. (Softhoy, 2010)

1.5.1 Observaciones sobre el análisis de las soluciones existentes

Capítulo 1: Fundamentación teórica

Luego de realizar un análisis sobre las herramientas que existen referentes a los Sistemas de Videoconferencia se puede decir que una de las desventajas principales que poseen es que están soportadas sobre software propietario, lo que no permite el uso libre del producto. También se analizó el BigBlueButton que el mismo no es propietario, pero no se ajusta a las características que se quiere en el proyecto, presenta dificultades con la instalación y su uso es complejo. A pesar de esto se puede resaltar que las mismas constituyen soluciones útiles dentro del mundo de los Sistemas de Videoconferencia y dentro de las temáticas que abarcan se pueden situar en un lugar meritorio debido a la información que permiten recopilar.

En el análisis de las metodologías a utilizar para el desarrollo del Sistema de Videoconferencia a realizar en el proyecto Factoría de Software de la Facultad 6 se obtuvieron pasos a tener en cuenta para el planteamiento de la propuesta, como por ejemplo: implementar la capa de acceso a datos y el modelo físico de la BD para este sistema, seleccionar las tendencias y tecnologías a utilizar en la misma, implementar las funcionalidades que desea el cliente que tenga el sistema y probarlas.

1.6 Características de las arquitecturas de diseño de BD

Para alcanzar la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la BD, en 1975, el comité ANSI-SPARC (del inglés Instituto Americano Nacional de Estándar –Normas Comité de Planificación y Requisitos) propuso una arquitectura de tres niveles. (Mato, 1999)

La arquitectura se divide en tres niveles generales: interno, externo y conceptual.

- ❖ **Nivel interno:** es también conocido como el nivel físico, es el que está más cerca del almacenamiento físico; es decir, es el que tiene que ver con la forma en que los datos están almacenados físicamente. El mismo describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento en la BD. (Mato, 1999)
- ❖ **Nivel externo:** es también conocido como el nivel lógico de usuario, es el más próximo a los usuarios; es decir, el que tiene que ver con la forma en que los usuarios individuales ven los datos. El mismo describe varios esquemas externos o vistas de usuario. Cada uno describe la parte de la BD que interesa a un grupo determinado de usuarios y oculta a ese grupo el resto de la BD. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas. (Mato, 1999)

Capítulo 1: Fundamentación teórica

- ❖ **Nivel conceptual:** es también conocido como el nivel lógico de la comunidad, o en ocasiones sólo como el nivel lógico, sin calificar. También describe la estructura de toda la BD para una comunidad de usuarios mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y describe entidades, atributos, relaciones, operaciones de los usuarios y restricciones. (Mato, 1999)

La arquitectura de tres niveles es útil para explicar el concepto de **independencia de datos** que se dividen en dos: Independencia lógica e Independencia física.

- ❖ **Independencia lógica:** es la inmunidad de los usuarios y los programas de usuario ante los cambios en la estructura lógica de la BD donde por estructura lógica se entiende el nivel conceptual o lógico de la comunidad. (Mato, 1999)
- ❖ **Independencia física:** es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual o los esquemas externos.

1.7 Tendencias y tecnologías actuales de BD

Para la realización del área de trabajo a elaborar se dividieron las herramientas necesarias en tres categorías de las cuales se debe seleccionar una tecnología en particular como son:

- ✓ Tipos de BD.
- ✓ Gestor de BD.
- ✓ Herramientas CASE.

1.7.1 Tipos de BD

Para crear una BD se deben realizar dos ejercicios de diseño: un diseño lógico y uno físico. El diseño lógico es un modelo abstracto de la BD desde una perspectiva de negocios, mientras que el diseño físico muestra como la BD se ordena en realidad en los dispositivos de almacenamiento de acceso directo.

El proceso de diseño identifica las relaciones entre los elementos de datos y la manera más eficiente de agruparlos para cumplir con los requerimientos de información. El proceso también identifica elementos redundantes y los agrupamientos de los elementos de datos que se requieren para programas de aplicaciones específicos. Los grupos de datos son organizados, refinados y agilizados

Capítulo 1: Fundamentación teórica

hasta que una imagen lógica general de las relaciones entre todos los elementos en la BD surja. (Laudon, 2004)

En el mundo existen varios tipos de BD brindan una serie de ventajas a la hora de manejar la información dentro de las cuales se encuentran:

❖ **Según la variabilidad de los datos almacenados:**

- **Bases de datos estáticas:** estas son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo. (Aguilar, 2011)
- **Bases de datos dinámicas:** estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. (Aguilar, 2011)

Después del análisis realizado de los tipos de BD, se seleccionó la BD dinámica, porque a diferencia de los demás tipos de BD la información está en constante cambio, ya que sobre ella se pueden realizar operaciones tales como: adicionar, actualizar y eliminar. Además de que son ideales para almacenar grandes volúmenes de información, como la que se originará en el Sistema de Videoconferencia que se va a realizar en el proyecto productivo Factoría de Software de la Facultad 6.

Algunos de los modelos con frecuencia utilizados en las BD son:

- ❖ **BD documentales:** son las derivadas de la necesidad de disponer de toda la información en el puesto de trabajo y de minimizar los tiempos de acceso a aquellas informaciones que se utilizan con frecuencia, no están estructuradas convenientemente. Esto se debe a que la procedencia de la información es muy variada: informes, notas diversas, periódicos y revistas. (Schmieg, 2008)
- ❖ **BD distribuidas:** un sistema de BD distribuidas se compone de un conjunto de nodos conectados entre sí mediante algún tipo de red de comunicaciones, en el cual cada sitio es un sistema de BD en sí mismo. Todos los sitios han convenido en trabajar juntos (si es necesario) con el fin de que un usuario de cualquier sitio pueda obtener acceso a los datos de cualquier punto de la red, tal como si todos los datos se encontraran almacenados en el sitio propio del usuario. (Rozic, 2004)
- ❖ **BD orientadas a objetos:** estas son capaces de almacenar tanto procesos como datos. Por este motivo las bases orientadas a objeto deben poder almacenar tanto información no convencional como imágenes estáticas o en movimiento, colecciones de sonidos, entre otros. Este tipo de BD

Capítulo 1: Fundamentación teórica

deriva directamente de la llamada programación orientada a objetos, típica por ejemplo del lenguaje C/C++. (Schmieg, 2008)

- ❖ **BD jerárquica:** fueron concebidas en los años 1960. Estas como su nombre lo indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol al revés en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas. (Jahzeel, 2007)

Las BD jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. (CORREA, 2010)

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

- ❖ **BD de red:** este es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo, se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). (Jahzeel, 2007)

También ofrece una gran ventaja sobre el modelo jerárquico, ya que presenta una solución eficiente al problema de redundancia de datos; pero, aun así, administrar la información en una BD de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- ❖ **BD relacionales:** este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia, a diferencia de otros modelos como el jerárquico y el de red. Es más fácil de entender y de utilizar para un usuario esporádico de la BD. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información. (Jahzeel)

- ❖ **BD deductivas:** es un sistema de BD pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la BD. Las mismas son llamadas también BD lógicas, a raíz de que se basa en lógica matemática. (Jahzeel)

- **BD seleccionada**

Capítulo 1: Fundamentación teórica

Entre las BD analizadas se elige la BD Relacional, la misma es un modelo que se usa con más frecuencia en la práctica, debido a las ventajas que ofrece sobre los demás modelos analizados anteriormente, como lo es el rápido entendimiento por parte de los usuarios. También brinda un ambiente agradable ya que permite que los usuarios familiarizados con la Programación Orientada a Objeto tengan una facilidad a la hora del trabajo con entidades (tablas), compuestas por registros (las filas de una tabla), que representarían las tuplas (las mismas pueden representar tanto entidades como interrelaciones), y campos (las columnas de una tabla). Los lenguajes matemáticos sobre los que se asienta el modelo relacional, el álgebra y el cálculo relacionales, aportan un sistema de acceso y consultas que favorece el trabajo con los datos. Además de los altos niveles de fiabilidad e integridad que aporta en el manejo de grandes cantidades de datos, esto brinda una gran confianza a la hora de gestionar el control de toda la documentación que se proporcionará en la realización de las funcionalidades del Sistema de Videoconferencia que se va a realizar.

1.8 Sistemas Gestores de BD Relacionales

Un sistema gestor de BD consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. (Abraham Silberschatz, 2002)

- ❖ **PostgreSQL:** es llamado también Postgres95, fue derivado del proyecto Postgres. Este proyecto lleva más de una década de desarrollo, siendo hoy en día uno de los sistemas libres más avanzados que existe, soportando la gran mayoría de las transacciones SQL, teniendo a su disposición varios lenguajes como por ejemplo C, C++, Java, Python, PHP y muchos más.

PostgreSQL es un Sistema de Gestor de BD Objeto-Relacional (ORDBMS) basado en el proyecto Postgres, de la Universidad de Berkeley. Fue el primero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como pueden ser: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de BD puramente orientado a objetos. (Faraon, 2011)

Entre otras características que presenta se encuentran las siguientes: (Faraon, 2011)

- Soporta distintos tipos de datos: además del soporte para los tipos BD. También soporta datos de tipo fecha, monetarios, elementos gráficos y cadenas de bits. También permite la creación de tipos propios.

Capítulo 1: Fundamentación teórica

- Incorpora una estructura de datos array.
- Incorpora funciones de diversas índoles: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

❖ Microsoft SQL Server

Es un sistema para la gestión de BD producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes Sistemas Gestores de Bases de Datos (EGBD) como son Oracle o MySQL. (Arbildo, 2010)

Algunas de sus características son

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

❖ Oracle

Es un Sistema de Gestión de BD relacional desarrollado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de BD más completos que se ha creado. Es la primera compañía de software que desarrolla e implementa software para empresas, a través de toda su línea de productos

Capítulo 1: Fundamentación teórica

de BD, aplicaciones comerciales, herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software. (Corporation, 2009)

Algunas características son:

- Soporte de transacciones.
- Estabilidad y escalabilidad.
- Soporte multiplataforma.

❖ MySQL

Es un sistema de gestión de BD relacional, multihilo y multiusuario con más de seis millones de instalaciones. Es hoy en día uno de los más importantes en el diseño y programación de BD de tipo relacional. Aparece en el mundo informático como una de las más utilizadas por los usuarios. (Lisy, 2010)

Las siguientes características son implementadas únicamente por MySQL: (Lisy, 2010)

- Múltiples motores de almacenamiento, permitiendo al usuario escoger la que sea más adecuada para cada tabla de la BD.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

➤ Gestor de BD seleccionado

De los gestores de BD antes analizados en la investigación se seleccionó PostgreSQL para este trabajo, porque el mismo brinda instalación ilimitada, a diferencia de MySQL que uno de los grandes defectos que posee es que no presenta un entorno gráfico que satisfaga a la mayor parte de los usuarios y está privatizado para el país, además con PostgreSQL no se puede hacer demanda por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia de este software. Además se ahorra considerablemente en los costos de operaciones, a diferencia de Oracle que es uno de los gestores más completos que existe, pero el mismo es un producto de elevado precio por lo que en general se utiliza en empresas muy grandes y multinacionales, además presenta elevada cantidad de documentación que para la realización de esta herramienta no es necesaria, ya que la BD que se va a realizar para este sistema es pequeña. Los costos de soporte técnico y mantenimiento son elevados, presenta además vulnerabilidades en la seguridad de la plataforma, por lo que se hace necesario

Capítulo 1: Fundamentación teórica

aplicar parches de seguridad. Además PostgreSQL es multiplataforma y brinda gran estabilidad y confiabilidad.

1.9 Herramientas CASE

Para confeccionar una BD se requiere de 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. Existen diferentes tipos de herramientas que se encargan del diseño de las BD.

❖ ERwin:

Es una herramienta para el diseño de BD, que brinda productividad en su 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 BD diseñada. Además ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la BD. Genera automáticamente las tablas y miles de líneas de procedimientos para los principales tipos de BD.

También hace fácil el diseño de una BD. Los diseñadores solo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad _ relación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y claves importantes. Establece una conexión entre una BD diseñada y una BD, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión, genera automáticamente tablas, vistas, índices, reglas de integridad referencial (claves primarias, claves foráneas), valores por defecto y restricciones de campos y dominios.

Soporta principalmente BD relacionales SQL y las que incluyen Oracle, Microsoft SQL Server. El mismo modelo puede ser usado para generar múltiples BD o convertir una aplicación de una plataforma de BD a otra. Permite especificar, analizar, diseñar el sistema antes de codificarlo. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. Tiene características como: (Valle, 2007)

- Mantiene la consistencia de los modelos del sistema de software.
- Chequeo de la sintaxis UML.
- Generación de Documentación automáticamente.
- Generación de Código a partir de los Modelos.

