Universidad de las Ciencias Informáticas

Facultad 3

Título: Análisis del Sistema de Información y diseño de una base de datos documental para la Unión Nacional de Juristas de Cuba

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autores: ***Sulivan Lominchar de Varona***

Yandy Peñate Peña

Tutor: ***Yiset Pérez Rizo***

Junio del 2011

“La mayoría de la gente no sabe lo que quiere hasta que se lo enseñas”

Steve Jobs

DECLARACION DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____ _____

Sulivan Lominchar de Varona — Autor

Yandy Peñate Peña — Autor

_____

Yiseth Pérez Rizo

Tutor

DATOS DE CONTACTO

Sulivan Lominchar de Varona:

Dirección: Calle Quinta, entre Tercera y Avenida Figueredo, Edificio 28, Apartamento 11, Jesús Menéndez, Bayamo, Granma.

Teléfono: 424310

Yandy Peñate Peña:

Dirección: Calle Independencia, Numero Ochenta y ocho, Entre Domingo Mujica y Sánchez Figuera, Jagüey Grande, Matanzas.

AGRADECIMIENTOS

Sulivan Lominchar de Varona:

Quiero agradecerle a mi familia especialmente a mi mamá por haberme apoyado y aguantado durante todos estos años, a mi novia, parte importante de esta tesis, a mis amigos y compañeros que ya quiero como hermanos, en especial Sysley (si no fuera por ti no me hubiera graduado), a Alexander (me diste la esperanza que necesitaba), Héctor y JJ (las dos patas restantes del trípode), Tito y Driggs (mis eterno compañeros del doble), Liván (te mereces mil agradecimientos); a todos los que de una forma u otra aportaron su grano de arena en esta tesis, a kiki que me evitó años de estudio, mis compañeros de cuarto: Manuel, Roberto, Rodolfo, Ana y Caro (gracias por soportar mis perretas), también deseo agradecer a todos los profesores que dieron su empujoncito a este proyecto, en especial Omarito, a mi tutora Yiseth, a Rudell, Pacheco, Yarina, Linnet y Casanova por sus acertados consejos, A nuestro oponente Yosvany por el magnifico trabajo que desempeñó con nosotros. A la gente de la Wilfredo Lam en especial a Malcolm. A Yandy gracias por aguantarme. A todos muchas gracias y si se me queda alguien por favor acepten mis disculpas

Yandy Peñate Peña:

Gracias a Dios, a mi madre, mi padre, mis abuelos maternos, a Clara y Valdito, a Tio y Libia, a mis primos en especial Noelito, a todos los que compartieron grupo conmigo y a los amigos que no, a todos las profesoras que fueron mas que eso, como la Profe Dariela, Cristina, Merlys, Agueda, los profes como Navarrete, Chacón y Pascual. A Sulivan y a todos sus agradecidos. A todos los que de alguna forma contribuyeron a que llegara a graduarme, gracias también a los que no porque aunque sin quererlo también nos ayudaron.

DEDICATORIA

A mi abuelo, que no pudiste verme ingeniero, si estas allá arriba y me estas mirando espero que estés orgulloso de lo bien que me criaste, para ti es esta tesis. A mi abuela, su otra dueña, por darme tu apoyo y cariño incondicionales. A ustedes dos, mis segundos padres.

Sulivan Lominchar de Varona

Les dedico el producto de nuestros esfuerzos a mis padres.

Yandy Peñate Peña

RESUMEN

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y en especial la informática se hace necesaria en las instituciones legales la búsqueda de soluciones tecnológicas en aras de ganar en eficiencia, rapidez y comodidad, y es por ello que han surgido las bases de datos documentales (BDD)[1]como alternativa a las grandes bibliotecas y centros de documentación. La Unión Nacional de Juristas de Cuba (UNJC) actualmente no cuenta con un sistema de archivos digitales que sea capaz de guardar toda la información doctrinal[2] que genera esta institución.

El siguiente Trabajo de Diploma se centra en la obtención de una base de datos (BD) [3]para esta prestigiosa institución aplicando buenas prácticas para lograr un diseño exitoso. Se alcanzó como resultado un modelo de datos [4]que, habiendo sido validado teórico y funcionalmente teniendo en cuenta un conjunto de aspectos importantes para el área de las bases de datos relacionales, y dentro de ellas las documentales, responde a las necesidades de este prestigioso órgano legal.

[1]Base de datos documental BDD: Son bases de datos especializadas en almacenar documentación, puede ser archivos de texto, audiovisuales, por solo mencionar algunos.
[2] Doctrina: Es resultado de la investigación e innovación de los juristas con respecto a temas del derecho o a las normas.
[a]Base de Datos BD: Sistema informático para el almacenamiento de datos.
[4] modelos de datos: modelos que aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos.

TABLA DE CONTENIDO

INTRODUCCIÓN

El siglo pasado fue fecundo con respecto a las Tecnologías de la Información y las Comunicaciones (TIC) debido a que hubo grandes cambios gracias a su ayuda en todos los aspectos del desarrollo económico-social de la humanidad, condicionando a su vez, una búsqueda gradual de conocimiento que conlleva a la necesidad del descubrimiento de nuevas tecnologías creando así un ciclo vital de desarrollo cognoscitivo.

A mediados de los años 80 vieron la luz los primeros ordenadores personales y las organizaciones como archivos, bibliotecas y centros de documentación comenzaron un proceso de cambio generalizado cuya principal característica era la transición del papel al formato electrónico como una forma de producir, almacenar y recuperar la Información, y dio lugar al surgimiento de las bases de datos, las cuales hoy en día se han convertido en una de las herramientas fundamentales de la sociedad y las cuales deben cumplir dos características principales:

1- El acceso a la información: es uno de los procesos más importantes en el desarrollo político-económico-social de cualquier país. Sin un rápido acceso a la información ningún país podría mantener su posición, y aún más importante, no podría continuar desarrollándose para mejorar su situación.

2- La recuperación de la información: es la forma de organizar, mantener y acceder a todo este conocimiento digital por parte de las personas cada vez que sea necesario de la forma más rápida posible.

A medida que se avanzó en el conocimiento de los Sistemas de Bases de Datos se necesitó incorporar documentos así como otros tipos de multimedia como sonido, video, gráficos y fotografía sin necesidad de tratarlos, solo acceder a ellos y almacenarlos y es así como una simple base de datos se convirtió en una poderosa biblioteca documental en la cual el usuario común podía guardar cualquier archivo imaginable para su uso posterior.

El ámbito jurídico no ha quedado fuera de este tema y los profesionales jurídicos han encontrado un sinnúmero de ventajas debido a la integración real de toda la documentación con los procesos que ya poseían automatizados, y en los últimos tiempos han visto la luz un amplio abanico de gigantescas bases de datos jurídicas para la administración de la información documental judicial.

En la Unión Nacional de Juristas de Cuba (UNJC) se maneja gran cantidad de información, mas no existe una base de datos que almacene toda esta carga documental de información doctrinal y debido a que, con

la nueva implementación de la revista CUBALEX, esa información tendrá una fuente de entrada mas, se hace necesario la creación de una Base de Datos Documental (BDD) que almacene estos documentos.

Por lo tanto surge el problema a resolver que sería: ¿Cómo modelar las necesidades de almacenamiento de la documentación jurídica-doctrinal de la Unión Nacional de Juristas de Cuba a un lenguaje entendible por el programador que permita su posterior implementación contribuyendo así a la correcta gestión de la misma?

El problema planteado tiene como objeto de estudio: Procesos de Gestión Documental.

Como objetivo general se plantea realizar el análisis del Sistema de Información y diseño de la Base de Datos Doctrinal de la Unión Nacional de Juristas de Cuba de manera que permita su posterior implementación contribuyendo así a la correcta gestión de la documentación jurídica-doctrinal de la UNJC.

Se establece entonces como campo de acción: Las bases de datos documentales

Idea a defender: con la realización del análisis del Sistema de Información y diseño de la Base de Datos Doctrinal de la Unión Nacional de Juristas de Cuba se podrá lograr su implementación contribuyendo así a la correcta gestión de la documentación jurídica-doctrinal de la UNJC.

Para dar solución al objetivo antes descrito se debe dar cumplimiento a los siguientes objetivos y tareas:
-Elaborar de un marco teórico de la investigación.
1. Estudio de los sistemas de recuperación de información.
2. Estudio de los sistemas de gestión documental.
3. Estudio de la metodología existente para el diseño de bases de datos
4. Estudio de los sistemas gestores de bases de datos existentes y selección de uno de ellos.
-Analizar de las características del proceso de gestión de la documentación doctrinal de la UNJC.
1. Análisis de los procesos de gestión de la documentación doctrinal de la UNJC.
2. Diseño de la base de datos documental de la UNJC.

3. Validación de la solución propuesta.

Métodos Científicos

Esta investigación se basa en métodos teóricos y empíricos, los cuales facilitarán el proceso investigativo y servirán como base para la organización del trabajo y posibilitar la comprensión del problema así como su análisis y estudio para posteriormente llegar a conclusiones para la solución.

Métodos empíricos:

- Entrevistas: Utilizada comúnmente en la etapa de análisis del sistema de información y diseño para comprender las funcionalidades esperadas de la base de datos.

Métodos Teóricos:

- Analítico – Sintético: Este método permite realizar un estudio de las actuales tendencias en cuanto a creación de bases de datos documentales proporcionando una ayuda a la hora de escoger el gestor de bases de datos a utilizar.
- Análisis Histórico – Lógico: Este método será de gran ayuda a la hora de analizar y estudiar las herramientas principales que son usadas en el manejo de los datos.
- Modelación: Este es un método utilizado comúnmente a la hora de realizar el diseño de una base de datos pues se considera necesario para definir un modelo de datos que cumpla con los requisitos necesarios.

El siguiente trabajo está estructurado de la siguiente forma:

Capitulo 1: En este capítulo se hará una descripción del estado del arte así como de los principales conceptos, definiciones y datos que deben conocerse antes de realizar el diseño y la posterior implementación de la base de datos.

Capitulo 2: A través de este capítulo se conocerán elementos que son necesarios a la hora de realizar el análisis de la aplicación así como sus resultados, además, se adentrará en los resultados del diseño de la aplicación.