Capítulo 1: Fundamentación teórica

- Ingeniería Inversa (crear modelo a partir código).

❖ Visual Paradigm

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de mejor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es multiplataforma y muy útil para la generación de código fuente en PHP. 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 BD que incluyen PostgreSQL, MySQL, Microsoft SQL Server, Oracle. (Alonso, 2009)

❖ Rational Rose.

Es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Además de brindar una serie de beneficios, Rational Rose es un software propietario, lo que hace que tenga desventajas respecto a otras herramientas que son libres. (Menendez)

Una de las grandes ventajas de esta herramienta es que utiliza la notación estándar en la arquitectura de Software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Otra de sus ventajas es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

➤ Herramienta CASE seleccionada

De las herramientas anteriores se seleccionó el Visual Paradigm, por ser una herramienta muy potente para el propósito ingenieril. La misma lleva una ventaja sobre las demás, ya que posee versiones multiplataforma. Visual Paradigm es además completamente compatible con la metodología de desarrollo de software que se decidió utilizar ya que es el proceso que permite representar gráficamente el sistema software, permitiendo resaltar los detalles más importantes como son:

- Identifica requisitos y comunica información.
- Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos.
- Permite ver las relaciones entre los componentes del diseño.

Capítulo 1: Fundamentación teórica

- Mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Además utiliza UML y BPMN (Proceso de Notación y Modelación de Negocios) como lenguaje de modelado que este último es el que se va utilizar en este trabajo. Ya que BPMN es una especificación que proporciona una notación gráfica para expresar los procesos de negocio en un Diagrama de Procesos de Negocio (BPD). Su objetivo es apoyar la gestión de procesos de negocio tanto por usuarios técnicos como por usuarios de negocio, proporcionando una indicación de que es lo suficientemente intuitiva para los usuarios de negocios. Sin embargo, pueden representar la semántica de procesos complejos. (Emily Valenzuela Ortiz, 2010)

Visual Paradigm también está diseñada para dar soporte a arquitectos de sistemas, diseñadores, desarrolladores, analistas de procesos de negocio y modeladores de datos en los procesos de desarrollo de software. Además cuenta con un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad, es rápida y eficiente y proporciona las funcionalidades necesarias para un diseño profesional de BD como el necesario en este proyecto.

1.10 Framework de desarrollo a utilizar

Para la realización de la implementación de la capa de acceso a datos de la BD se orientó la utilización del framework de desarrollo Qt.

Qt Creator es un excelente IDE (entorno de desarrollo integrado) multiplataforma para desarrollar aplicaciones en C++ de manera sencilla y rápida. Como su nombre lo indica, está basado en la librería Qt y cuenta con las siguientes características principales: (Aguilera, 2009)

- Editor avanzado para C++.
- Diseñador de formularios integrado.
- Herramientas para la administración y construcción de proyectos.
- Completado automático.
- Depurador visual.

❖ Principales características de Qt Creator:

- Posee un avanzado editor de código C++.
- Además soporta los lenguajes: C#.NET Languages (Mono), Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.

Capítulo 1: Fundamentación teórica

- Posee también una GUI integrada y diseñador de formularios.
- Herramienta para proyectos y administración.
- Ayuda integrada sensible al contexto.
- Depurador visual.
- Resaltado y auto-completado de código.
- Soporte para refactorización de código.
- Es adaptable a varios lenguajes de programación y posee componentes propios que permiten un mayor desarrollo de aplicaciones.

1.11 Características del ORM para el mapeo de la BD.

- ❖ **ORM:** (Object Relational Mapping), es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el sistema de BD relacional utilizado en el desarrollo de una aplicación. (Carrero, 1998-2011)

Ventajas

- Rapidez en el desarrollo: la mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la BD leyendo el esquema. Crea el modelo adecuado.
- Abstracción de la BD: al utilizar un sistema ORM, lo que se consigue es separar totalmente el sistema de BD que se utilice, y así si en un futuro se debe de cambiar de motor de BD, se tendrá la seguridad de que este cambio no afectará al sistema.
- Reutilización: permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- Seguridad: los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser las inyecciones SQL.
- Mantenimiento del código: facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciéndolo mucho más sencillo.
- Lenguaje propio para realizar las consultas: estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta. (Carrero, 1998-2011)

Capítulo 1: Fundamentación teórica

Desventajas

- Inconsistencia entre el modelo relacional y el modelo orientado a objeto.
- Aplicaciones algo más lentas: esto es debido a que todas las consultas que se hagan sobre la BD, el sistema primero deberá de transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos. (Carrero, 1998-2011)

❖ Ejemplos de ORM orientados a QT:

- **ODB:** es un mapeo objeto-relacional (ORM) para C + +. Proporciona herramientas, APIs y soporte para las librerías que permiten persistir objetos en C + + con una BD relacional sin tener que lidiar con tablas, columnas o SQL de forma manual.

ODB es muy flexible y personalizable. Puede, o bien ocultar completamente la naturaleza relacional de la BD subyacente o exponer algunos de los detalles como sea necesario. Por ejemplo, se puede generar el esquema de BD relacional para sus clases persistentes, y hacer uso de la lengua objeto de consulta simple, segura y potente en lugar de SQL. Puede asignar tipos de SQL para los miembros de datos individuales, utilizando el esquema de BD existente, y ejecutar consultas SQL. (Kolpackou, 2011)

- **DQuest:** es un ORM para C++ integrado a Qt. Su objetivo es proporcionar un entorno de desarrollo rápido de aplicaciones con acceso a BD. Está diseñado para el entorno móvil, pero también es útil para equipos de sobremesa y aplicaciones.
- Con DQuest puede declarar un modelo de BD usando C + + directamente. La lectura / escritura se puede hacer a través de la interfaz C + +. No se necesita ninguna sentencia SQL para obtener el beneficio de usar Sqlite en su aplicación. El mismo posee las siguientes características: (Lau, 2011)

La declaración de BD del modelo y el registro es sencilla.

- Declara modelos en C + +.
- Apoyo a la herencia del modelo.
- Operaciones compatibles: índice de crear la tabla, seleccionar, borrar, insertar, consulta en la existencia de la tabla.
- Apoyo Sqlite, utilizable en la plataforma móvil de código abierto (licencia BSD).

Capítulo 1: Fundamentación teórica

- **QxOrm:** es un nuevo Object Relational Mapping (ORM) de código abierto desarrollado con bibliotecas C++, diseñado para proporcionar:
 - Persistencia (basado en módulo Qt /QtSql).
 - Serialización (XML y binario).
 - Reflexión (invocar métodos de la clase y el acceso a las propiedades).

Ha sido probado en Windows, Visual C + + 2008, 2010 y MinGW y Linux. Se basa en una función de ajuste no intrusiva y sencilla que se puede comparar con el archivo XML de mapeo de Hibernate. Tiene por objetivo simplificar C++ y proporciona una gran cantidad de funcionalidades. (Equipo de Qt apps, 2010)

❖ Características del ORM a utilizar en la implementación de la capa de acceso a datos.

- **Qdjango** es un sencillo pero potente asignador de objetos (ORM) escrito en C++ y construido en la parte superior de la biblioteca Qt. Siempre que sea posible se trata de seguir Django ORM API, de ahí su nombre. Es liberado bajo los términos de la Licencia Pública General GNU versión 3. Utiliza el patrón Active Record para el mapeo de la BD y como patrón de arquitectura el Modelo Vista Controlador (MVC). Qdjango se basa en Objetos de Qt Meta System, sus características son las siguientes: (Liané, 2003-2009)
 - Soporte para una amplia gama de BD.
 - Los campos de la BD se declaran usando Qt Q_PROPERTY macro.
 - Puede crear y eliminar BD y tablas de índices para los modelos registrados.
 - Acceso hilo consciente de la BD.
 - Apoyo para acceder a los modelos de QtScript.

❖ Pasos a seguir para utilizar Qdjango:

Qdjango posee dos carpetas Include y Lib. Estas deben ser incluidas en el .pro del proyecto para que Qt las reconozca; en el caso de la carpeta Include, la cual contiene los .h del ORM, se incluye en el INCLUDEPATH del .pro de la siguiente manera:

```
INCLUDEPATH += /home/yudirenia/Escritorio/Yudirenia/Include
```

En este caso /home/yudirenia/Escritorio/Yudirenia/Include es un ejemplo de la dirección donde se encuentra actualmente la carpeta Include.

Capítulo 1: Fundamentación teórica

La carpeta Lib, donde se encuentran las tres librerías utilizadas por Qdjango se incluye en el LIBS del .pro del proyecto de la siguiente manera:

```
LIBS += -L/home/yudirenia/Escritorio/Yudirenia/Lib -lqdjango -lqdjango-http -lqdjango-script
```

-L se escribe para que Qt sepa que son librerías.

/home/yudirenia/Escritorio/Yudirenia/Lib es la dirección donde se encuentran las librerías.

lqdjango -lqdjango-http -lqdjango-script son los nombres de las tres librerías utilizadas por Qdjango.

Ya incluidas las librerías y los .h se crean las clases correspondiente a cada tabla de la BD y se ponen a heredar de QDjangoModel para que esas clases tengan acceso a los métodos implementados en QDjangoModel, ejemplo: save, remove.

En la clase que va a hacer función de controladora del sistema, en la cual se vallan a implementar los métodos que harán referencia a la BD se incluye la clase Qdjango.h como clase principal del ORM y QDjangoQuerySet.h para tener acceso a las funciones SQL del ORM.

1.12 Conclusiones parciales.

Durante el desarrollo del capítulo fueron abordados todos los elementos teóricos que sustentan la investigación dentro de los que se pueden citar los conceptos asociados al dominio del problema que posibilitan un mejor entendimiento de lo planteado en la situación problemática. Fue realizado un análisis de las diferentes arquitecturas de diseño de BD, donde se analizaron las características y funcionalidades que las mismas poseen. También se realizó un análisis de las diferentes tendencias y tecnologías que existen y que mantienen una relación parcial con el objeto de estudio, realizando posteriormente una observación sobre estas con el objetivo de facilitar la comprensión de la importancia de la investigación y el alcance y aporte científico de la misma. Mediante la explicación detallada de la situación problemática abordada se logra el entendimiento del entorno donde coexiste el objeto de estudio de la investigación, pretendiendo conquistar en el lector la mayor claridad posible en los temas tratados.

Capítulo 2: Diseño de la solución propuesta.

Capítulo 2: Diseño de la solución propuesta

2.1 Introducción

En este capítulo se presentan las clases persistentes de la modelación de la BD del Sistema de Videoconferencia. Se describe el diseño físico de la BD y el modelo de la capa de acceso a datos. También se realiza el diagrama de componentes para la implementación de la capa de acceso a datos.

2.1.2 Modelo de objetos del negocio

El modelo de objetos del negocio o diagramas de clases del negocio es uno de los artefactos que se genera en el flujo de trabajo Modelación del Negocio. Es un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio. (Colectivo de profesores, 2010)

Para la identificación de las clases persistentes que representan entidades del Sistema de Videoconferencia en la BD se consultó el modelo de objetos diseñado por los analistas del mismo.

2.2 Diagrama de clases del diseño

Una clase de diseño es un elemento importante en la implementación del sistema. Esta describe un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y semántica. Se realizó un análisis de los diagramas de clases del diseño obtenidos del trabajo de los diseñadores del proyecto que facilitó la tarea de identificar las clases persistentes. (Marcello Visconti, 2008)

2.2.1 Clases persistentes

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente. Son aquellas que contienen toda la información valiosa e importante que necesariamente debe ser guardada en una BD.

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

Capítulo 2: Diseño de la solución propuesta.

genera a partir de la entrada del diagrama de clases del diseño propuesto por el rol de analista del proyecto.

A continuación se muestra el diagrama de clases persistente del Sistema de Videoconferencia en el que se representan las clases persistentes con sus atributos y las relaciones que existen entre dichas clases.

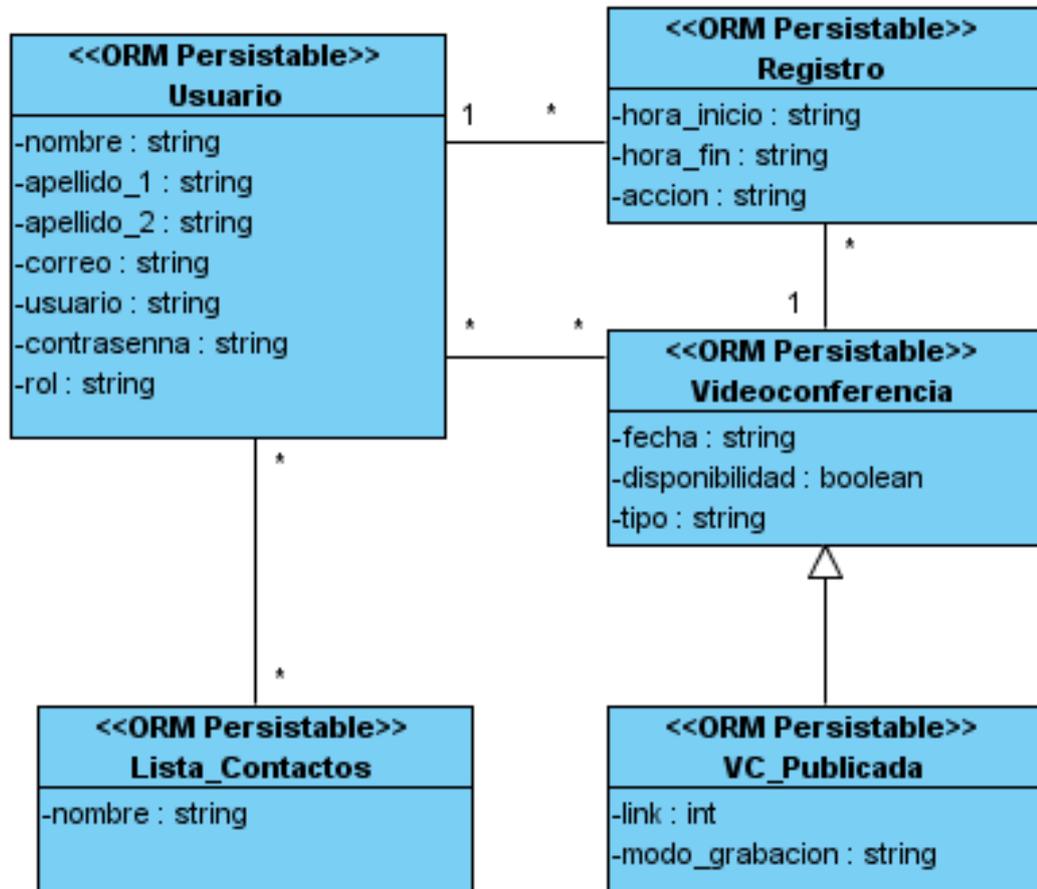


Ilustración 1: Diagrama de clases persistentes

2.3 Diseño de la BD

La BD para el Sistema de Videoconferencia del Proyecto Factoría de Software de la Facultad 6 es el medio para el almacenamiento de todos los datos relacionados dentro de dicho sistema y para acceder de forma controlada a los mismos.

Capítulo 2: Diseño de la solución propuesta.

El diseño de la misma es el proceso por el que se determina la organización de una BD, incluido su estructura, contenido y las aplicaciones que se han de desarrollar. El rendimiento adecuado del sistema se deberá a una correcta selección de las tablas que formarán parte de la BD, así como de la información que contendrá cada una de ellas, las relaciones que se establezcan entre las mismas y del correcto uso de las formas de normalización.

2.3.1 Modelo Entidad Relación (MER)

Un Diagrama o Modelo Entidad Relación a veces denominado por sus siglas, "DER" es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

Mediante el mismo se pretenden visualizar los elementos que pertenecen a una BD, que reciben el nombre de entidades, las cuales se corresponden con el concepto de clase de la Programación Orientada a Objeto y donde cada tupla de una futura relación representaría un objeto de la Programación Orientada a Objetos. (García, 2005)

En estos diagramas los rectángulos representan a las entidades y los rombos representan a las interrelaciones. Las interrelaciones son enlazadas con sus entidades por arcos, y el grado de la interrelación es indicado en el arco.

- **Entidades:** son objetos reales o abstractos, que pueden ser identificadas claramente y las mismas colecciona una serie cantidad de información; usualmente denotan a una persona, lugar, cosa o evento de interés informacional. Las entidades son descritas a través de atributos, las mismas pueden ser descritas por el mismo tipo de atributos son consideradas como conjuntos y referidas como un conjunto de entidades. Los conjuntos de entidades son representados en el diagrama del MER como un rectángulo. (García, 2005)
- **Interrelaciones:** son asociaciones o conexiones que existen entre dos o más entidades. Son representadas gráficamente por un rombo o diamante conectado a los conjuntos de entidades a través de líneas. Una interrelación debe tener como mínimo dos líneas de conexión. (García, 2005)
- **Asociación:** es una forma de abstracción en que una interrelación entre conjuntos de entidades con igual nivel de abstracción es considerada como un conjunto de mayor nivel. (García, 2005)
- **Grado:** está dado por un número de conjuntos de entidades asociados. Las interrelaciones unarias, binarias y ternarias tienen grado 1, 2 y 3 respectivamente. (García, 2005)

Capítulo 2: Diseño de la solución propuesta.

Para una asociación binaria R entre los conjuntos de entidades A y B, la cardinalidad máxima debe ser una de las siguientes: (García, 2005)

✓ **Uno a Uno:**

Una entidad en A se asocia con a lo sumo una entidad en B, y una entidad en B se asocia con a lo sumo una entidad en A.

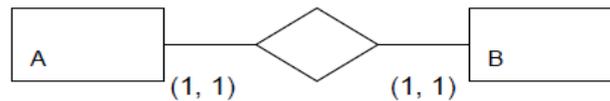


Ilustración 2: Cardinalidad uno a uno

✓ **Uno a Muchos:**

Una entidad en A se asocia con cualquier número de entidades en B. Una entidad en B, sin embargo, se puede asociar con a lo sumo una entidad en A.

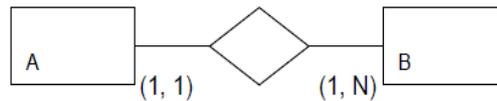


Ilustración 3: Cardinalidad uno a muchos

✓ **Muchos a Muchos:**

Una entidad en A se asocia con cualquier número de entidades en B, y una entidad en B se asocia con cualquier número de entidades en A.



Ilustración 4: Cardinalidad muchos a muchos

A continuación se mostrará el MER correspondiente a la BD del Sistema de Videoconferencia.

Capítulo 2: Diseño de la solución propuesta.

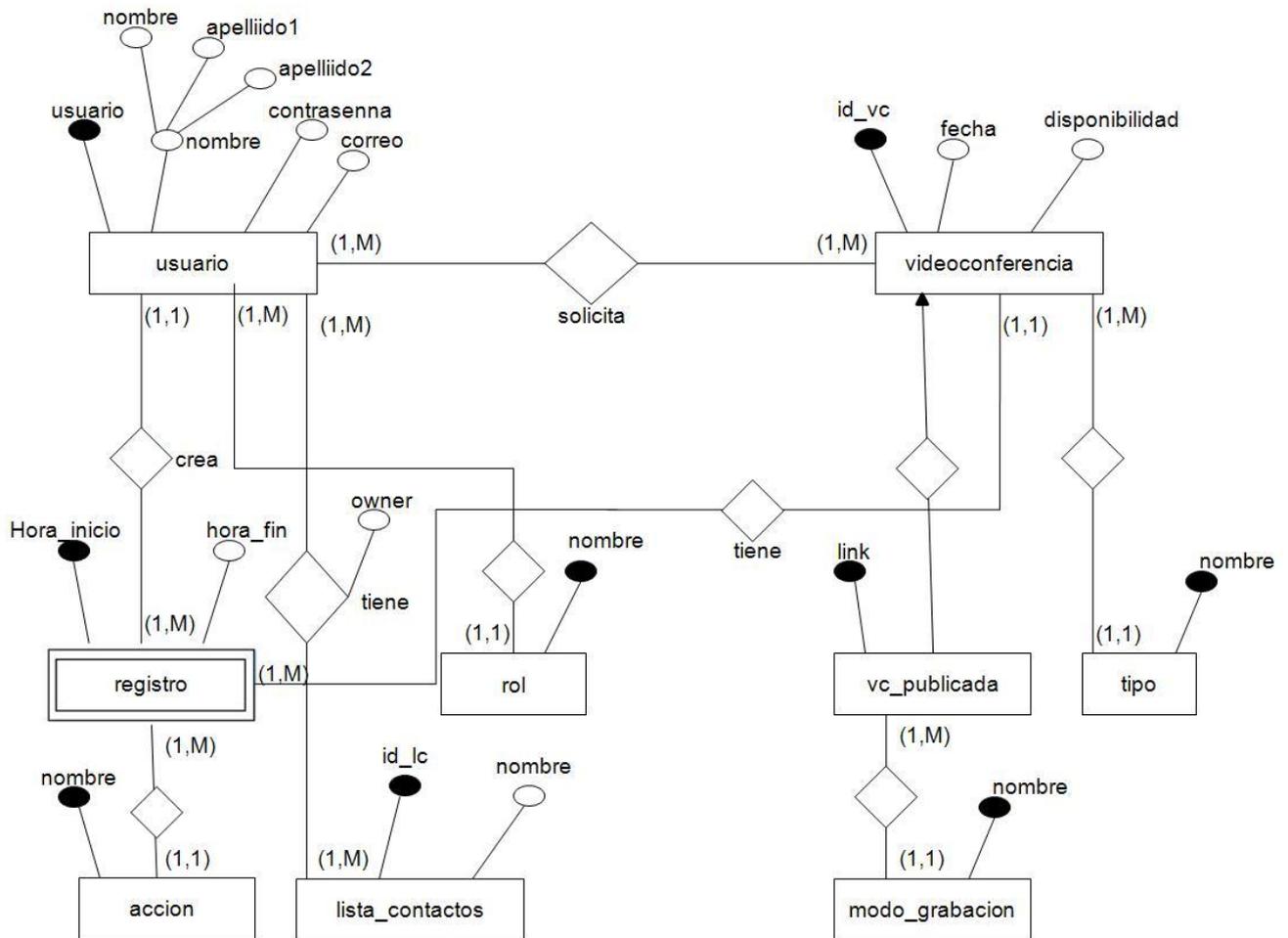


Ilustración 5: Modelo Entidad Relación

2.4 Normalización de la BD

La normalización de BD relacionales toma un esquema relacional y le aplica un conjunto de técnicas para producir un nuevo esquema que representa la misma información pero contiene menos redundancias y evita posibles anomalías en las inserciones, actualizaciones y borrados. Además este proceso consiste en comprobar en secuencia si el esquema original está en 1FN, 2FN y 3FN, analizando las dependencias funcionales en cada paso. (Sicilia, 2008)

Capítulo 2: Diseño de la solución propuesta.

2.4.1 Formas Normales

Las formas normales son aplicadas a las tablas de una BD. Decir que una BD está en la forma normal N, es decir, que todas sus tablas están en la forma normal N.

En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F.Codd. (June 1970).

- **Primera forma normal (1 FN)**

Una tabla está en 1FN si sus atributos contienen valores atómicos es decir que en ninguna tabla existan atributos multivaluados o compuestos, de ser así el proceso sería, eliminar aquellas columnas que se repiten y ubicarlas en tablas completamente separadas. (Sicilia, 2008)

La BD para el Sistema de Videoconferencia cumple con esta forma normal, la misma contaba con una tabla que tenía atributos compuestos. Este es el caso de la tabla usuario donde el atributo nombre-completo puede tener nombre, apellido 1 y apellido 2. El mismo se divide en atributos simples representan atributos más básicos con su propio significado independiente.

- **Segunda forma normal (2 FN)**

Para que una relación esté en segunda forma normal (2FN) tiene necesariamente que estar en 1FN. La 2FN compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está en 2FN (16). (Sicilia, 2008)

La BD propuesta además de encontrarse en primera forma normal por lo explicado anteriormente, también se encuentra en segunda forma normal, porque en la misma no existen dependencias parciales, todos los atributos que no son clave de las tablas, o sea que no son ni claves primarias ni candidatas, dependen en su totalidad de la clave primaria de dicha tabla.

- **Tercera forma normal (3 FN)**

La tercera forma normal plantea que además de estar en 2 FN, tienen que haberse eliminado aquellas dependencias transitivas entre los atributos que no son clave. Entiéndase por transitiva aquella en la cual las columnas que no son clave son dependientes de otras columnas que tampoco son clave. (Sicilia, 2008)

Después de haberse eliminado los atributos multivaluados y compuestos y las dependencias parciales existentes en la BD, se comprobó que la misma estaba en segunda forma normal.

Capítulo 2: Diseño de la solución propuesta.

Después de este proceso, queda demostrado que la BD se encuentra normalizada hasta la tercera forma normal. Este último paso permitió que se eliminaran las redundancias de datos que aún quedaban de la segunda forma normal, evitando así soportar anomalías frente a las actualizaciones que se hicieran en la BD.