Capitulo 3: Este capítulo describe la validación del sistema y el conjunto de pruebas teóricas y funcionales que se le realizaron a la aplicación así como el resultado de las mismas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abordarán temas relacionados con la informática jurídica de la cual se conocerán sus diferentes clasificaciones y algunas características de las mismas; se podrá conocer además qué son los Sistemas de Recuperación de Información (SRI), los tipos y modelos que existen, así como sus componentes, y las distintas operaciones y algoritmos que utilizan estos sistemas. También se profundizará en las bases de datos documentales, su definición, características, topologías, interfaces y sistemas gestores, además de las particularidades que encierra una base de datos jurídica y cómo se realiza la recuperación de información en la misma.

1.2 Informática Jurídica Documental

La Informática Jurídica Documental es la manifestación más difundida de la informática jurídica y su estudio gira sobre cuatro ejes fundamentales: primero, es una herramienta metodológica que presta ayuda a la Ciencia del Derecho y a la Filosofía del Derecho, como segundo eje tiene su objeto de estudio enmarcado entre dos temas fundamentales: las técnicas documentales centradas al tratamiento de la información jurídica que está contenida en la legislación, la jurisprudencia y la doctrina y el segundo tema es las metodologías aplicadas en las ciencias de la información que manejen bases de datos jurídicas. Su tercer eje viene dado por su tema central, constituido por el tratamiento de la información jurídica fundamentado sobre dos operaciones: condensación e indización, y finalmente, el cuarto eje sobre el que gira la Informática Jurídica Documental son las Bases de Datos Jurídicas.

De esta rama parte el análisis de toda la información contenida en los documentos jurídicos para la formación de bancos de datos documentales. Consiste en la aplicación de técnicas informáticas volcadas a la documentación jurídica en diferentes tópicos como son el análisis, archivado y recuperación de la información contenida en la jurisprudencia, legislación, doctrina o cualquier tipo de documentación de contenido relevante, además, aplica técnicas documentales, entendiendo documentación como el acto de reunir información sobre un tema y su tratamiento en vista a su definición.

También comprende todo tipo de análisis documental que es el conjunto de acciones realizadas en vistas de representar un contenido documental, de forma distinta a la original, con el objetivo de facilitar su

consulta o búsqueda en un momento posterior al almacenamiento de esta acumulación de documentos originales o reproducidos, introducidos en la memoria documental de modo que faciliten la operación de recuperación o localización de contenido informativo.

Existen un grupo de características que son importantes a la hora estudiar o desarrollar cualquier aplicación dentro de la informática jurídica documental:

1- La aplicación técnico-jurídica como un sistema de tratamiento y recuperación de la información en el cual se aplican los parámetros de:

- Indización: Elaboración de una lista rígida de descriptores mediante la calificación de la información contenida en el documento fuente a partir de los descriptores considerados apropiados (palabras o locuciones claves), los cuales, son tomados de una lista previamente elaborada de acuerdo con el tipo de información tratado.
- Full Text (texto completo): No es más que el almacenamiento del texto integral en la computadora con el fin de recuperar toda la información contenida en él por cualquiera de las materias que hace referencia.
- Abstract: Es el acto de organizar la información contenida en un acta fuente de forma lógica a través del empleo de restrictores de distancia con el fin de lograr su recuperación así como una presentación sintética.

2- La formación de bancos de datos usando como punto de partida archivos mensuales sistematizados, ya sea sectorizados o integrales.

3- La utilización de lenguajes o mecanismos de recuperación de información utilizando como soporte instrumentos lingüísticos.

La informática jurídica se puede dividir en dos ramas más específicas, una se encarga de todo el proceso de gestión y la otra de la parte decisional.

(Jenny Abella, Informática Jurídica Documental)

1.2.1 Clasificaciones de la Informática Jurídica Documental

La *informática jurídica de gestión* es la rama de la informática jurídica que está encargada de organizar y controlar la información jurídica de documentos, expedientes, libros, entre otros; aplicando programas de administración que permitan la creación de identificadores y descriptores que clasifiquen dicha información. Esta clase de informática es conocida como de administración y control la cual es

ampliamente utilizada en tribunales, notarías, asociaciones jurídicas, por solo mencionar algunos ejemplos. Se utiliza para lograr un seguimiento de los trámites y procesos para mantener actualizada y controlada la información. Esta puede clasificarse como:

1- Registral: Se ocupa de todo tipo de registros ya sean públicos o privados facilitándole a los usuarios datos fieles en registros oficiales con rapidez de uso y facilidad de acceso.

2- Operacional: Facilita la actuación y funcionamiento de las oficinas relacionada con el derecho en las que se permitirá que la máquina lleve toda la actuación repetitiva, o sea, el control de asuntos.

3- Decisional: Utiliza modelos predefinidos para la correcta solución de casos concretos.

La otra rama de la informática jurídica es *la informática jurídica decisional* la cual comprende una gran variedad de esfuerzos y proyectos que planean obtener de la aplicación de la informática a lo jurídico, resultados que, no solo recuperen información de la máquina, también que esta sea capaz de resolver por sí misma problemas jurídicos o al menos, auxilie en la solución de estos problemas, constituyendo un avance a la técnica de la rama del derecho.

La informática jurídica trabaja con varios tipos de información que son la base y parte del resultado del proceso judicial y los cuales funcionan como un perfecto engranaje que mueve la maquinaria de la ley.

(Rodolfo Herrera y Alejandra Núñez, Derecho Informático)

1.2.2 Tipo de Información Jurídica

Es la información jurídica la base de la informática jurídica, esta se puede clasificar en tres tipos principales:

1- Doctrina Jurídica: Es el pensar de los juristas con respecto a temas del derecho o a las normas, no posee fuerza legal mas es una importante fuente inmediata del derecho y su valor depende del prestigio que posee el jurista que la formuló.

2- Jurisprudencia: Es el conjunto de fallos de los tribunales judiciales que sirven de antítesis. Absolutamente todas las sentencias conforman la jurisprudencia a pesar de que no es obligatoriamente una fuente del derecho.

3-Legislación: Es el conjunto de leyes de un estado; son reglas de carácter social y obligatorio las cuales, impuestas por la autoridad pública de forma permanente, tienen sanción por la fuerza.

En el caso de la base de datos objeto de esta tesis, el tipo de información jurídica a guardar es toda la relacionada con la doctrina jurídica, teniendo en cuenta la necesidad de facilitar la búsqueda a un sistema de recuperación de información que se implemente posteriormente.
(http://uscoderecho.wikispaces.com/Definiciones+y+Clasificacion+de+la+Informatica+Juridica)

1.3 Bases de datos jurídicas en Cuba

En Cuba hasta el momento el control y organización de la información jurídica presente en legislaciones, doctrinas y jurisprudencias, se realiza manualmente, a través de la identificación e indización de palabras claves que describen lo esencial de la información almacenada en estos documentos, la cual es organizada a través de lenguajes documentales como el caso de las listas de términos o los tesauros. Lográndose tener en formato duro una mejor conceptualización de esta.

En el contexto digital es muy poco lo que se ha avanzado al respecto. Solo existen algunos casos en los que se han implementado sistemas informáticos que manejan la información jurídica en formato digital con la misma lógica que se hace manualmente a través de los lenguajes documentales antes mencionados. Ejemplo: en el Centro Nacional de Documentación e Información Judicial (CENDIJ), el análisis de la información jurídica se realiza a través de listados de términos bien estructurados lingüísticamente que llegan a funcionar como un mini-tesauro, permitiendo el rápido acceso a cualquier tipo de documento jurídico que se desee analizar y esté almacenado en su base de datos

1.4 Sistemas de Recuperación de la información (SRI)

La recuperación de información es el siguiente nivel a la determinación de las necesidades de información. La recuperación se hace mediante consultas a la base de datos donde se guarda la información organizada mediante un lenguaje de consulta adecuado. Es necesario tener en cuenta los elementos claves que permiten realizar la búsqueda, proporcionando un alto grado de precisión (índices, palabras claves y fenómenos que pueden surgir en el proceso de búsqueda como ruido o silencio documental).El principal problema a la hora de realizar la búsqueda es la cantidad de información que se recupera, ya que, dependiendo del tipo de búsqueda, se puede obtener un gran número de documentos o uno bien reducido. Este fenómeno recibe el nombre de ruido y silencio documental respectivamente.

El objetivo primordial de la recuperación de información (RI) es desarrollar, mediante el estudio, métodos que, ya sean algorítmicos o intelectuales, faciliten el siguiente grupo de operaciones:

1- Indización: Es una operación realizada de forma intelectual que a su vez se divide en análisis, que es la identificación de los temas y conceptos más importantes del documento y normalización, cuyo principal objetivo es transformar los conceptos que expresan el contenido del documento en sus descriptores.

2- Selección: Es la identificación de un conjunto de documentos relevantes dentro de un grupo mayor para resolver una necesidad de información dada. También es denominada recuperación, la cual, siendo la parte más significativa del proceso, a veces sirve para dar nombre a todo.

3- Ordenación: Determinación del orden más adecuado de presentación al usuario del grupo de documentos recuperados o seleccionados. La idea es ofrecerle los documentos en un orden decreciente por relevancia debido a la probabilidad de satisfacer la necesidad de información. A este proceso se le denomina ranking.

4- Interconexión: Es el establecimiento de un conjunto de relaciones hipertextuales, caminos, y en general, estructuras de navegación en secciones del mismo o de distintos documentos.

5- Categorización: Es la asignación de cada documento a un grupo, clase o subclase de un cuadro de clasificación, taxonomía u ontología.

6- Abstracción: Es la producción de resúmenes de los documentos los cuales dependiendo de las circunstancias podrían sustituir la lectura del documento completo.

7- Visualización: Es la representación de forma gráfica de un conjunto de informaciones, conceptos o procesos no necesariamente representativos.

Los sistemas de SRI siguen como pauta un modelo conceptual definido que representa los procesos y entidades que interactúan en el proceso de recuperación de información.

(Antonio Gabriel López Herrera, Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa)

1.4.1 Modelos de un SRI

Un Sistema de Recuperación de Información (SRI) se puede construir utilizando varios modelos de búsqueda y su nombre va dado por la forma en que la realizan:

Modelo Booleano Puro: El resultado de una ecuación de búsqueda booleana es un conjunto de documentos relevantes en el caso de que los haya. Estos son seleccionados siguiendo la lógica booleana la cual plantea que el documento es verdadero cuando contiene al menos un término de la pregunta si el operador es un OR, cuando contiene todos los términos de la pregunta si el operador es un AND o cuando no contiene ninguno de los términos de la pregunta (cuando es un NOT). Plantea la lógica booleana que solo existen verdaderos y falsos así que en este caso solo habrán dos conjuntos de documentos: relevantes y no relevantes lo cual no permite establecer grados de importancia en estos. Otro de los problemas a los que se enfrenta este tipo de modelo es que son anti-intuitivos pues a un usuario poco experimentado le son confundibles los términos booleanos.

Modelo Vectorial: Este modelo plantea que en un espacio vectorial, donde la cantidad de vectores está dada por los términos de indización, y que pueden tomar valores de ceros y unos, todos los documentos situados cercanos a la "pregunta" en el espacio vectorial serán semejantes a esta, siendo este umbral de semejanza el que define los documentos relevantes o no. Este modelo permite situar a los documentos por orden de relevancia poniendo en primer lugar a los que cumplen todos los términos. Este modelo sin embargo es, en la práctica, poco implementado, pues requiere de muchos recursos computacionales para ejecutarse y presenta problemas de recálculo cada vez que se añaden nuevos documentos.

Modelo Booleano-Vectorial: En este modelo al igual que el anterior, los documentos y las preguntas se presentan como vectores pero en lugar de calcular su similitud sobre la base de clústers y espacios vectoriales lo hace contando los elementos en común de los vectores respectivos. La selección del documento se realiza mediante la lógica booleana pero se organiza mediante el método vectorial.

(Antonio Gabriel López Herrera, Modelos de Sistemas de Recuperación de Información Documental Basados en Información Lingüística Difusa)

1.5 Base de datos

Una base de datos es un conjunto de datos que giran en torno a un mismo contexto de información y los cuales son guardados de forma sistemática para poder ser utilizados en cualquier momento, son el pilar fundamental de cualquier sistema de información. En su mayoría se pueden encontrar en formato digital y permite guardar gran cantidad de información de forma ordenada y disponible para ser utilizada

fácilmente, permitiendo una recuperación flexible y ágil. También son una serie de datos interrelacionados entre sí que permiten ser recuperados por un sistema de recuperación de información para uso de una entidad o grupo de entidades determinadas. Algunas operaciones que se pueden realizar sobre ellas son agregar, modificar y eliminar archivos o datos.

Las bases de datos tienen un sinnúmero de aplicaciones, entre ellas la de almacenar documentos que es el propósito de la base de datos para la Unión Nacional de Juristas de Cuba.

(Luis Rodríguez Yunta, Bases de datos documentales: estructura y uso)

1.6 Bases de Datos Documentales

Una base de datos es un conjunto de información que se encuentra estructurada mediante registros y almacenada en un soporte electrónico. Cada registro en sí constituye una unidad autónoma de información que a su vez se encuentra estructurado en campos y tipos de datos de los que se recogen en la base de datos. En una base de datos documental cada registro coincide con un documento de cualquier tipo, desde una publicación digital hasta un archivo audiovisual. Las bases de datos documentales (BDD) se crean y mantienen de forma continuada para resolver las necesidades de información de grupos que van desde un pequeño colectivo hasta el conjunto de la sociedad. Este tipo de recurso electrónico se puede consultar directamente en formato electrónico o usarse para elaborar productos impresos**.**

Las bases de datos se pueden dividir en grupos atendiendo a diversas tipologías, en el caso de los registros de las bases de datos documentales pueden incluir o no el contenido completo de los documentos que describen.

(Luis Rodríguez Yunta, Bases de datos documentales: estructura y uso)

1.6.1 Tipología de las Bases de Datos Documentales

Existen tres grandes modelos de bases de datos:

BD de Información Factual: Este modelo en particular recoge toda la información, generalmente numérica, de estadísticas, resultado de operaciones, convocatorias, encuestas, entre otros más.

BD Directorios: Estas bases de datos se especializan en recoger información sobre personas o instituciones con una material específico.

BD Documental: Cada registro se corresponde a un documento: Publicación digital, audiovisual, gráfico o sonoro, archivos, documentos electrónicos, entre otros.

Las Bases de Datos Documentales a su vez se clasifican en:

BDD de Texto Completo: Son aquella que contienen los propios documentos en formato electrónico con su texto completo, algunas veces incluyen también campos con la información fundamental para facilitar su descripción y recuperación. En estos sistemas la operación de búsqueda y la consulta del documento se realizan sin salir del propio sistema en función.

Archivos Electrónicos de Imágenes: Están constituidos por un grupo de referencias que actúan como enlace hacia la imagen del documento original que pueden ser fotografías, imágenes de televisión, entre otros más. También puede ser un documento digitalizado en forma de imágenes. En este tipo de base de datos la búsqueda es limitada a los campos de referencias bibliográficas y no se pueden localizar otros términos presentes en el texto completo del documento original.

Bases de Datos Referenciales: Sus registros no contienen el texto original, solo la información necesaria para describir y permitir localizar documentos digitales, sonoros, electrónicos, audiovisuales, entre otros más. En este tipo de BDD solo se obtiene una referencia del documento que después se necesitara localizar en otro servicio (biblioteca, fototeca, fonoteca, por solo mencionar algunos.), aunque puede incluir campos que faciliten su localización o enlaces directos a los originales a través de otros programas como tratamiento de texto o navegadores de internet.

1.7 Características de las bases de datos jurídicos

Siendo las bases de datos un grupo de documentos ordenados cronológicamente a los que se puede acceder a través de índices y sumado a las nuevas tecnologías y medios, se puede concluir que resuelven los principales problemas tradicionales de la información jurídica: difusión del ordenamiento jurídico, acceso fácil a las fuentes del derecho, obtención de forma exhaustiva de cualquier texto legislativo o jurisprudencial, y aumento de la seguridad jurídica por medio de interrelaciones legislativas, jurisprudenciales y doctrinales.

Se considera una base de datos jurídica a aquel conjunto de documentos jurídicos básicos los cuales mediante la aplicación de técnicas informáticas jurídicas serán utilizados con una actitud divulgadora pública y generalizada de su contenido. Este grupo de bases de datos poseen características propias: son bases de datos a texto completo donde se emplean distintos tipos de unidades documentales, además contienen un enorme volumen de información almacenada con distintos grados de vigencia, precisan de una constante actualización y su carácter exhaustivo es su garantía de seguridad jurídica, poseen además como característica que su aplicabilidad está delimitada a un ámbito jurisdiccional concreto, existe interconexión de la documentación jurídica, las fuentes documentales están claramente delimitadas y cuentan con sistemas de tratamientos de información adaptados a las necesidades de los profesionales del ámbito jurídico.
(Ma. Luisa Albite Diez, La Recuperación de Información en Bases de Datos Jurídicas)

1.7.1 Opciones de Búsqueda

En general todos los sistemas de recuperación de información permiten realizar diferentes tipos de búsquedas:
Búsqueda Directa: Se teclea directamente una o varias palabras en el espacio reservado para ello por los lenguajes de interrogación dentro de la base de datos. Existen dos tipos dentro de esa modalidad:

- *Interrogación en texto libre* donde el usuario realiza la consulta sin tener en cuenta la organización en campos de los registros de la base de datos.
- *Interrogación en campos individuales* donde la consulta es realizada sobre el campo o los campos que selecciono previamente.

Búsqueda a Través de Índices: En lugar de teclear un término, el usuario hace su consulta a través de índices o listas alfabéticas con todas las entradas ya sea de uno o diferentes campos concretos.
Búsqueda Jerarquizada: La pregunta se realiza a través de una estructura jerárquica donde no solo se localizan los registros de dicho término sino además todos aquellos donde aparezca algún concepto más específico de su campo semántico.
Búsqueda a Través de Códigos: En algunas BDD la búsqueda no se realiza a través de texto sino de código alfanumérico o numérico.

1.8 Modelo de datos

Primero es necesario conocer que un modelo es una representación gráfica de un elemento extraído del mundo real que puede ser un objeto o un evento. Un modelo de datos es un conjunto de conceptos que permiten describir los datos, sus relaciones, la semántica que existe entre ellos y las restricciones de consistencia.

Existen tres grupos de modelos de datos:

1- los modelos externos o lógicos: basados en objetos los cuales permiten representar el conjunto de datos que requiere cada usuario con las estructuras características del lenguaje de programación que se vaya a usar.

2- Modelos globales o lógicos basados en registros: Estos modelos ayudan a escribir los datos para el conjunto de usuarios.

3- Modelo físico de datos: Este es el modelo orientado a la máquina.

El modelo de datos es un lenguaje utilizado para realizar la descripción de una base de datos. Generalmente permite describir las estructuras de datos de la base, o sea, el tipo de dato y sus relaciones, las restricciones de integridad, que son el conjunto de condiciones que los datos deben poseer para reflejar la realidad de lo que se desea; y las operaciones de manipulación de los datos como son agregado, borrado, modificado y recuperado.

1.9 Modelo conceptual

Un modelo de datos conceptual es aquel que describe la estructura de los datos y las restricciones de integridad. Es utilizado durante la etapa del análisis del problema dado y está dirigido a representar los elementos que intervienen en el problema y sus relaciones. Es cercano al usuario y se utiliza independientemente del Sistema Gestor de Base de Datos que se quiera utilizar. Uno de los modelos conceptuales más utilizados es el modelo de entidad-relación con el cual se pretende visualizar los elementos que pertenecen a la base de datos recibiendo el nombre de entidades las cuales son homólogas a las clases en la programación orientada a objetos y en la cual la tupla de una relación encuentra su homóloga en el objeto.

Modelo Lógico

El modelo de datos lógico se centra en las operaciones y es implementado en algún manejador de la base de datos. Es el modelo más cercano al ordenador y depende del sistema gestor que se vaya a utilizar. El resultado de este modelo puede ser el modelo relacional o el modelo jerárquico. En este momento se realiza el proceso de normalización.