Como resultado de la normalización realizada se obtuvo una propuesta de diseño libre de anomalías en la actualización y mejora de la independencia de los datos, quedando en la 3FN y con la estructura siguiente:

usuario (usuario, nombre, apellido1, apellido2, correo, contraseña, #rol)

videoconferencia (id_vc, fecha, disponibilidad, #tipo)

registro (hora_inicio, hora_fin, ip, #usuario, #id_vc, #accion)

lista_contactos (id_lc, nombre)

vc_publicadas (link, id_vc, #modo_grabacion)

modo_grabacion (nombre)

tipo_videoconferencia (nombre)

rol (nombre)

usuario_lista_contactos (#usuario, #id_lc, owner)

usuario_videoconferencia (#usuario, #id_vc)

2.5 Diagrama Entidad Relación (DER) de la BD

El modelo físico de datos se diseñó a partir de un análisis de las clases persistentes, el MER y las realizaciones de los casos de uso del sistema. Este modelo fue realizado con la herramienta Visual Paradigm que genera de forma automática la BD en PostgreSQL. A continuación se muestra el modelo lógico.

Capítulo 2: Diseño de la solución propuesta.

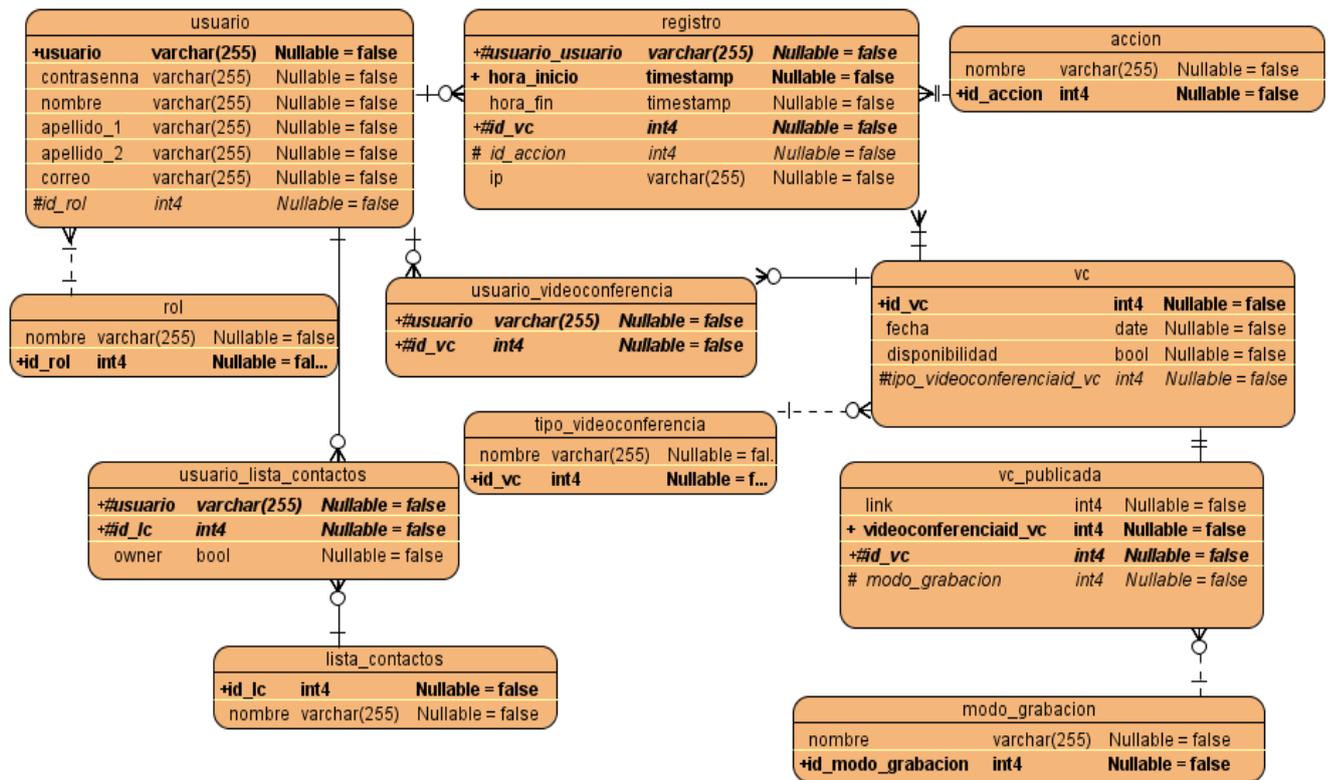


Ilustración 6: Diagrama Entidad Relación

2.6 Descripción de las tablas del diagrama entidad relación de la BD

Descripción detallada de la entidad “usuario”.

Nombre: usuario		
Descripción: Almacena los datos de los usuarios que solicitan una video conferencia.		
Atributos	Tipo	Descripción
+usuario	varchar(255)	Identificador de los usuarios.
contrasenna	varchar(255)	Permite la entrada del usuario a la BD.
nombre	varchar(255)	Nombre de los usuarios.
apellido1	varchar(255)	Primer apellido de los usuarios.
apellido2	varchar(255)	Segundo apellido de los usuarios

Capítulo 2: Diseño de la solución propuesta.

correo	varchar(255)	Dirección de correo de los usuarios.
#rol	varchar(255)	Rol que desempeña cada usuario.

Tabla 1: Descripción de la entidad usuario

Descripción detallada de la entidad “videoconferencia”.

Nombre: videoconferencia		
Descripción: Almacena las video conferencias con que cuenta el sistema.		
Atributos	Tipo	Descripción
+id_vc	int4	Identificador de cada videoconferencia.
fecha	date	Almacena los datos del momento de solicitud de la videoconferencia.
disponibilidad	bool	Si está disponible o no la videoconferencia.
#tipo_videoconferencia	varchar(255)	Tipos de videoconferencias (auditorio y sobremesa).

Tabla 2: Descripción de la entidad videoconferencia

Descripción detallada de la entidad “usuario_videoconferencia”.

Nombre: usuario_videoconferencia		
Descripción: Guarda la relación existente entre los usuarios y las videoconferencias.		
Atributos	Tipo	Descripción
+#usuario	varchar(255)	Identificador de los usuarios.
+# id_vc	int4	Identificador de cada videoconferencia.

Tabla 3: Descripción de la entidad usuario_videoconferencia

Descripción detallada de la entidad “registro”.

Nombre: registro		
Descripción: Almacena las acciones realizadas por los usuarios.		
Atributos	Tipo	Descripción

Capítulo 2: Diseño de la solución propuesta.

+#usuario_usuario	varchar(255)	Nombre del usuario responsable
+hora_inicio	timestamp	Hora en que el usuario inicia la videoconferencia.
hora_fin	timestamp	Hora en que el usuario cierra la videoconferencia.
#accion	varchar(255)	Acción que realiza el usuario.
# id_vc	int4	Identificador de cada videoconferencia.
ip	varchar	El ip de la pc donde se realizó la videoconferencia

Tabla 4: Descripción de la entidad registro

Descripción detallada de la entidad “usuario_lista_contactos”.

Nombre: usuario_lista_contactos		
Descripción: Almacena la lista de contactos de cada usuario.		
Atributos	Tipo	Descripción
+#usuario	varchar(255)	Identificador de los usuarios.
+#lista_contactos	int4	Lista de contactos de un usuario.
owner	bool	Si es dueño o no de esa lista de contacto.

Tabla 5: Descripción de la entidad usuario_lista_contactos

Descripción detallada de la entidad “vc_publicada”.

Nombre: vc_publicada		
Descripción: Almacena el link de la videoconferencia grabada o que se está transmitiendo.		
Atributos	Tipo	Descripción
+# id_vc	int4	Identificador de cada videoconferencia.
link	int4	Link de la videoconferencia grabada o en transmisión.

Capítulo 2: Diseño de la solución propuesta.

modo_grabacionnombre	varchar(255)	Almacena el tipo de grabación (un solo flujo o en varias pistas).
----------------------	--------------	---

Tabla 6: Descripción de la entidad vc_publicada

Descripción detallada de la entidad “accion”.

Nombre: accion		
Descripción: Almacena la accion realizada por el usuario.		
Atributos	Tipo	Descripción
nombre	varchar(255)	Nombre de la acción realizada por el usuario.
#Id_accion	int4	Identificador de cada acción

Tabla 7: Descripción de la entidad accion

Descripción detallada de la entidad “rol”.

Nombre: rol		
Descripción: Almacena el rol que ocupa cada usuario.		
Atributos	Tipo	Descripción
nombre	varchar(255)	Nombre del rol que ocupa el usuario (administrador, ponente y espectador).
#id_rol	int4	Identificador de cada rol

Tabla 8: Descripción de la entidad rol

Descripción detallada de la entidad “tipo_videoconferencia”.

Nombre: tipo_videoconferencia		
Descripción: Almacena los tipos de videoconferencias existentes()		
Atributos	Tipo	Descripción
nombre	varchar(255)	Nombre del tipo de videoconferencia.
#Id_tipo_vc	Int4	Identificador de cada tipo_videoconferencia.

Capítulo 2: Diseño de la solución propuesta.

Tabla 9: Descripción de la entidad tipo_videoconferencia

Descripción detallada de la entidad “lista_contacto”.

Nombre: lista_contacto		
Descripción: Almacena la lista de contactos de cada usuario.		
Atributos	Tipo	Descripción
#id_lc	int4	Identificador de cada usuario de la lista_contacto.
nombre	varchar(255)	Nombre de la lista de contactos.

Tabla 10: Descripción de la entidad lista_contacto

Descripción detallada de la entidad “modo_grabacion”.

Nombre: modo_grabacion		
Descripción: Almacena los tipos de grabación existentes.		
Atributos	Tipo	Descripción
nombre	varchar(255)	Nombre de los distintos tipos de grabación.(en el servidor, en varios flujos, un solo flujo)
#Id_modo_grabacion	Int4	Identificador de cada modo_grabacion.

Tabla 11: Descripción de la entidad modo_grabacion

2.7 Diagrama de componente

❖ **Componente:**

Es la parte modular de un sistema, desplegable, reemplazable que encapsula implementación y un conjunto de interfaces, para proporcionar la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos. Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. (Ingeniería, 2011)

❖ **Características de los componentes:**

- Tienen relaciones de traza con los elementos del modelo que implementan.

Capítulo 2: Diseño de la solución propuesta.

- Pueden implementar varios elementos, por ejemplo: varias clases; sin embargo la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando.

2.8 Diagramas de componentes para la implementación de la capa de acceso a datos.

Modelan la vista estática de un sistema. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia compilación y ejecución, pudiendo mostrar las interfaces que estos soporten. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, por lo cual cada diagrama describe un apartado del sistema. (Ingeniería, 2011)

Capítulo 2: Diseño de la solución propuesta.