Modelo físico

Tiene como salida la implementación de la base de datos en sí, con todas sus restricciones. Considera las estructuras de almacenamiento y los métodos necesarios para proporcionar un acceso eficiente a la base de datos en memoria secundaria. Para la implementación de este modelo se debe tener bien claro que sistema gestor se va a usar pues este modelo debe adaptarse a él, estableciendo los permisos de usuarios por ejemplo, además de la selección de índices para acelerar el proceso y la creación de vistas.
Además de conocer qué modelos se usan para realizar un correcto diseño de la base de datos, es necesario conocer las formas en que se modelan los datos dentro de las mismas.

1.9.1 Tipos de modelados de datos

Debido a su sencillez, el modelado de datos es una de las herramientas más usadas por los diseñadores de bases de datos pues es independiente del gestor a utilizar, lo que representa una de sus principales ventajas. En un modelado de datos solo se debe reflejar la existencia de los datos más importantes sin importar el dominio al cual pertenecen, ni la utilización que se les vaya a dar, así como no se tiene en cuenta tampoco las limitaciones de espacio, almacenamiento o tiempo de ejecución.

Jerárquico:
Un modelo de datos jerárquico se utiliza generalmente para diseñar bases de datos que utilicen esa forma de almacenamiento de la información. En este modelo la organización de los datos es similar a un árbol donde el nodo padre puede tener varios hijos. Un nodo sin padre es llamado raíz, y a su vez, un nodo sin hijos es llamado hoja. Su desventaja principal radica en la incapacidad de brindar una eficiente solución a la redundancia de datos. Este modelo fue desarrollado para modelar situaciones donde dominan las relaciones de uno a uno o de uno a muchos. Estos tipos de bases de datos no permiten el acceso directo

a las instancias de uno de sus hijos, si no son seleccionados previamente las instancias de los padres de los que depende.

De Red:
La diferencia de este modelo con respecto al jerárquico es que en este último, el concepto de nodo sufre una ligera modificación, pues un mismo nodo ahora podrá tener varios padres lo cual no se permite en el modelo anterior representando así una gran mejora ofreciendo una solución eficiente al problema de redundancia de datos, sin embargo, representa una gran dificultad a la hora de la administración, siendo usado más por programadores que por usuarios finales.

Relacional:
Este modelo es el más usado en la actualidad para la modelación de situaciones reales y para la administración dinámica de datos. Su idea fundamental es el uso de entidades compuestas por registros (tablas y filas respectivamente) que representarían tuplas y campos o columnas. Para lograr un mejor diseño, una base de datos relacional pasa por un proceso que se denomina normalización. La forma en que se almacenan los datos en una base de datos relacional carece de importancia logrando facilidad de entendimiento y uso para un usuario esporádico. Toda la información es gestionada mediantes consultas de gran flexibilidad y poder de administración de la información. Sus principales ventajas son: la independencia física y lógica, pues la forma de almacenar los datos no influye con su manipulación lógica y a su vez las aplicaciones que utilizan dicha base de datos no se modificarán si se cambian elementos de la base de datos; flexibilidad pues ofrece distintas vistas en función de usuarios y aplicaciones; y la uniformidad dado que las estructuras lógicas tienen una única forma conceptual.

Otro de los conceptos importantes que se deben manejar con respecto al modelo de datos, son los lenguajes de modelado de datos.

1.9.2 Lenguajes de modelado de datos
Se denomina como lenguaje de modelado de datos a un conjunto de signos y su forma de utilización en combinaciones de disposición para realizar el modelaje de un diseño de software.

Lenguaje de modelado unificado (UML):

Este lenguaje es usado para realizar la visualización, construcción y documentación de un sistema de software. El lenguaje UML fue una de las innovaciones en materia de conceptos que más expectativas ha causado durante muchos años convirtiéndose en un estándar de la industria del software. UML brinda un estándar para describir un plano o modelo del sistema a implementar incluyendo conceptos como proceso de negocio y funciones del sistema y algunos aspectos más concretos como son expresiones específicas del lenguaje de programación a utilizar, esquemas de bases de datos y componentes reutilizables.

En la base de datos a implementar se utilizará este lenguaje de modelado para realizar el diseño del software.

1.10 Fases del diseño

A la hora de diseñar una base de datos es importante conocer cuáles son los pasos a seguir o qué fases componen el diseño de la misma.

En la fase inicial se procede al análisis de los requisitos que debe cumplir la base de datos. En esta fase se hará una descripción de la información que se quiere gestionar y los procesos que deberá realizar el sistema. Durante este período se realizarán una serie de entrevistas a los usuarios finales caracterizando sus necesidades tanto en los datos como en las operaciones que se realizan con estos. Además se realizaran entrevistas a expertos en este tema que pueden aportar conocimientos e ideas que posteriormente pudieran servir en la conformación de la base de datos. El resultado de esta fase es un conjunto de requisitos de datos que son una especificación de la información que se va a guardar, también se obtendrán los requisitos funcionales que son las especificaciones de las operaciones que se realizarán con los datos.

Durante la fase del diseño conceptual se hace una traducción del análisis de requisitos al esquema conceptual obteniendo una representación generalmente gráfica del conjunto de entidades que conformarán la BD y sus relaciones. En esta fase se realiza el modelo de entidad-relación, y en el caso de otros tipos de aplicaciones, el modelo UML y los diagramas de casos de uso, de colaboración, de secuencia, entre otros. La última fase es la implantación del gestor. Durante esta fase se hace el diseño lógico traduciendo del modelo conceptual al modelo relacional y el diseño físico, el cual determina la organización de archivos y las estructuras de almacenamiento interno.

1.11 Herramientas de modelado

Las herramientas de modelado poseen una gran importancia a la hora de realizar el diseño de un Sistema Gestor de Base de Datos (SGBD) pues sirven de puente traductor por el que llegan de un lado los requisitos que pide el cliente y desemboca por el otro el diseño de una forma entendible para el programador.

Enterprise Architect:

Posee un alto rendimiento, interfaz intuitiva tanto para llevar un modelado avanzado al escritorio como para el equipo completo de desarrollo e implementación. Posee un conjunto de características que lo hacen un software de gran calidad como son la alta capacidad, velocidad, estabilidad y buen rendimiento, posee trazabilidad de extremo a extremo y está construido sobre las bases del UML 2.1.

Rational Rose:

Es una de las mejores elecciones para el ambiente del modelado de software con soporte de generación de código en diferentes lenguajes, proporciona un lenguaje común de modelado que facilita la creación de software de calidad más rápidamente. Posee una amplia gama de funcionalidades haciéndolo uno de los más usados a nivel mundial.

Erwin:

Es una herramienta más específica para el modelado orientado a la creación de bases de datos que brinda productividad en su diseño, generación, y mantenimiento de aplicaciones que van desde un modelo lógico de la aplicación hasta el modelo físico ya perfeccionado con las características específicas de la base de datos. Soporta principalmente bases de datos relacionales SQL y bases de datos que incluyan Oracle, Microsoft SQL Server y Sybase. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma a otra.

Visual Paradigm

Esta herramienta para UML es ampliamente utilizada en el mundo entero por los creadores de software pues permite modelar sus diseños a este grupo de profesionales además de permitir la integración de las

aplicaciones empresariales a las bases de datos y es capaz de generar código e ingeniería inversa para algunos lenguajes de programación. Este software es eficiente a la hora de manejar grandes estructuras requiriendo solo una configuración de escritorio común. Soporta el ciclo de vida completo del desarrollo de software desde análisis y diseño orientado a objetos, pasando por la construcción y terminando en pruebas y despliegue.
Con esta herramienta se construyen aplicaciones de calidad de manera rápida y con un ínfimo coste pues permite dibujar todos los tipos de diagramas de clases, generación de código a partir de estos y viceversa, y documentación. Es una aplicación con soporte a aplicaciones web, fácil de instalar y actualizar, todas sus ediciones son compatibles entre sí y es multiplataforma.

A pesar de que todas estas herramientas son de gran profesionalidad y con un sinnúmero de características que las hacen ideales a la hora de modelar un software, se empleará el visual Paradigm pues es uno de los más usados en la Universidad de Ciencias Informáticas así como uno de los más completos y funcionales, con millones de usuarios en el mundo entero, dando como resultado el modelo de la base de datos a implementar.

1.12 Sistemas Gestores de Bases de Datos

Un gestor de bases de datos es un software que se encarga de crear y administrar una base de datos en este caso documental, por las limitaciones que presenta Cuba, se hace necesario que el gestor de bases de datos que se vaya a usar sea libre, potente y profesional, los más usados y con mejores prestaciones son:

MySQL:
Es un sistema de gestión relacional, multihilos y multiusuario el cual es propiedad de Sun Microsystems y este a su vez de Oracle Corporation. Desde abril del 2009 es desarrollado como software libre en un esquema de licenciamiento dual, GNU GPL por un lado para cualquier uso compatible con esta licencia pero con algunas funcionalidades como por ejemplo, para su uso en productos privativos debe ser comprada la licencia específica.

Postgres SQL:

Es un sistema de gestión de bases de datos relacionales orientadas a objetos y totalmente libre publicado bajo la licencia BSD. Como proyecto de código abierto, su desarrollo es manejado por una comunidad de desarrolladores que trabajan de forma desinteresada.

Posee como características una alta concurrencia evitando así que si se accede a una tabla en la cual un proceso está escribiendo, se inicie un bloqueo; posee una amplia variedad de tipos nativos como arrays, direcciones de red, texto ilimitado y figuras geométricas, para no mencionarlos todos.; Además se manejan los conceptos de triggers, vistas, funciones, entre otros más.

Mongo DB:

Es un sistema escalable, libre y totalmente profesional orientado a documentos lo cual lo hace el ideal para crear bases de datos cuya finalidad sea guardar información de revistas digitales y otras páginas web.

Oracle:

Es un sistema de gestión de bases de datos objeto-relacional desarrollado por Oracle Corporation. Es considerado uno de los sistemas más completos pues se destaca en cuanto a soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Su dominio, en sus últimas versiones han sido certificadas para trabajar sobre GNU-Linux

Cassandra:

Es un gestor de código abierto de base de datos distribuida y uno de los proyectos más importantes de la fundación Apache. Es de tipo Nosql adoptada por Google y Facebook. Utiliza un sistema de modelado de datos basado en columnas y súper columnas el cual difiere del modelo relacional de las bases de datos anteriores. Es usado mayormente para gestionar información en sitios web, blogs y revistas digitales.

En el caso de Cassandra es un gestor de tipo Nosql orientado a columnas que al igual que MongoDB se usa para realizar bases de datos fundamentalmente usadas en revistas digitales, páginas web, blogs, entre otros más.

MongoDB es además orientado a documentos lo que no significa que se puedan gestionar estos, sino, que la información la guarda en forma de un documento en sí, haciéndolo perfecto, así como el Cassandra, para sitios web.
Mysql es un gestor de licencia dual pero luego de ser adquirido por la Oracle y por las condiciones impuestas por el bloqueo ya no le es posible a Cuba acceder a él.
Oracle es un software propietario, y puesto que Cuba se encuentra en un proceso de migración hacia el software libre y las políticas en la UCI son orientadas al mismo, hacen a este gestor inapropiado.
Después de analizar todas estas características se propone utilizar Postgres SQL pues además de ser uno de lo más profesionales, es en el tema de base de datos relacionales uno de los más utilizados actualmente y es bastante ágil y flexible, siendo estos uno de los principales requisitos de la base de datos a diseñar.

1.13 Herramientas de Administración de base de datos para Postgres SQL

Las herramientas de administración de datos son las encargadas de administrar una base de datos determinada dentro de un gestor ya instalado, en este caso el Postgres SQL. Estas herramientas permiten realizar múltiples tareas pues facilitan la interacción directa del programador con la base de datos.

Navicat:
Este potente administrador de bases de datos relacionales incluye un amplio abanico de herramientas para gestionar, crear, y sincronizar bases de datos ya sea servidores locales o remotos. Además integra una amplia gama de características y funcionalidades que facilitan ampliamente toda la gestión de BD en Mysql, Postgres, Oracle, por solo mencionar algunos.; facilita la migración de ficheros y posee una interfaz simple, de fácil adaptación para los nuevos usuarios.

EMS Postgres SQL Manager:
Es una útil aplicación que soporta todas las características y gestiona la información que recopila de la planilla, tomando todos o determinados campos sincronizando todos los servidores compatibles con esta base de datos permitiendo visualizarlo desde cualquier PC. Sus principales características son el espacio entre tablas, la posibilidad de renombrar argumentos, y una fácil administración de todos los objetos

Postgres así como una herramienta para la interpretación de las consultas entre otras. Su interfaz es muy sencilla aunque poderosa y optimiza ágilmente la navegación entre los datos.

Microsoft SQL Server:

Es un sistema de administración de bases de datos basado en modelos relacionales y presentado por Microsoft Corporation. Es un servidor que solo trabaja sobre sistemas operativos de Windows y posee integración con Acces y Power Shell utilizando su entorno de desarrollo y seguridad respectivamente.

PGAdmin III:

Está diseñado para responder a las necesidades de los usuarios, desde escribir consultas simples hasta el desarrollo de las bases de datos más complejas. El interfaz grafico soporta todas las funciones de Postgres y facilita enormemente la administración. Es la más completa y popular con licencia Open Source. Se puede usar en casi todos los sistemas operativos. Es la herramienta para Postgres más utilizada.

Debido a que Navicat, el Microsoft SQL Server y EMS Postgres SQL Manager son privativas y que la política informática tanto de la Universidad como del país es enfocada hacia el software libre, para la administración de Postgres se utilizará el PGAdmin III.

1.14 Conclusiones parciales

Se abordó en este capítulo todo lo referente a la informática jurídica así como los datos necesarios a conocer sobre Sistemas de Recuperación de Información a la hora de diseñar una base de datos documental, se explicaron las diferentes clasificaciones de las bases de datos indicando a cuáles pertenecerá la que se desea diseñar, además se brindó una breve explicación de los Sistemas Gestores de Base de Datos Documentales existentes en el mundo y cuál de ellos se usará para el proyecto.

CAPITULO 2: DISEÑO DE LA APLICACIÓN.

2.1 Introducción

En este capítulo se expondrán algunos conceptos, procedimientos y elementos claves en el proceso de diseño de la presente base de datos documental de la Unión de Juristas de Cuba, además se hará un estudio de los Sistemas de Recuperación de Información y los tipos de bases de datos documentales. Se expondrán también una relación de las tablas de la base de datos y una breve descripción de estas.

2.2 Metodologías para el diseño de bases de datos

Una metodología es un conjunto o grupo de procedimientos así como técnicas y ayudas para el desarrollo de un software, en este caso una base de datos. Una metodología traza las actividades a seguir en el desarrollo de principio a fin de la base de datos y qué es lo que debe realizar en cada actividad, señalando la entrada y el producto de salida, sin olvidar quién está involucrado.
Actualmente la más utilizada es la Metodología de análisis de Sistemas de Información y diseño de bases de datos documentales de Luis Codina dado que es bien explícita en cuanto a los instrumentos de análisis, las bases conceptuales y los procedimientos para interpretar problemas de información y diseñar un software de este tipo y es por ello que es esta la que se adaptará a las necesidades y especificaciones obtenidas por la UNJC para el diseño de una base de datos documental que cumpla con los parámetros de organización doctrinal de la información que requiere esta organización.

2.3 Metodología de Lluís Codina aplicada a la elaboración de una base de datos para la Unión de Juristas de Cuba

Toda metodología para el diseño de una base de datos relacional o documental consta de tres etapas principales. Primero se tendría un aparato conceptual para crear los fundamentos conceptuales y teóricos para el entendimiento correcto del sistema a extrapolar, definiendo las entidades básicas que entran a jugar en nuestro proyecto así como la toma de enfoques estratégicos para las posteriores etapas. El aparato instrumental sería el encargado de brindar los elementos para el análisis y diseño, o sea el ¿cómo

hacer?, y por último, el aparato procedimental donde se establece las fases y los procedimientos básicos, señalando sus objetivos, así como identifica y describe los productos que deben obtenerse de cada fase de análisis, incluido el producto final.

1- Fase de Análisis:

Dada la filosofía planteada por Lluís Codina en su metodología; la realidad es llamada sistema de objetos al que se pretende representar en una imagen conceptual y bien delimitada por los objetivos perseguidos; esta abstracción es llamada: sistema de información, en este caso, se quiere llegar más allá, se pretende lograr un Sistema de Recuperación de Información. Para llegar a esta meta se debe subdividir el sistema de objetos en dos subsistemas, uno de actividad humana y otro de conocimientos**.**

Sistema de Actividad Humana (SAH):

El SAH es la parte de la realidad que involucra las personas y sus procesos que necesitan este sistema, o sea en el universo de discurso de la base de datos documental. En el caso concreto de la BD que se desea implementar, el sistema de actividad humana serían los juristas autorizados a acceder a la base de datos y todos sus procesos de gestión de la información en función de lograr guardarla o recuperarla, según sea el caso.

Sistema de Conocimientos (SCO):

El SCO involucra los tipos de entidades sobre los que se tendrá registro en la base de datos, o sea, ya se empieza a abstraer del término o entidad física por ejemplo libros, tesis y artículos de revistas jurídicas que son el pilar fundamental de la doctrina jurídica. La herramienta recomendada para dar solución y salida a esta fase es el Modelo Entidad Relación (MER)**.** Dado que ya se ha hecho todo este análisis previo se procede a la subsiguiente fase del diseño.

2- Fase de Diseño:

El objetivo de esta fase es obtener un Modelo Conceptual de la base de datos y una propuesta de tratamiento documental. El modelo conceptual asiste el proceso de implantación. La propuesta de tratamiento documental establece la forma y la norma que se seguirá para la descripción y representación de los documentos.

(Lluís Codina, Metodología de Análisis de Sistemas de Información y diseño de Bases de Datos Documentales, Aspectos Lógicos y Funcionales).

2.4 Patrones de diseño de bases de datos

En el diseño de las bases de datos y en el modelado de datos en general son encontrados elementos repetitivos en disímiles modelos, los que correctamente identificados pasan a ser parte de los patrones de diseño, es por ello que un patrón se puede definir como el fragmento de un modelo que es recurrente. Es una solución a un problema específico que se ha mantenido a pesar del tiempo. Existen muchos patrones, mas en el caso de la base de datos de esta tesis solo se usará uno:

Patrón una tabla por dominio: Este patrón plantea el uso de una tabla nomencladora para cada dominio lo que trae como ventaja que es fácil de entender cuando se escriben consultas, posee un mejor rendimiento en las consultas dado la menor cardinalidad de los datos asociados y además permite una fácil implementación de interfaces de usuario dado un mapeo casi exacto de las tablas.

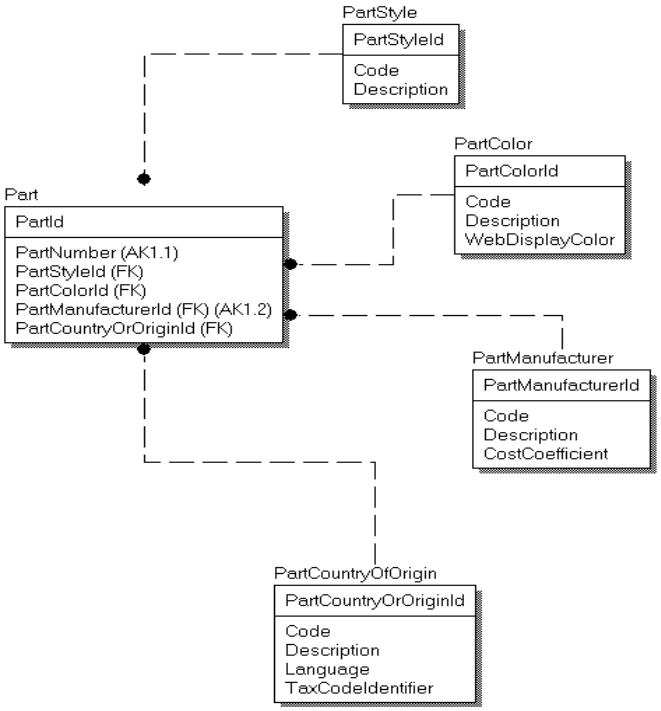


Figura 1: Patrón una tabla por dominio.