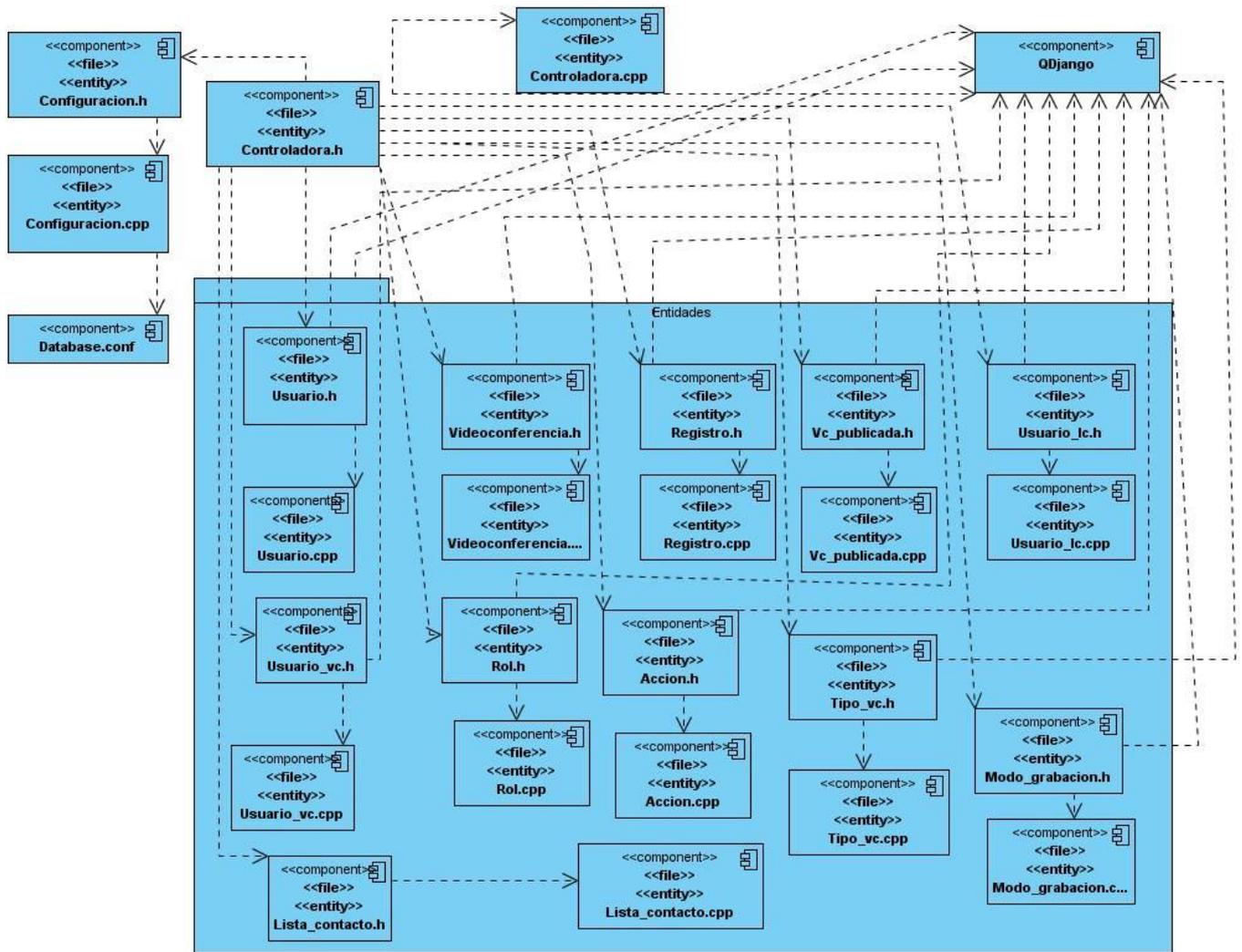


Ilustración 7: Diagrama de componentes

Capítulo 2: Diseño de la solución propuesta.

2.9 Conclusiones parciales

En este capítulo se presentó el diseño de la BD propuesta. El mismo fue realizado basándose en la ingeniería llevada a cabo por el analista del proyecto, rol que diseñó, entre otros artefactos, el diagrama de clases del diseño. Este sirvió de entrada para el diagrama de clases persistentes definido por un total de 5 clases. Se presentó además el modelo entidad relación para la BD, puntualizando un total de 11 entidades. Para facilitar el entendimiento de cada una de sus partes se mostró la descripción de las entidades con sus atributos, el tipo y una pequeña descripción de cada una de ellas.

Al culminar este capítulo se obtuvo como resultado entre otros, una BD, la cual se logró normalizar llevando a cabo todo el proceso para ello, extrayendo las dependencias funcionales y a partir de estas, se normalizó hasta llegar a la tercera forma normal, evitando de esta manera todos los problemas de redundancia e inconsistencia que podrían existir en la BD.

Capítulo 3: Validación del diseño realizado

Capítulo 3: Validación del diseño realizado

3.1 Introducción

En este capítulo se realiza la validación teórica del diseño de la BD del Sistema de Videoconferencia del proyecto productivo Factoría de Software de la Facultad 6, donde se miden elementos tales como: redundancia de información, integridad de datos y seguridad de la información. En el diseño propuesto se muestran elementos donde se centró el esfuerzo de los diseñadores; además de la validación funcional de la BD que se basa en la realización de pruebas de carga intensiva a la BD para comprobar su funcionamiento. El objetivo fundamental de este capítulo es validar el diseño realizado con aspectos teóricos y funcionales.

3.2 Validación de la teoría del diseño

Los aspectos que en este epígrafe se incluyen tienen que ver con la veracidad, consistencia y confiabilidad de la información y con la privacidad y confidencialidad de los datos. Las BD tienen dentro de sus características elementos que pueden ser utilizados para garantizar la calidad de la información almacenada y procesada. Dentro de los antes mencionados la seguridad de los datos representa lo más importante ya que garantiza el control de acceso a los datos por el personal autorizado y que estos no se dañen debido a acciones que no hayan sido controladas antes.

3.2.1 Análisis de la integridad de los datos

La integridad de los datos se refiere a la validez, corrección, consistencia y completitud de los datos almacenados en la BD. Tiene como función proteger la BD contra operaciones que introduzcan inconsistencias en los datos. Un control de integridad o restricciones es aquel que permite definir con precisión el rango de valores válidos para un elemento y/o las operaciones que serán consideraciones válidas en la relación de tales elementos. (Velazco, 2008)

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

Capítulo 3: Validación del diseño realizado

3.2.1.1 Categorías de integridad de los datos:

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina **restricciones de dominio**. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las BD relacionales y en todos sus estados o instancias, las reglas se deben cumplir todo el tiempo. Estas reglas son la de integridad de entidades y la de integridad referencial. Antes de definir las es preciso conocer los conceptos de **nulo** y **dominio**.

- **Nulo:** es un indicador que le dice al usuario que dato falta o no es aplicable. Por conveniencia, un dato que falta normalmente se dice que tiene valor nulo, pero el valor de nulo no es un valor de dato real. En vez de eso es una señal o un recordatorio de que el valor falta o es desconocido. (Velazco, 2008)
- **Dominio:** posibles valores que puede tener un campo. Un dominio no es más que un tipo de dato; posiblemente un tipo simple definido por el sistema o por el usuario. El dominio de un atributo define los valores posibles que puede tomar este atributo. Además de los dominios naturales, usados como tipos de datos. El administrador del sistema puede generar sus propios dominios definiendo el conjunto de valores permitidos. Esta característica usada en forma correcta, se convierte en mecanismo de control, restricción y validación de los datos a ingresar. (Velazco, 2008)

3.2.1.2 Reglas de integridad

❖ Integridad de entidad

Esta regla se aplica a las claves primarias de las relaciones base, ningún atributo que forme parte de una clave primaria puede aceptar valores nulos. Por definición, una clave primaria es irreducible y se utiliza para identificar de modo único los registros. Irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreductibilidad. Esta regla sólo se aplica a las relaciones base y a las claves primaria, no a las claves foráneas ni alternativas. (Marquez, 2001)

Este tipo de integridad se cumple en la BD propuesta ya que cada tabla tiene definida su clave primaria, la cual no está nula ni repetida. Es de vital importancia especificar la clave principal para cada

Capítulo 3: Validación del diseño realizado

registro, ya que esta representa la herramienta fundamental que utiliza el servidor de BD para seleccionar la información que se necesite, evitando así la duplicidad de los registros.

❖ Integridad de dominio

Una restricción de integridad de dominio es una regla que define valores válidos para los atributos de las diferentes tablas de una BD. Algunas tareas que aseguran la integridad de dominio son: definir los tipos de datos correctamente para cada campo y definir campos no nulos (NOT NULL) para evitar resultados inconsistentes. Para velar por la integridad de dominio en las BD, no se definieron atributos nulos y se precisaron correctamente cada uno de los tipos de datos para cada uno de los atributos. (Marquez, 2001)

❖ Integridad Referencial

La regla de Integridad referencial define que la BD no debe contener valores de claves foráneas sin concordancia. Esta regla se aplica a las claves foráneas. Si en una relación hay alguna clave foránea, entonces sus valores deben coincidir con los valores de la clave primaria a la que hace referencia, o bien, debe ser completamente nulo. La integridad referencial implica que en todo momento todos los datos involucrados en la BD sean correctos y sin repeticiones innecesarias. Todas las BD relacionales presentan esta propiedad pues el software gestor es responsable de su cumplimiento. (Marquez, 2001)

➤ La Integridad referencial también vigila que se cumplan las siguientes reglas:

- No se podrá introducir un valor en la tabla relacionada si antes no ha sido introducida en la tabla principal.
- No se puede eliminar un registro de una tabla principal si existen registros coincidentes en la tabla relacionada.
- No se puede cambiar un valor de la clave primaria en la tabla principal si el registro tiene registros relacionados.

Un ejemplo donde se puede ver en práctica este tipo de restricción es el que se muestra a continuación:

Los atributos **usuario** e **id_lc** de la tabla **usuario_lista_contactos** son claves foráneas, debido a que provienen y son claves primarias de las tablas **usuario** y **lista_contacto** respectivamente. Además estas claves cumplen con las restricciones siguientes:

Capítulo 3: Validación del diseño realizado

- a. Los atributos **usuario** e **id_lc** en la tabla **usuario_lista_contactos** tienen los mismos dominios que en sus tablas de origen.
- b. Al introducir las claves de las tablas **usuario** y **lista_contacto** en la tabla generada de una relación de mucho a mucho, en este caso **usuario_lista_contactos**, estos tienen que encontrarse en su entidad de origen, es decir, **usuario_lista_contactos** no puede tener un **usuario** e **id_lc** que no estén presentes en la tabla **usuario** y **lista_contacto** respectivamente.

3.3 Datos requeridos

Este término establece que una columna no tenga un valor nulo. Se define efectuando la declaración de una columna con la sentencia NOT NULL cuando la tabla que contiene las columnas se crea por primera vez, como parte de la sentencia CREATE TABLE.

En la tabla **usuario** se puede ver un ejemplo de lo antes mencionado, ya que los atributos pertenecientes a la misma no aceptan valores nulos. Esto no quiere decir que todas las columnas tengan que aceptar valores no nulos. Se puede dar el caso de que exista una columna con valor nulo, pero esto no se manifiesta en la BD. En la descripción de la tabla que se presenta a continuación quedó manifestada esta restricción.

```
-- Table: usuario
```

```
-- DROP TABLE usuario;
```

```
CREATE TABLE usuario
```

```
(  
  id_usuario serial NOT NULL,  
  contraseña character varying(255) NOT NULL,  
  nombre character varying(255) NOT NULL,  
  apellido_1 character varying(255) NOT NULL,  
  apellido_2 character varying(255) NOT NULL,  
  correo character varying(255) NOT NULL,
```

Capítulo 3: Validación del diseño realizado

```
id_rol integer NOT NULL,  
CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario),  
CONSTRAINT usuario_fk FOREIGN KEY (id_rol)  
REFERENCES rol (id_rol) MATCH SIMPLE  
ON UPDATE CASCADE ON DELETE RESTRICT  
)  
WITH (  
  OIDS=FALSE  
);
```

3.4 Análisis de la seguridad de los datos

Las BD son componentes esenciales de cualquier aplicación. Debido a que información considerablemente sensible o secreta puede ser almacenada en tales BD, se debe considerar seriamente la forma de protegerlas. La seguridad de los datos en las mismas es de suma importancia ya que se basa en garantizar la coherencia de los datos, demostrando que sólo los usuarios autorizados puedan efectuar las operaciones correctas sobre la BD. Esto se consigue mediante un control sobre los usuarios que acceden a la misma y los tipos de operaciones que están autorizados a realizar. Este control se llama gestión de autorizaciones, y permite crear o borrar usuarios y conceder o retirar derechos a efectuar determinados tipos de operaciones por ejemplo: crear y/o borrar objetos, modificar datos.

Actualmente el SGBD con que se va a trabajar en esta BD para el Sistema de Videoconferencia del proyecto productivo Factoría de Software de la Facultad 6, garantiza el control de acceso por parte de los usuarios a la BD de acuerdo a los privilegios que tienen asignados y los lugares donde ellos se encuentren ubicados.

El gestor utilizado para el desarrollo de la BD propuesta, trae por defecto mecanismos para restringir el acceso al sistema, tales como: la definición de roles y la configuración de los archivos `pg_hba.conf` y `postgresql.conf`, para establecer la conexión a la BD.

Para validar la autenticación del usuario que se vaya a conectar a la BD, es necesario que se controle mediante un archivo que trae PostgreSQL por defecto: **`pg_hba.conf`**. Este fichero se utiliza para definir

Capítulo 3: Validación del diseño realizado

los diferentes tipos de accesos que un usuario tiene en el clúster. Especifica cómo, dónde y desde qué sitio un usuario puede utilizar el clúster PostgreSQL. El formato de este fichero se interpreta de la siguiente manera: **Tipo de conexión** (El tipo de conexión mediante el cual se va a conectar el usuario), **Nombre de la BD** (El nombre de la BD a la que se tendrá acceso), **Usuario** (El usuario definido por el administrador para acceder a la BD), **IP de conexión del cliente** (dirección IP que se va a conectar a la BD) y **Método de Autenticación** (Método de conexión a la BD). Excepto aquellas líneas restantes incluidas en el fichero que comiencen con el carácter # que son interpretadas como comentarios.