2.5 Modelo Conceptual:

Tema y dominio de la base de datos:

Esta base de datos propiedad de la Unión Nacional de Juristas de Cuba, está enfocada a la salva y posterior recuperación de de documentos de tipo doctrina: ensayos, artículos, informes de investigación (publicados y no publicados), tesis, libros, revistas y boletines. Con el objetivo de facilitar la consulta de los documentos y a los efectos de la gestión y recuperación, la información doctrinal puede clasificarse en materias: Administrativa, Civil, Penal, Económica, Laboral, Constitucional, Ambiental, Registral, Notarial, Telecomunicaciones, Informática, por solo citar algunos ejemplos.

2.6 Estándares de nomenclatura en la base de datos

Para la nomenclatura de la base de datos se usará el PascalCase que es un procedimiento común en la programación basado en el CamelCase el cual forma un identificador uniendo tantas palabras como sean necesarias con la única diferencia que la primera letra de cada una debe estar en. Además en todos los casos los nombres deberán ser descriptivos, o sea, deberán estar acordes con el propósito de lo que están nombrando.

Ejemplo: primerApellido

Apariencia de los esquemas: Se seguirá el estándar de PascalCase poniendo la letra inicial de cada palabra en mayúsculas pero la inicial del nombre como tal en minúsculas. Ejemplo: esquemaBaseDatos.

Nombre de las tablas: Con las tablas se seguirá el mismo procedimiento que con los esquemas. Ejemplo: tablaDocumento

Nombre de los campos: Se seguirá el mismo estándar usando PascalCase con la primera en minúscula. Ejemplo: documentoGuid.

Nombre de las llaves primarias: Las llaves primarias usarán el método anterior y se le agregará antes del nombre la letra “k” seguido por el nombre de la tabla. Ejemplo: kDocumento.

Nombre de las llaves foráneas: Al igual que las llaves primarias, será el nombre de la tabla donde es llave primaria antecedido por la letra “k”. Ejemplo kAutor.

2.7 Tipos de datos

El tipo de datos viene dado de acuerdo al dominio que maneja cada variable ya sean numéricos, alfanuméricos, fechas, para no mencionarlos todos. Los tipos de datos son estándares para todos los Sistemas Gestores de Bases de Datos.

Numeric: Este es el tipo de datos que se usa comúnmente para guardar solo números, su grado de precisión está dado por la cantidad de cifras antes y después de la coma y se representa (x, y) en el caso de la base de datos doctrinal es de (19,0) pues serán números muy grandes pues no se tiene un límite exacto, pero solo números enteros por lo que después de la coma la precisión es cero.

Date: Este tipo de datos comúnmente se usa para almacenar fechas y en este caso es importante a la hora de almacenar los datos de las fechas mencionadas en los documentos.

Varchar: Este tipo de datos se utiliza para representar cadenas de caracteres que contienen cualquier tipo de símbolo. Comúnmente se utiliza para guardar guids, nombres, y pequeños textos. Para los atributos de este tipo se reserva un estimado de 56 caracteres para permitir la internacionalización (UTF-8 =ASCII*3). El mismo precepto fue considerado para campos de tamaño superior.

Text: Se utiliza para guardar textos de gran tamaño. El tipo text es reservado en este caso para campos de texto más extensos que los alcanzables por el varchar (255), sea el caso del resumen del documento, notas del autor y la referencia bibliográfica, para la elección de este último entre este grupo tan exclusivo, esta base de datos se basa en su carácter súper mixto y variable.

Bytea: Se utiliza para guardar todo tipo de documentos sin importar su formato (JPEG, PDF, WORD, entre otros mas.). En esta base de datos en particular los documentos, imágenes o documentos con imágenes, se guardarán como cadenas de bytes aprovechando el tipo de datos BLOB.

En el caso de los identificadores universales se escogió el tipo ***guid***, de postgres con una extensión de 40 caracteres (varchar 40). Se decidió no establecerlos como llave ya que su extensión y formato dificultaría cualquier manipulación del mismo y serán ingresados a través de la función de PHP com_create_guid().

2.8 Trabajo con índices

Un índice no es más que un subgrupo ordenado de las columnas de la tabla, con una entrada que apunta a la tupla correspondiente. Trabajando con un subconjunto indexado se puede realizar un procesamiento más veloz para cada solicitud de consulta, pues se elimina la necesidad de búsqueda en la tabla completa, optando en su lugar por enfocarse en una mínima porción de ella. La indexación es una técnica que optimiza el acceso a datos, acelera las consultas lo que provoca una mejora de rendimiento. Un índice se define sobre las columnas de la tabla que más a menudo son usadas, sobre los campos que más frecuentemente se realicen operaciones de búsqueda. Una tabla puede ser indexada por campos tanto de tipo numérico como de tipo carácter, además puede ser indexada también por un campo que contenga valores null con excepción de las llaves primarias. Su uso permite mejorar el rendimiento de las bases de datos, permitiendo recuperar filas específicas mucho más rápido de lo que podría hacerse en su ausencia. Se pueden usar índices con los comandos SELECT, UPDATE y DELETE vinculados a las condiciones de búsqueda, además en concatenaciones con las tablas.
(http://www.postgresql.org/docs/8.3/static/textsearch.html)

2.8.1 Tipos de índices

BTREE: Gestiona las consultas usando operadores de igualdad o de rango de datos. Su estructura tiene la forma de un árbol invertido donde las estructuras superiores son llamadas ramas y las inferiores hojas. Generalmente los índices en BTREE tienen uno o más niveles de ramas donde son estas ramas las que contienen la columna índice (claves) y la dirección de otro bloque y los nodos hojas poseen la clave de cada fila de la columna. Estos índices se utilizan además para evitar las grandes operaciones de

ordenación y obtienen su mejor resultado cuando son aplicados sobre columnas de alta cardinalidad, o sea, que contengan muchos valores diferentes.

HASH: Este tipo de índices solo pueden manejar simples comparaciones de igualdad. El planificador de consulta considera su uso cada vez que una columna indexada está involucrada en una comparación con el operador igual. No se admiten búsquedas usando el operador IS NULL.

GIST: Estos no son solo un tipo de índice, sino una infraestructura sobre las que se pueden implementar muchas estrategias de indexación, los operadores usados varían según estas estrategias. GIST es sinónimo de búsqueda generalizada sobre un árbol. Una de sus ventajas es que permite desarrollar tipos de datos personalizados con sus adecuados métodos de acceso. Gist sustituyo los antiguos r-trees, es usado para tipos complicados como tipos geométricos, búsquedas en texto, por solo mencionar algunos.

GIN: Este índice además conocido como índice invertido puede manejar valores que contienen más de una clave como los arreglos. Al igual que GIST puede apoyar diferentes estrategias de indexación que son definidas por el usuario. Es usado para buscar en texto, tarea en la cual llega a ser más rápido que gist pero más lento para añadir nuevos valores, por tanto optamos por quedarnos con este para las búsquedas en texto.
(http://www.postgresql.org/docs/8.3/static/textsearch.html)

2.8.2 Selección de Índices

Una de las mejores formas de escoger un índice es, primero que todo, evaluar las consultas más importantes y luego, para cada una, determinar cuál será el plan que escogerá el optimizador dentro de los posibles índices a crear; luego se necesita saber si añadiendo mas índices a la lista se obtendría un mejor plan y de ser así se hace. Un índice de tipo BTREE es el más óptimo para recuperaciones de rango, mientras que un índice HASH lo es para las recuperaciones por coincidencias exactas. Antes de que cualquier índice sea añadido a la lista es necesario preguntarse qué impacto podría tener en las actualizaciones de la carga de trabajo pues todos los índices de un atributo de trabajo serán actualizados

siempre que el valor de ese atributo cambie, por lo que se hace necesario ralentizar algunas actualizaciones en la carga de trabajo para acelerar las consultas.

Cuando Indexar: Es necesario no construir índices que no beneficien a ninguna consulta, siempre que sea posible que aceleren a mas de una.

Escoger la llave de búsqueda: Siempre es necesario tener en cuenta que los atributos de las cláusulas WHERE son candidatos a ser indexados.

Llaves de búsqueda con atributos múltiples: Los también llamados índices compuestos son considerados, bien si una cláusula WHERE incluye condiciones en más de un atributo de la relación, o si se habilita la estrategia de evaluación "solo índices" para las consultas importantes la cual pudiera provocar que existan atributos en la llave de búsqueda que no están en la clausula WHERE.

Balancear el costo de mantenimiento de los índices: Para ello se considera el impacto de cada uno de los índices candidatos en las actualizaciones de la carga de trabajo. Es necesario valorar si mantener un índice ralentiza las operaciones de actualización frecuente, caso en el que es necesario eliminarlo; o bien si el índice puede acelerar una actualización dada.
(http://www.postgresql.org/docs/8.3/static/textsearch.html)

2.8.3 Ventajas

El usar índices evita la sobrecarga del CPU y del disco así como la concurrencia, además permite mayor rapidez en la ejecución de las consultas, también es una ventaja en los campos que no contengan datos duplicados.
(http://www.postgresql.org/docs/8.3/static/textsearch.html)

2.8.4 Desventajas

En aquellas tablas donde se utilizan continuamente operaciones de escritura no son aconsejables ya que los índices se actualizan cada vez que se modifica una columna, tampoco se aconsejan en tablas demasiado pequeñas ya que en estas no se necesita ganar tiempo en consultas. Otra desventaja es que ocupan espacio y en determinadas ocasiones este es mayor que el de los propios datos.

En el caso de la base de datos objeto de estudio se usara el tipo de índice BTREE y HASH, y la nomenclatura será i_nombreCampo_Tabla. Ejemplo: **i_nombreSegundoNombre_Autor.**
(http://www.postgresql.org/docs/8.3/static/textsearch.html)

2.8.5 Índices utilizados

GIN:
En nuestra experimentación dado una misma consulta con todos los tipos de índice arrojo que era notablemente el más eficiente.
Nombre de autor (de la tabla Autor)
Legislaciones (RN + No + nombre + organismo + fecha) (de la tabla Legislación)
Título del documento (de la tabla Documento)
(Ver Anexo 3).

B-Tree:
Para la selección de esta tomamos una misma consulta y modificamos los tipos de índice, como se puede apreciar en los anexos apenas hay diferencias de tiempo, pero las muestras fueron tomadas para los valores mínimos, resultando mucho más estable las secuencias arrojadas por el B-Tree.
keyword (de la tabla Keyword)
descriptor (de la tabla Descriptor)
noDescriptor (de la tabla NoDescriptor)
(Ver anexo 4).

2.9 Selección y argumentación de los requisitos del sistema.

El conjunto de requisitos marca la pauta que el sistema debe cumplir para complacer las exigencias y expectativas del usuario y además es una fuente vital para lograr la comunicación entre los diseñadores y programadores del sistema por un lado y todo usuario futuro por el otro.

2.9.1 Requisitos funcionales:

Los requisitos funcionales son un conjunto de capacidades o condiciones que el sistema debe cumplir, estableciendo de esta forma el cómo se debe comportar el mismo.

Gestionar documentos: Al llegar un documento el usuario asignado a su revisión le asigna las palabras claves y llena el resto de los campos con sus respectivos datos, en el caso de un anexo se fotografía el anexo y se guarda a su vez con sus datos.

1- Adicionar documento: El sistema permitirá introducir un documento con los datos adyacentes al mismo.
2- Modificar un documento: El sistema permitirá modificar en cualquier momento la información de un documento guardado.
3- Eliminar documento: El sistema debe permitir eliminar un documento que se encuentre en la base de datos junto a toda la información perteneciente a él.

Además el usuario podrá realizar las siguientes acciones durante o después de guardado un documento:

Adicionar palabra clave: el usuario adicionará en una tabla las palabras claves del documento para poder usarlas a la hora de realizar una búsqueda y además para que no se repitan al almacenar otro documento con una palabra clave igual. Esta información se eliminará si todos los documentos con esa palabra clave son eliminados.

Adicionar legislación: el usuario adicionará en una tabla las legislaciones contenidas dentro del documento para que no se repitan al almacenar otro documento con una legislación igual. Esta información se eliminará si todos los documentos con esa legislación son eliminados.

Adicionar anexo: el usuario adicionará en una tabla los anexos contenidos dentro del documento para que no se repitan al almacenar otro documento con un anexo igual. Esta información se eliminará si todos los documentos con ese anexo son eliminados.

Adicionar tipo de anexo: el usuario adicionará en una tabla los tipos de anexos contenidos dentro del documento para que no se repitan al almacenar un anexo con un tipo de anexo igual. Esta información se eliminará si todos los anexos de ese tipo son eliminados.

Adicionar idioma: el usuario adicionará en una tabla el idioma contenido dentro del documento para que no se repita al almacenar otro documento con el mismo idioma. Esta información se eliminará si todos los documentos con ese idioma son eliminados.

Adicionar materia: el usuario adicionará en una tabla las materias contenidas dentro del documento para que no se repitan al almacenar otro documento con las mismas materias. Esta información se eliminará si todos los documentos de esa materia son eliminados.

Adicionar fecha mencionada: el usuario adicionará en una tabla las fechas mencionadas dentro del documento para que no se repitan al almacenar otro documento con una fecha igual. Esta información se eliminará si todos los documentos que contienen esa fecha son eliminados.

Adicionar autor: el usuario adicionará en una tabla el autor del documento para que no se repitan al almacenar otro documento con el mismo autor. Esta información se eliminará si todos los documentos de ese autor son eliminados.

Adicionar tipo de documento: el usuario adicionará en una tabla el tipo de documento para que no se repitan al almacenar otro documento con el mismo tipo. Esta información se eliminará si todos los documentos de ese tipo son eliminados.

Adicionar referencia bibliográfica: el usuario adicionará en una tabla las referencias contenidas dentro del documento para que no se repitan al almacenar otro documento con las mismas referencias. Esta información se eliminará si todos los documentos con esa referencia son eliminados.

Adicionar identificador o topónimo: el usuario adicionará en una tabla los identificadores o topónimos contenidos en el documento para que no se repitan al almacenar otro documento con los mismos topónimos o identificadores. Esta información se eliminará si todos los documentos con ese identificador o topónimo son eliminados

Todos los datos que se adicionarán en tablas podrán ser modificados al cambiar el documento en cuestión.

2.9.2 Requisitos no funcionales

Los requisitos no funcionales tienen como función primordial lograr un producto de gran usabilidad, atractivo, seguridad y rapidez. Estos requisitos no describen ni la información que se guardara ni ningún tipo de función a realizar, solo son propiedades o cualidades que el sistema debería tener.

Seguridad:

1- Debe realizarse copias de seguridad de forma tal que se puedan recuperar íntegramente los datos en caso de fallo, con preferencia semanal.
2- El sistema gestor seleccionado debe presentar facilidades para administrar los roles de usuarios logrando restringir de esta forma el acceso a los datos.

Soporte:

Se necesita un servidor para la base de datos que soporte un gran volumen de datos.

Software:

Servidor de base de datos Postgres SQL 8.3

Hardware:

Lado del Servidor:

1- Tarjeta de Red: 1
2- Procesador : Dual Core 2.5 GHZ
3- RAM: 4GB (mínimo)
4- Disco duro: 1 TB (mínimo)
5- UPS: 1

Lado del cliente:

1- Procesador: 2.00 GHZ (mínimo)
2- RAM: 256 MB (mínimo)
3- Tarjeta de Red: 1
4- UPS: 1

2.10 Modelo Entidad-Relación

Un diagrama o modelo entidad-relación, también denominado Entity Relationship (ER) es una herramienta que se utiliza para el modelado de datos de cualquier sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.
En el caso de la base de datos doctrinal para la UNJC el diagrama entidad relación sería:

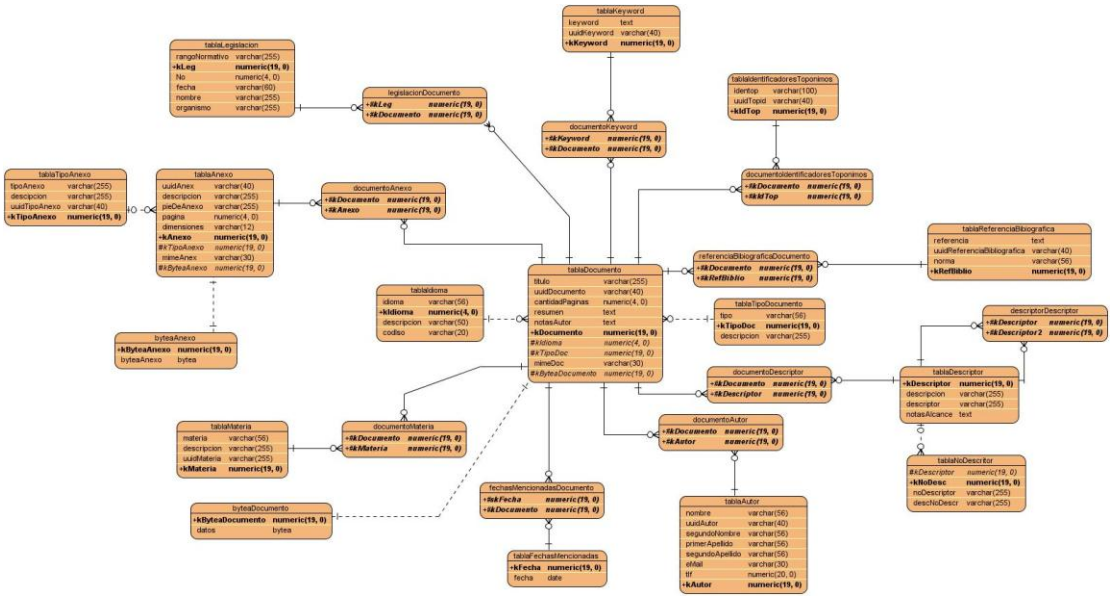


Figura 2 Diagrama de diseño de la base de datos.

2.11 Diccionario de Datos

Tabla Documento:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento Documental***	***Idioma***	***Controles de validación***
título	Título del documento. El título se transcribe de la siguiente forma: *Título: antetítulo: subtítulo*	Alfanumérico	Sí	Lenguaje libre	Lenguaje del documento	No puede quedar vacío. Si por alguna razón, el documento careciera de título, al

						guardarse el documento en la base de datos se debe asignar un título descriptivo
cantidadPáginas	Cantidad de páginas que ocupa un documento. Se transcribe de la siguiente forma: cantidad de páginas iniciales enumeradas con números romanos en minúsculas: cantidad de páginas en números tradicionales: cantidad de páginas finales en números	Numérico	No			No puede quedar vacío

	romanos minúsculas, si posee ilustraciones o si posee retratos. Se pueden utilizar las normas ISBD del 2007(IFLA).					
resumen	resumen del documento	texto	Sí	Lenguaje controlado	español	No puede quedar vacío
notasAutor	Conjunto de notas sobre el libro escritas por el autor.	texto	No		español	puede quedar vacío
textoCompleto	es el documento en sí	blob	Sí	lenguaje libre	El idioma del documento	no puede quedar vacío
kDocumento	es un número que funciona como id del documento y es único para cada	numérico	no			no puede quedar vacío, no puede ser igual al de

	uno preferentemente incremental					otro documento
mimedoc[5]	Tipo de archivo del documento (pdf, doc, entre otros más.)	alfanumérico				No puede quedar vacío

Tabla autor:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
nombre	nombre del autor del documento	alfanumérico	Sí	Lenguaje controlado	español	puede pertenecer a varios de los autores o a uno solo, no puede quedar vacío
segundoNombre	segundo nombre del	alfanumérico	Sí	Lenguaje controlado	español	Puede pertenecer

[5] ***Multipurpose Internet Mail Extensions*** o **MIME** (en español "extensiones multipropósito de correo de internet") son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, entre otros mas.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.

	autor del documento					a varios de los autores o a uno solo, puede quedar vacío.
primerApellido	primer apellido del autor del documento	alfanumérico	Sí	Lenguaje controlado	español	Puede pertenecer a varios de los autores o a uno solo, no puede quedar vacío.
segundoApellido	segundo apellido del autor del documento	alfanumérico	Sí	Lenguaje controlado	español	Puede pertenecer a varios de los autores o a uno solo, no puede quedar vacío.
kAutor	es un número	numérico	no			no puede

	que funciona como id del autor y es único para cada uno preferentemente incremental					quedar vacío, no puede ser igual al de otro autor
tlf	Teléfono del autor	numérico	no			Solo puede albergar números

Tabla IdentificadoresToponimos:

Etiqueta	***Dominio***	***Tipo***	***Indexación***	***Tratamiento Documental***	***Idioma***	***Controles de Validación***
identop	Identificadores y topónimos contenidos dentro del documento, pueden ser nombres propios, lugares geográficos, entre otros más.	alfanumérico	Sí	Lenguaje controlado	español	No puede quedar vacío.
kIdtop	es un número que funciona	numérico	no			No puede quedar

	como id del topónimo o identificador y es único para cada uno preferentemente incremental					vacío, no puede ser igual al de otro identificador o topónimo.