3.5 Validación funcional y tipos de pruebas

3.5.1 Herramienta para un llenado voluminoso y eficiente de la BD

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto de software y representa una revisión final de las especificaciones del diseño y de la codificación. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa.

Para la realización de pruebas de un llenado voluminoso de la BD, se empleó la herramienta EMS PostgreSQL Data Generator 2008 es una herramienta impresionante para generar datos de prueba a las tablas de BD con PostgreSQL, brinda la posibilidad de guardar y editar secuencias de comandos.

Esta herramienta permite rellenar varias **tablas** de BD con datos de prueba al mismo tiempo, define tablas y campos para generar datos, configura valores de rangos, crea listas de valores de forma manual con selección de consultas SQL, establece los parámetros para cada tipo de campo. El empleo de esta herramienta posibilitó chequear la integridad de los datos.

Capítulo 3: Validación del diseño realizado

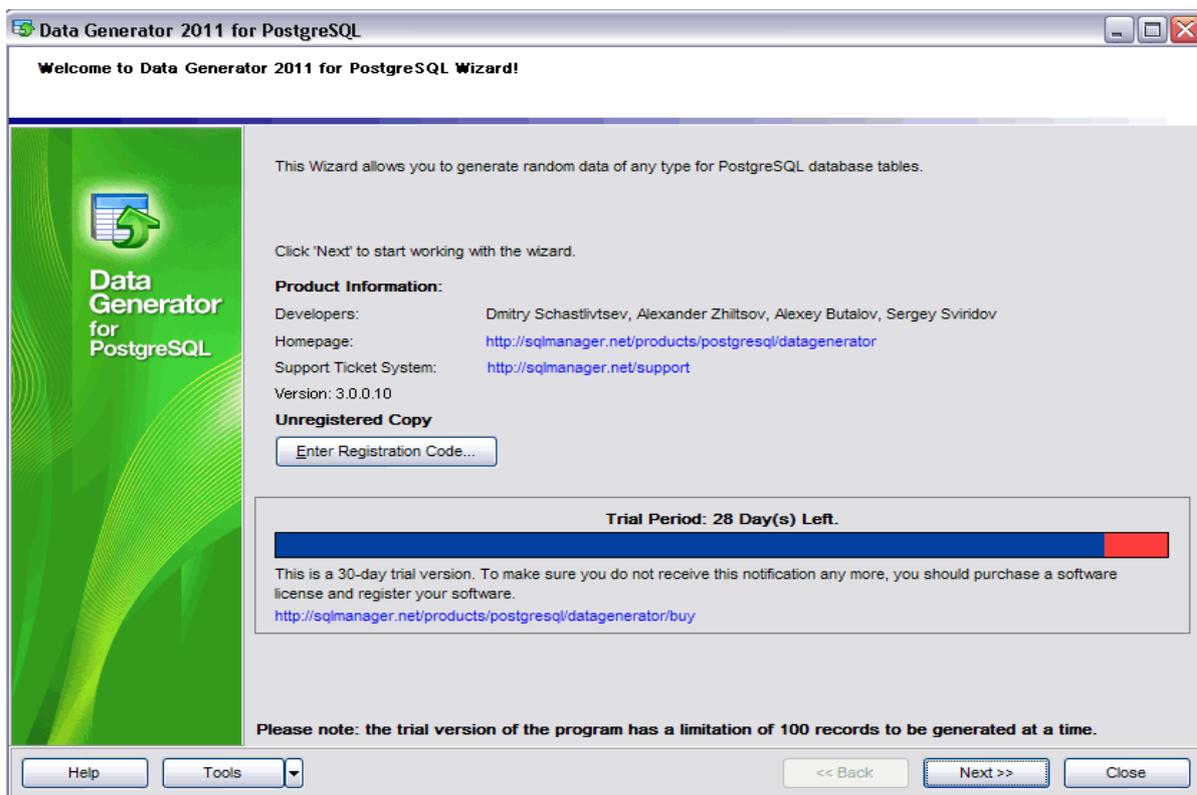


Ilustración 8: Herramienta para la prueba de volumen de la BD

Capítulo 3: Validación del diseño realizado

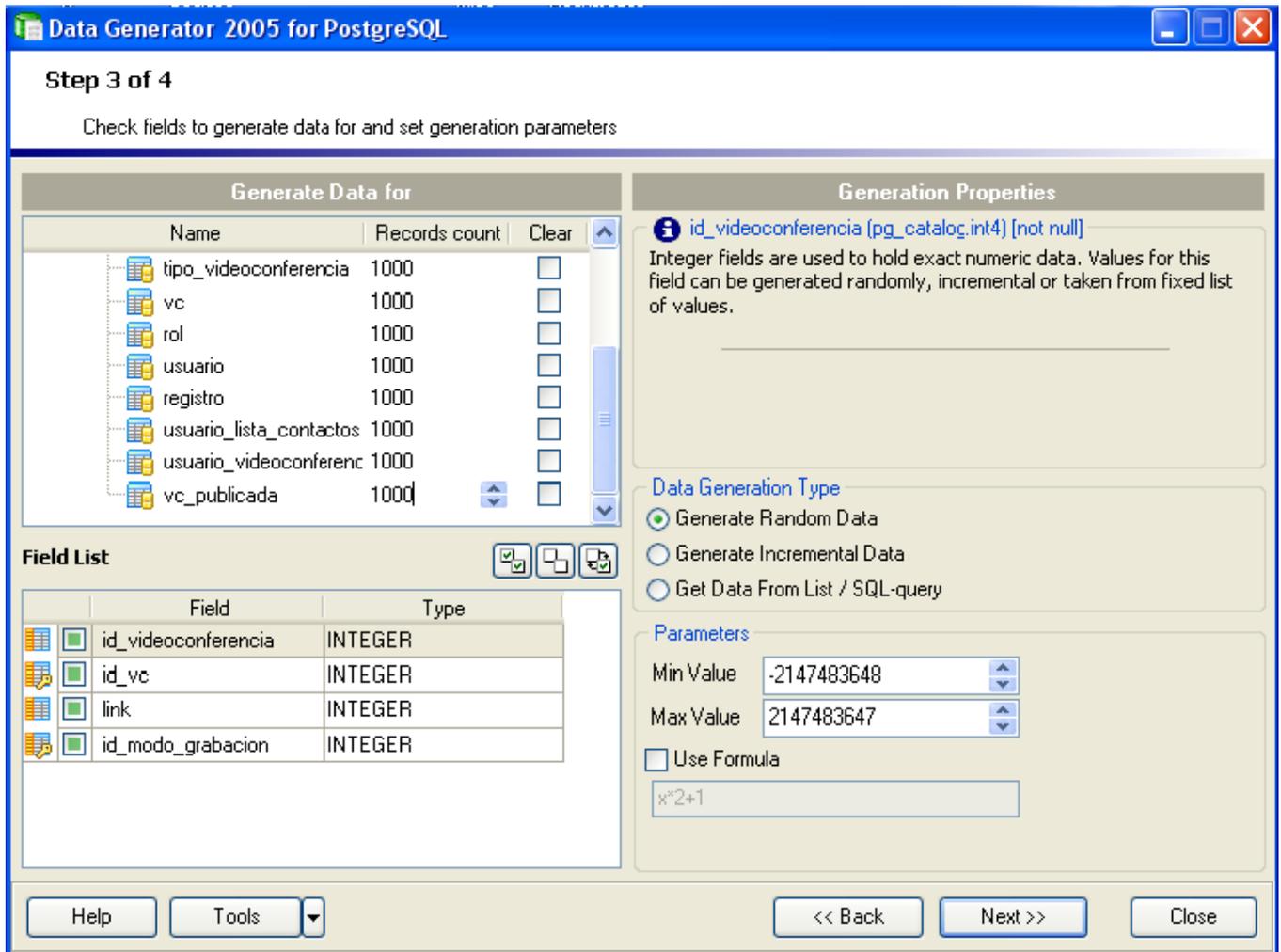


Ilustración 9: Ejemplo de la realización de la prueba de volumen

Ilustración 10: Eficiencia del llenado de la BD

Para la realización de la prueba de volumen a la BD se eligió esta herramienta la misma permitió introducirles hasta 1000 filas a cada tabla sin ningún error, con este procedimiento se pretende examinar el funcionamiento del sistema cuando está trabajando con grandes volúmenes de datos, simulando las cargas de trabajo esperadas. Nótese que la información generada por la herramienta no tiene ningún valor desde el punto de vista del usuario por su propio carácter aleatorio.

Al introducir los datos no se presentaron problemas de límite de capacidad, ni de volumen de datos. No se detectaron desbordamientos de matrices, columnas, atributos, tipos de datos, ni peticiones excesivas de memoria. No se detectaron problemas con los tipos de datos definidos en el diseño de la

Capítulo 3: Validación del diseño realizado

BD. Lo anteriormente planteado garantiza que el gestor que se utilizó y el diseño de las estructuras de la BD soportan completamente el almacenamiento de los niveles de información requeridos.

3.5.2 Herramienta para la administración de BD

Para una buena administración de la BD del sistema se utilizó la herramienta PgAdminIII está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar BD complejas. La interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL. Además está diseñada para ejecutarse en la mayoría de los Sistemas Operativos.

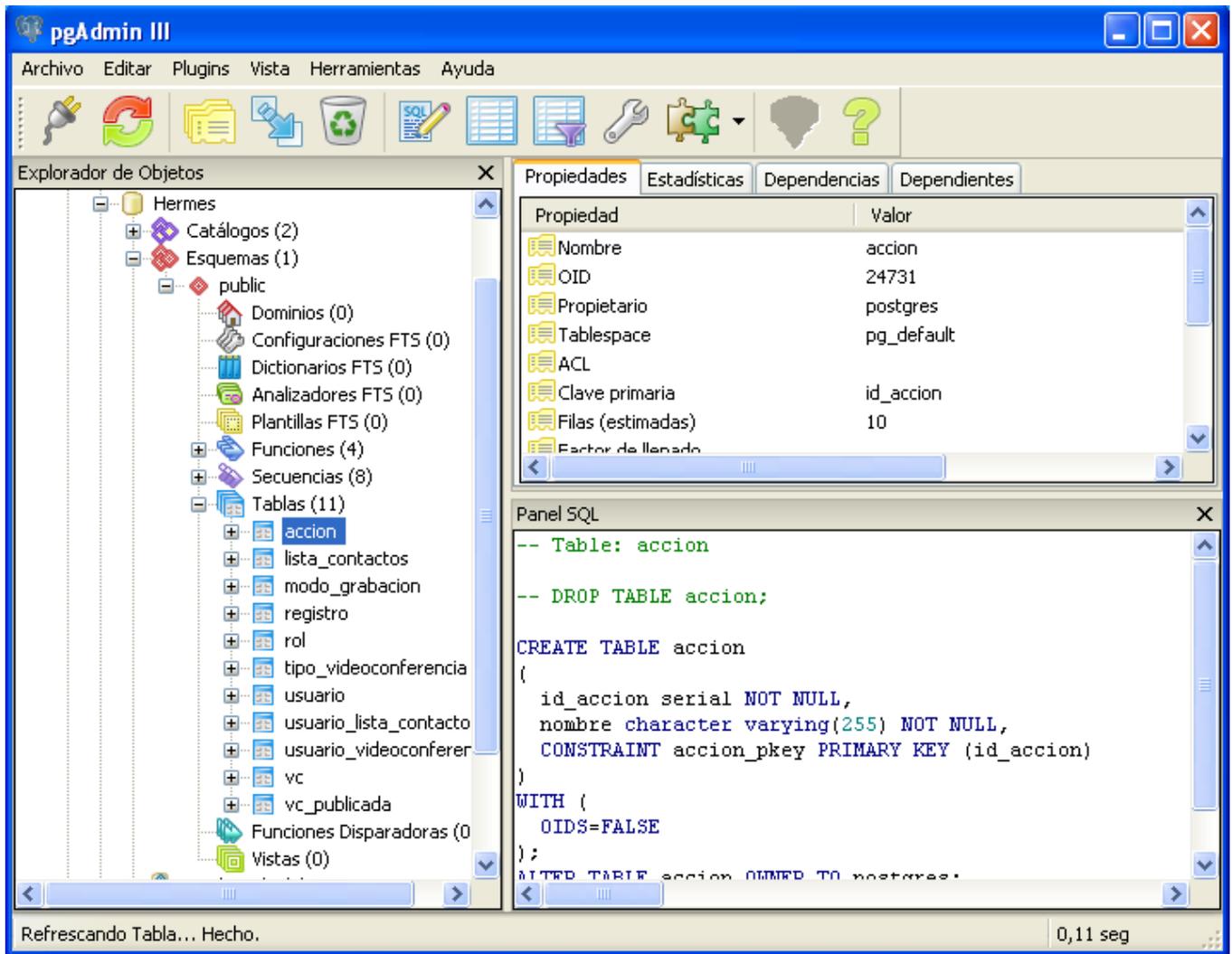


Ilustración 11: Herramienta de administración de BD

Capítulo 3: Validación del diseño realizado

Esta herramienta se utilizó para la administración de la BD, la realización de consultas y funcionalidades de la misma. Además brinda gran seguridad a la hora del trabajo dentro de la BD ya que se pueden rectificar errores que se puedan cometer en el llenado de los datos de cada una de las tablas.

3.6 Pruebas de software

Son un conjunto de herramientas, técnicas y métodos que evalúan el desempeño de un programa. Constituyen una etapa imprescindible durante el proceso de desarrollo del software. Las mismas son el único instrumento adecuado para determinar el status de la calidad de un producto de software. Además permiten detectar y corregir el máximo de errores posibles antes de la entrega al cliente del software desarrollado. Ya que en este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Por lo que el éxito de las mismas puede mejorar la percepción de calidad del usuario final y lograr su satisfacción.

El objetivo principal de las pruebas es asegurar que el software cumpla con las especificaciones requeridas y eliminar los posibles defectos que éste pudiera tener. Es importante considerar que las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección. Las pruebas, por su gran importancia se llevan a cabo durante todo el ciclo de vida del producto, su mayor punto de desarrollo se encuentra en la etapa de implementación.

Entre sus objetivos se encuentran:

- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Reducir costos de mantenimiento.
- Obtener información concreta acerca de fallas, que pueda usarse como apoyo en la mejora de procesos, y en la de los desarrolladores.

Si se producen modificaciones en el programa, habrá que probar de nuevo todas las partes afectadas por las modificaciones.

Dentro de las pruebas se destacan principalmente dos tipos de pruebas:

- Caja Negra.

Capítulo 3: Validación del diseño realizado

- Caja Blanca.

3.6.1 Pruebas de caja negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software. (Sobre la disciplina prueba, 2010-2011)

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las BD externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

❖ Técnicas de prueba de Caja Negra.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- **Técnica de la Partición de Equivalencia:** divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- **Técnica del Análisis de Valores Límites:** prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- **Técnica de Grafos de Causa-Efecto:** permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

3.6.2 Pruebas de caja blanca

Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o el código. Mediante los métodos de prueba de la caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que: (Sobre la disciplina prueba, 2010-2011)

Capítulo 3: Validación del diseño realizado

- Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
- Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Se ejerciten las estructuras internas de datos para asegurar su validez.

❖ **Métodos de Prueba de la caja blanca**

- **La prueba del camino básico:** permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

Los pasos que se siguen para aplicar esta técnica son:

- A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
 - Se calcula la complejidad ciclomática del grafo.
 - Se determina un conjunto básico de caminos independientes.
 - Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.
- **La prueba de condición:** es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
 - **La prueba de flujo de datos:** se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
 - **La prueba de bucles:** es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.

❖ **Prueba de caja blanca aplicada**

Como prueba de caja blanca aplicada se realizó la prueba del camino básico la misma garantiza que durante la prueba en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez. Para aplicar la técnica del camino básico se deben realizar los siguientes pasos.

Los pasos que se siguen para aplicar esta técnica son:

Capítulo 3: Validación del diseño realizado

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado, cada nodo del grafo corresponde a una o más sentencias de código.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Un grafo de flujo está formado por 3 componentes fundamentales que ayudan a su elaboración y comprensión, estos brindan información para confirmar que el trabajo se está haciendo adecuadamente. Componentes del grafo de flujo:

- **Nodo:** son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.
- **Aristas:** son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.
- **Regiones:** son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento. (López, 2009)

Para realizar la prueba del camino básico es preciso calcular la complejidad ciclomática del algoritmo a analizar. A continuación se enumera las sentencias de código del procedimiento realizado sobre el método buscar usuario:

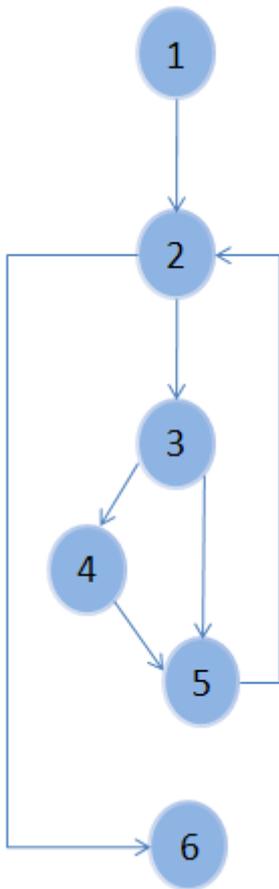
//Buscar los usuarios cuyo nombre sea Yudirenia

```
QDjangoQuerySet<Usuario> users; 1
QDjangoQuerySet<Usuario> user_encontrados; 1
user_encontrados=users.filter (QDjangoWhere("nombre",QDjangoWhere::Equals,"Yudirenia")); 1
    Usuario user; 1
    for (int i = 0; i < user_encontrados.size(); ++i) 2
    {
```

Capítulo 3: Validación del diseño realizado

```
if (user_encontrados.at(i, &user)) 3
{
    qDebug() << "Usuarios encontrados: " << user.GetName(); 4
}5
}6
```

- Grafo de flujo asociado al código anterior:



- Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. (López, 2009)

Capítulo 3: Validación del diseño realizado

$$V(G) = (A - N) + 2$$

$$V(G) = (7 - 6) + 2$$

$$V(G) = 3$$

Siendo —“A” la cantidad total de aristas y —“N” la cantidad total de nodos. Se puede usar también:

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$

Siendo —“P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = R \quad V(G) = 3$$

Siendo —“R” la cantidad total de regiones, para cada fórmula —V(G) representa el valor del cálculo.

Luego de haber realizado el cálculo por las 3 vías planteadas se llega a la conclusión que el algoritmo presentado anteriormente tiene una complejidad ciclomática de 3 que lo que significa que existen tres posibles caminos lógicos por donde recorrer el algoritmo.

1. 1 - 2 - 3 - 4 - 5 - 2 - 6
2. 1 - 2 - 3 - 5 - 2 - 6
3. 1 - 2 - 6

Una vez extraído los caminos básicos se deben realizar los casos de pruebas para este procedimiento.

Para realizarlos es necesario cumplir con las siguientes exigencias:

- ✓ Descripción: se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.
- ✓ Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- ✓ Entrada: se muestran los parámetros que entran al procedimiento.
- ✓ Resultados Esperados: se expone resultado que se espera que devuelva el procedimiento.

Capítulo 3: Validación del diseño realizado

Luego de tener elaborados los Grafos de Flujos y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Se escogen los datos de forma que las condiciones de los nodos predicados estén adecuadamente establecidas, con el fin de comprobar cada camino. Pero como cada caso de prueba para cada camino básico es el mismo se escoge el camino 1 como ejemplo donde se obtuvo como resultado, que cada una de las sentencias del programa se han ejecutado por lo menos una vez y no existió error alguno.

- **Caso de prueba para el camino básico 1**

4. Camino 1: 1- 2 - 3 - 4 - 5 - 2 - 6

Caso de prueba: Buscar usuario cuyo nombre sea Yudirenia.

Parámetros de entrada: ----

Valor de Usuarios encontrados: por definir

Resultados:

Se comprueba que la instrucción "For" se cumpla y que se puedan obtener los datos de la tabla usuario.

3.7 Conclusiones parciales

En este capítulo se analizaron aspectos a tener en cuenta para realizar el diseño de la BD, como son: la integridad de los datos y la redundancia de los mismos.

- Se analizó la seguridad de la BD, realizando un control de acceso y limitando los permisos de los usuarios.
- Se realizó un llenado voluminoso de la BD con la herramienta EMS Data Generator For PostgreSQL 2008 para la realización de pruebas funcionales y se probaron algunas consultas SQL obteniéndose tiempos de respuestas adecuados.
- Se realizó la prueba de caja blanca al software obteniéndose el valor complejidad del mismo.

Conclusiones generales

3.8 Conclusiones generales

Con el presente trabajo de diploma se provee al proyecto productivo Factoría de Software de la Facultad 6 de la UCI, de una BD en la cual se encuentra de forma centralizada todos los datos referentes al nuevo Sistema de Videoconferencia, lo que va a facilitar el desarrollo de futuras aplicaciones para este centro.

- ❖ Se hizo un análisis del funcionamiento de la BD, donde la misma arrojó resultados satisfactorios a las pruebas realizadas. El presente trabajo ha dado como resultado una BD el cual se logró mediante el cumplimiento de las tareas propuestas que se dividieron en:
 - Se cumplió el objetivo general planteado a través del desarrollo de una BD que fue normalizada hasta la Tercera Forma Normal (3FN), lo que hizo posible que los datos pertenecientes a las tablas no fueran redundantes ni inconsistentes.
 - También se realizó la implementación de la capa de acceso a datos con la utilización del ORM Qdjango garantiza una rápida conexión a la BD.
 - La realización del modelo lógico de la BD, posteriormente el diseño físico de la misma y por último la validación de la solución propuesta.

La BD diseñada está en condiciones de ser puesta en práctica para darle accesibilidad a la información existente en el Sistema de Videoconferencia del proyecto productivo Factoría de Software de la Facultad 6.

Recomendaciones

Recomendaciones

Después de haber desarrollado la BD concernientes al Sistema de Videoconferencia del proyecto productivos de la Facultad 6 en la presente investigación se recomienda:

- El refinamiento del modelo lógico propuesto en próximas iteraciones del desarrollo del software.
- Actualización periódica de la BD, logrando que se mantenga la integridad de los datos.
- El empleo del gestor de BD PostgreSQL para almacenar y gestionar la información en otros sistemas, como vía de lograr la soberanía tecnológica del país.

Referencias bibliográficas

Referencias bibliográfica

- Aguilar, David. 2011.** [Online] 2011. <http://www.slideshare.net/DavidPaLMundo/base-de-datos-6448796>.
- Alonso, Evelyn Menéndez. 2009.** Herramientas CASE para el proceso de desarrollo de Software. [Online] 2009. <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
- Angie Pao 29. 2010.** [Online] 2010. http://www.gratisblog.com/all_the_drama/i147406-base_de_datos_iv._modelos_de_datos.htm.
- Arbildo, Yerson Arce. 2010.** Todo sobre sistemas.. [Online] 2010. <http://isystemss.blogspot.com/2010/09/que-herramientas-de-analisis-y.html>.
- Barrios, Miguel. 2009.** Informática 1:Videoconferencia. [Online] 2009. <http://miguelbarrios-informatica.blogspot.com/2009/04/videoconferencia.html>.
- Codd, Edgar Frank. 2009.** Base de Datos: Modelo de datos. [Online] 2009. <http://www.imgeek.net/?p=542>.
- Cordero, Mariela G. López. 2008.** Videoconferencia. [Online] 2008. <http://marielaglopez.blogspot.com/>.
- Date. 2001.** Introduccion a las BD. 2001.
- 2010.** Encapsular campo. [Online] 2010. <http://msdn.microsoft.com/es-es/library/a5adyhe9%28VS.80%29.aspx>.
- Fernández-Baillo, David Carrero. 1995-2010.** Diccionario Informático. [Online] 1995-2010. <http://www.glosarium.com/term/1053,14,xhtml>.
- González-Aponcio, Zenona. 2010.** Videoconferencia - Presentation Transcript. [Online] 2010. <http://www.slideshare.net/ansaca/videoconferencia>.
- Hidalgo, Pachuca. 2009.** Sistema de Universidad Virtual. [Online] 2009. <http://virtual.uaeh.edu.mx/riv/videoconferencia.php>.
- Jahzeel. CUADRO COMPARATIVO DE BASES DE DATOS. [Online]** <http://jahzeel.espacioblog.com/post/2007/02/12/uadro-comparativo-bases-datos>.
- Jeff. 2008.** Sistema de Información. [Online] 2008. <http://es.kioskea.net/contents/systeme-d-information/si-systeme-d-information.php3>.
- Laudon. 2006.** Sistema de Información. [Online] 2006. <http://www.edicionsupc.es/ftppublic/pdfmostra/OE04301M.pdf>.

Referencias bibliográficas

- Lisy.** 2010. MySQL. [Online] 2010.
http://www.miportal.edu.sv/portal/comunidad/blogs/lisy_xoxo/archive/2010/08/16/my-sql.aspx.
- Marrero, Lic.Belinia Capote.** 2003. [Online] 2003.
http://bvs.sld.cu/revistas/aci/vol11_2_03/aci030203.htm.
- 1999-2010.** Masadelante - Definición de plug-in. [Online] 1999-2010.
<http://www.masadelante.com/faqs/plug-in>.
- Mato, Rosa María.** 1999. Base de Datos. 1999.
- Mendez, Luis E.** 2009. [Online] 2009. http://www.slideshare.net/luis_mm/dim-dim-presentation-1164251.
- Menendez, Rosa.** Lo nuevo de Rational Rose. [Online]
<http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>.
Modalidades. [Online] <http://www2.udec.cl/~pacortes/tarea2.htm#introd>.
Modelo de Base de Datos. [Online] <http://www.scribd.com/doc/17170125/Modelos-de-Bases-de-Datos>.
- 2010.** Oracle. [Online] 2010. <http://www.ecured.cu/index.php/Oracle>.
- Peco, Daniel.** PostgreSQL. [Online] http://danielpecos.com/docs/mysql_postgres/x15.html.
- Pino, Maria.** 2004. Base de Datos. [Online] 2004. http://www.mariapinto.es/e-coms/bases_datos.htm.
- 2009.** Programación Orientada a Objetos. [Online] 2009. <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
- RECOL.** 2007. [Online] 2007.
http://www.recol.es/index.php?option=com_content&task=view&id=110&Itemid=417.
- Rozic, Sergio Ezequiel.** 2004. Base de Datos. 2004.
- Sánchez, Ing. Fernando Luque.** 2005. Aplicaciones en N-Capas en Visual Basic .Net. [Online] 2005.
http://www.elguille.info/colabora/NET2005/FernandoLuque_NCapas.htm.
- Schmieg, Blix by Sebastian.** 2008. Fundamentación Informáticos. [Online] 2008.
<http://fundamentosinformaticosjl.wordpress.com/category/base-de-datos/>.
- Server, SharePoint.** 2007. [Online] 2007. <http://jmhogua.blogspot.com/2007/01/programacin-en-capas-primera-parte-capa.html>.
- 2010.** Softhoy. [Online] 2010. <http://www.softoy.com/software/bigbluebutton-manten-conferencias-linea-esta-aplicacion-open-source.html>.
Tutorial Base de Datos. [Online] http://sistemas.itlp.edu.mx/tutoriales/basedat1/tema1_4.htm.
- Valle, Jose.** 2007. Herramientas CASE para BD. [Online] 2007.
<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>.
- WP, Daily.** 2008. Definición de modelo de datos. [Online] 2008. <http://definicion.de/modelo-de-datos/>.

Referencias bibliográficas

- BD, C. d. (2005).** MODELO ENTIDAD RELACIÓN (MER) conf 2., (p. 1). <http://eva.uci.cu/mod/resource/view.php?id=31736>.
- June. (1970).** A Relational Model of Data for Large Shared Data Banks Communications of the ACM.
- profesores, C. d. (2010).** Fase de Inicio. Disciplina de Modelamiento del negocio., (p. 4). <http://eva.uci.cu/mod/resource/view.php?id=21011>.
- Sicilia, M.-A. (2008).** *normalización de bases de datos relacionales* . Retrieved from <http://cnx.org/content/m18350/latest/>
- Jonas, L. P. (2009).** Trazabilidad en la Producción. Retrieved from <http://www.aie.cl/files/file/comites/ca/articulos/junio-06.pdf>
- Maximiliano González, Javier Madrid, Anabela Torrealba.** PATRONES DE DISEÑO Singleton. [Online] Microsoft Powerpoint.
- Welicki, León. 2011.** Patrón Singleton. [Online] 2011. <http://msdn.microsoft.com/es-es/library/bb972272.aspx>.
- ¿Qué es Oracle?** [Online] / auth. Corporation Oracle. - 2009. - <http://www.mrlocooracle.com/ora.html>.
- Conozca más acerca de PostgreSQL aquí:.** [Online] / auth. Faraon El // Conozca más acerca de PostgreSQL aquí:... - 2011. - <http://postgresql.uci.cu/foro/viewtopic.php?f=24&t=81>.
- Fundamentos de Ingeniería de Software** [Book] / auth. Marcello Visconti Hernán Astudillo. - <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-DisenoOO.pdf> : [s.n.], 2008.
- Grupo de Investigación Aplicada para Estudiantes de Ing. de Sistemas de la UNEFA** [Online] / auth. CORREA PATRICIA ALEJANDRA // Grupo de Investigación Aplicada para Estudiantes de Ing. de Sistemas de la UNEFA. - 2010. - <http://ing-de-sistemas-unefa.lacoctelera.net/>.
- Historia de la videoconferencia** [Online] // Modalidades. - <http://www2.udec.cl/~pacortes/tarea2.htm>.
- Procesos de Negocios** [Online] / auth. Emily Valenzuela Ortiz Susony, Lourdes Moscoso, Victor Rodríguez. - 2010. - <http://grupo5.comuf.com/?p=100>.
- Sistemas de información gerencial: administración de la empresa digital** [Book] / auth. Laudon Jane P, Kenneth C, Laudon. - [s.l.] : books.google.com/books?isbn=9702605288....., 2004.
- Bienvenido a QDjango** [Online] / auth. Liané Jeremy. - 2003-2009. - <http://opensource.bolloretelecom.eu/projects/qdjango/wiki>.

Referencias bibliográficas

Conceptos básicos de ORM (Object Relational Mapping) [Online] / auth. Carrero. - 1998-2011. - http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349.

Desarrollo del Componente Trabajador del Subsistema Capital Humano . 2009. [Book Section] / auth. López. - 2009.

Bibliografía consultada

Angie Pao 29. [En línea] 2010. http://www.gratisblog.com/all_the_drama/i147406-base_de_datos_iv._modelos_de_datos..htm.

Arbildo, Yerson Arce. Todo sobre sistemas.. [En línea] 2010. <http://isystemss.blogspot.com/2010/09/que-herramientas-de-analisis-y.html>.

Bienvenido a QDjango [Online] / auth. Liané Jeremy. - 2003-2009. - <http://opensource.bolloretelecom.eu/projects/qdjango/wiki>.

BD, C. d. (2005). MODELO ENTIDAD RELACIÓN (MER) conf 2., (p. 1). <http://eva.uci.cu/mod/resource/view.php?id=31736>.

Conceptos básicos de ORM (Object Relational Mapping) [Online] / auth. Carrero. - 1998-2011. - http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349.

Conozca más acerca de PostgreSQL aquí:. [Online] / auth. Faraon El // Conozca más acerca de PostgreSQL aquí:... - 2011. - <http://postgresql.uci.cu/foro/viewtopic.php?f=24&t=81>.

Date. Introduccion a las BD. 2001.

Encapsular campo. [En línea] 2010. <http://msdn.microsoft.com/es-es/library/a5adyhe9%28VS.80%29.aspx>.

Desarrollo del Componente Trabajador del Subsistema Capital Humano . 2009. [Book Section] / auth. López. - 2009.

Fernández-Baillo, David Carrero. Diccionario Informático. [En línea] 1995-2010. <http://www.glosarium.com/term/1053,14,xhtml>.

Fundamentos de Ingeniería de Software [Book] / auth. Marcello Visconti Hernán Astudillo. - <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-DisenoOO.pdf> : [s.n.], 2008.

González-Aponcio, Zenona. Videoconferencia - Presentation Transcript. [En línea] 2010. <http://www.slideshare.net/ansaca/videoconferencia>.

Grupo de Investigación Aplicada para Estudiantes de Ing. de Sistemas de la UNEFA [Online] / auth. CORREA PATRICIA ALEJANDRA // Grupo de Investigación Aplicada para Estudiantes de Ing. de Sistemas de la UNEFA. - 2010. - <http://ing-de-sistemas-unefa.lacoctelera.net/>.

Hidalgo, Pachuca. Sistema de Universidad Virtual. [En línea] 2009. <http://virtual.uaeh.edu.mx/riv/videoconferencia.php>.

Historia de la videoconferencia [Online] // Modalidades. - <http://www2.udec.cl/~pacortes/tarea2.htm>.

Bibliografía

- Jeff.** Sistema de Información. [En línea] 2008. <http://es.kioskea.net/contents/systeme-d-information/si-systeme-d-information.php3>.
- Jahzeel.** CUADRO COMPARATIVO DE BASES DE DATOS. [En línea] <http://jahzeel.espacioblog.com/post/2007/02/12/uadro-comparativo-bases-datos>.
- Laudon.** Sistema de Información. [En línea] 2006. <http://www.edicionsupc.es/ftppublic/pdfmostra/OE04301M.pdf>.
- Lisy.** MySQL. [En línea] 2010. http://www.miportal.edu.sv/portal/comunidad/blogs/lisy_xoxo/archive/2010/08/16/my-sql.aspx.
- Mato, Rosa María.** Base de Datos. 1999. Modelo de Base de Datos. [En línea] <http://www.scribd.com/doc/17170125/Modelos-de-Bases-de-Datos>.
- Masadelante - Definición de plug-in. [En línea] 1999-2010. <http://www.masadelante.com/faqs/plug-in>.
- Marrero, Lic. Belinia Capote.** [En línea] 2003. http://bvs.sld.cu/revistas/aci/vol11_2_03/aci030203.htm. 16. Modalidades. [En línea] <http://www2.udec.cl/~pacortes/tarea2.htm#introd>.
- Mendez, Luis E.** [En línea] 2009. http://www.slideshare.net/luis_mm/dim-dim-presentation-1164251. Programación en Capas Primera Parte (Capa de Acceso a Datos). [En línea] 2010. <http://jmhogua.blogspot.com/2007/01/programacin-en-capas-primera-parte-cap.html>.
- Pino, Maria.** Base de Datos. [En línea] 2004. http://www.mariapinto.es/e-coms/bases_datos.htm.
- Rozic, Sergio Ezequiel.** Base de Datos. 2004.
- RECOL.** [En línea] 2007. http://www.recol.es/index.php?option=com_content&task=view&id=110&Itemid=417.
- Abraham Silberschatz, F. K., Henry y S Sudarshan** (2002). Fundamentos de base de datos. Cuarta Edición. Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
- Sánchez, Ing. Fernando Luque.** Aplicaciones en N-Capas en Visual Basic .Net. [En línea] 2005. http://www.elguille.info/colabora/NET2005/FernandoLuque_NCapas.htm.
- Schmieg, Blix by Sebastian.** Fundamentación Informáticos. [En línea] 2008. <http://fundamentosinformaticosjl.wordpress.com/category/base-de-datos/>.
- Soft hoy. [En línea] 2010. <http://www.soft hoy.com/software/bigbluebutton-manten-conferencias-linea-esta-aplicacion-open-source.html>.
- WP, Daily.** Definición de modelo de datos. [En línea] 2008. <http://definicion.de/modelo-de-datos/>.
- Welicki, León.** 2011. Patrón Singleton. [Online] 2011. <http://msdn.microsoft.com/es-es/library/bb972272.aspx>.
- June. (1970).** *A Relational Model of Data for Large Shared Data Banks Communications of the ACM.*

Bibliografía

profesores, C. d. (2010). Fase de Inicio. Disciplina de Modelamiento del negocio., (p. 4). <http://eva.uci.cu/mod/resource/view.php?id=21011>.

Sicilia, M.-A. (2008). *normalización de bases de datos relacionales* . Retrieved from <http://cnx.org/content/m18350/latest/>

[Online] / auth. Kolpackou Boris. - 2011. - <http://www.codesynthesis.com/products/odb/>.

[Online] / auth. Lau Ben. - 2011. - <http://code.google.com/p/d-quest/> .

Conceptos básicos de ORM (Object Relational Mapping) [Online] / auth. Carrero. - 1998-2011. - http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349.

QxOrm [Online] / auth. apps Equipo de Qt. - 2010. - <http://qt-apps.org/content/show.php/QxOrm?content=124240>.

Glosario de términos

- **Plugin**

Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. (1999-2010)

- **Programación Orientada a Objetos**

Como su mismo nombre indica, la programación orientada a objetos se basa en la idea de un objeto, que es una combinación de variables locales y procedimientos llamados métodos que juntos conforman una entidad de programación.

- **Multiplexar**

Técnica que permite transmitir diferentes comunicaciones a través de un único canal.

- **Encapsular**

La operación de refactorización Encapsular campo permite crear rápidamente una propiedad a partir de un campo existente y, a continuación, actualizar el código de forma transparente con referencias a la nueva propiedad.

- **Entidad**

Es un objeto definido dentro del universo o realidad observada. Son ejemplos de entidades: persona, auto, profesor, alumno, sistema y escuela.

- **Herramienta CASE**

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. CASE es también definido como el Conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (Valle, 2007).

- **Códec**

Es la abreviatura de codificador-decodificador. Describe una especificación desarrollada en software, hardware o una combinación de ambos