Tabla referenciaBibliográfica:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
referencia	Referencia dentro del documento	texto	Sí	Lenguaje controlado	español	No puede quedar vacío.
norma	norma con la que se hizo la referencia	alfanumérico	Sí	Lenguaje controlado	español	no puede quedar vacío
kRef	es un número que funciona como id de la referencia y es único para cada una preferentemente incremental	numérico	no			no puede quedar vacío, no puede ser igual al de otra referencia

Tabla idioma:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
idioma	Idioma en que está escrito español, inglés, entre otros más.	alfanumérico	No		español	no puede quedar vacío
descripción	Descripción del Idioma en que está escrito español, Inglés, entre otros más.	alfanumérico	No		español	no puede quedar vacío
codIso	Código del subtipo del idioma. Ejemplo si es español argentino, español castellano, entre otros más.	alfanumérico	No		español	no puede quedar vacío
kIdioma	es un número que funciona	numérico	no			no puede quedar

	como id del idioma y es único para cada uno preferentemente incremental					vacío, no puede ser igual al de otro idioma

Tabla keyword:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
keyword	Palabras claves asignadas pertenecientes al tesauro.	alfanumérico	Sí	Lenguaje controlado	español	no puede quedar vacío
kKey	es un número que funciona como id de la palabra clave y es único para cada uno preferentemente incremental	numérico	no			no puede quedar vacío, no puede ser igual al de otra palabra clave

Tabla materia:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
materia	Materia dentro de la doctrina sobre la que trata el documento	alfanumérico	Sí	Lenguaje controlado	español	No puede quedar vacío.
descripción	descripción de la materia dentro de la doctrina sobre la que trata el documento	Fijo	Sí	Lenguaje controlado	español	No puede quedar vacío.
kMateria	es un número que funciona como id de la materia y es único para cada uno preferentemente incremental	numérico	no			no puede quedar vacío, no puede ser igual al de otra materia

Tabla fechasMencionadas:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento*	*Idioma*	*Controles*

				Documental		de Validación
día	Valor de día de la fecha mencionada	numérico	Sí	numérico		No puede quedar vacío.
*kF*echa	es un número que funciona como id de la fecha y es único para cada uno preferentemente incremental	numérico	no			no puede quedar vacío, no puede ser igual al de otra fecha
mes	Valor de mes de la fecha mencionada	alfanumérico	Sí	Lenguaje controlado		No puede quedar vacío.
año	Valor de año de la fecha mencionada	numérico	Sí	*numérico*		No puede quedar vacio

Tabla tipoDocumento:

Etiqueta	*Dominio*	*Tipo*	*Indexación*	*Tratamiento Documental*	*Idioma*	*Controles de Validación*
tipo	Corresponde al	alfanumérico	Sí	Lenguaje	español	Solo puede

	grupo al que pertenece: *Artículo, Tesis, Boletín, Libro…*			controlado		pertenecer a uno de los tipos delimitados, no puede quedar vacío
descripción	descripción del tipo de documento dentro de la doctrina sobre la que trata el documento	Fijo	*Si*	Lenguaje controlado	español	No puede quedar vacío.
*kT*ipo	es un número que funciona como id del tipo de documento y es único para cada uno preferentemente incremental	numérico	no			no puede quedar vacío, no puede ser igual al de otro tipo de documento

Tabla legislación:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento***	***Idioma***	***Controles***

				Documental		***de validación***
nombre	Corresponde a las leyes mencionadas en los documentos	alfanumérico	Sí	Lenguaje controlado	español	No puede quedar vacío.
número	Corresponde al número de la legislación	numérico	Sí	Lenguaje controlado	español	Puede quedar vacío.
fecha	fecha de la legislación puede ser una fecha completa o parte de ella	alfanumérico	Sí	Lenguaje controlado	español	Puede quedar vacío.
rangoNormativo	Corresponde al rango normativo de la legislación dada	alfanumérico	Sí	Lenguaje controlado	español	Puede quedar vacío.
organismo	Corresponde al organismo al que pertenece	alfanumérico	Sí	Lenguaje controlado	español	No puede quedar vacío.
*kL*eg	es un número que funciona como id de la legislación y es	numérico	no			no puede quedar vacío, no puede ser

	único para cada uno preferentemente incremental					igual al de otra legislación

Tabla anexo:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento Documental***	***Idioma***	***Controles de validación***
kAnexo	Es el identificador del anexo que posee el documento	numérico	no			no puede quedar vacío, no puede ser igual al de otra legislación
descripción	*Descripción del anexo*	*alfanumérico*	*Si*	*Lenguaje natural*		*No puede quedar vacío*
pieAnexo	*Título que está en el pie del anexo*	*Alfanumérico*	*Si*	*Lenguaje natural*		*No puede quedar vacío*
página	Página en que se encuentra el	*Numérico*	*no*			*No puede quedar vacío*

	anexo					
dimensiones	*Dimensiones del anexo*	*numérico*	*no*			*No puede quedar vacío*
imgAnexo	Imagen anexa que puede ser una foto, un gráfico, por solo mencionar algunos.	*bytea*	*no*			*No puede quedar vacío*
mime	Tipo de archivo (jpeg, bitmap, por solo mencionar algunos)	*alfanumérico*	*no*			*No puede quedar vacío*

Tabla tipoAnexo:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento Documental***	***Idioma***	***Controles de validación***
kTipoAnexo	Es el identificador del tipo de anexo que	numérico	no			no puede quedar vacío, no puede ser

	posee el documento					igual al de otra legislación
tipoAnexo	*Nombre del tipo de anexo*	*alfanumérico*	*no*			*No puede quedar vacío*
descripción	*Descripción del tipo de anexo*	*Alfanumérico*	*no*			*No puede quedar vacío*

Tabla descriptor:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento Documental***	***Idioma***	***Controles de validación***
kDescriptor	Es el identificador del descriptor que posee el documento	numérico	no			No puede quedar vacío, no puede ser igual al de otro descriptor.
descriptor	*Nombre del descriptor*	*alfanumérico*	*no*			*No puede quedar vacío*
descripción	*Descripción del descriptor*	*Alfanumérico*	*no*			*No puede quedar vacío*

notasAlcance	*Identifica su significado en caso de que sea una palabra con más de uno, Ej.: palma (puede ser de la mano o la palma real).*	*texto*	*no*			*Puede quedar vacío*

Tabla noDescriptor:

Etiqueta	***Dominio***	***Tipo***	***indexación***	***Tratamiento Documental***	***Idioma***	***Controles de validación***
kNoDesc	Es el identificador del no descriptor que posee el documento	numérico	no			No puede quedar vacío, no puede ser igual al de otro no descriptor.
noDescriptor	*Nombre del no descriptor*	*alfanumérico*	*no*			*No puede quedar vacío*
descNoDescr	*Descripción*	*Alfanumérico*	*no*			*No puede*

	del no descriptor					*quedar vacío*

2.12 Optimización de la base de datos en Postgres

Una base de datos nunca alcanzará su rendimiento máximo hasta que no esté optimizada pues con la optimización se prevé que haga su máximo esfuerzo con menos recursos. Para optimizar una base de datos, en este caso en Postgres, es necesario aplicar una serie de técnicas que garanticen que continuará realizando el mismo trabajo que se desea pero consumiendo menos recursos y tiempo. En este epígrafe se resumen una serie de pasos referidos a la optimización para servidores de bases de datos en Postgres.

Optimizar el diseño de los datos y la aplicación: Lo primero que se debe tener en cuenta a la hora de realizar este tipo de optimización es cuáles datos deben ser persistentes, qué relaciones y atributos son más importantes, y por último estructurar toda la información de forma tal que cumpla con los requerimientos del sistema. El próximo paso es normalizar los datos de forma tal que se evite lo más posible las redundancias, mas como algunas veces la normalización afecta algunos problemas de rendimiento, se hace necesario en estos casos desnormalizar, teniendo en cuenta realizarlo donde sea necesario solamente y que el problema esté dado por la normalización y no por otras causas. Una buena idea puede ser remover algunas reglas y restricciones de la misma poniendo a chequear estas a la aplicación, mas es necesario, comprobar que el problema no persistirá cambiando solo el lugar, pues se compromete de esta forma la integridad de los datos lo cual impide futuramente que otra aplicación haga uso de ellos. Durante el proceso de diseño de los datos se debe tener en cuenta cuáles atributos se indexan y cuáles no, observando cuáles son los campos de mayor frecuencia de acceso, cuáles son utilizados comúnmente en consultas de comparación, entre otros. Otro aspecto a tener en cuenta es que la aplicación que accede a sus datos lo haga correctamente y trate de minimizar la cantidad de conexiones y transacciones en la medida de lo posible.

Optimizar los índices y los caminos de acceso: Los índices son utilizados en una base de datos con el fin de mejorar e incrementar la velocidad de búsqueda dentro de la BD. Estos se crean a partir de varias columnas de una tabla o a partir de la copia de una parte de la tabla, en el caso de las bases de datos

relacionales. Su utilización es uno de los aspectos importantes a la hora de buscar rendimiento pues una incorrecta estrategia de indexado puede repercutir negativamente en este. Los índices que no se usan usualmente o que no forman parte de una restricción, causan demora en las inserciones, actualizaciones y borrados en las tablas a las que están ligados, además hacen que se retarden las operaciones de VACUUM, resguardo y restauración de la base de datos y permite que el planificador de ejecución de consultas se tome más tiempo en realizar el análisis sobre qué camino de ejecución tomará para las consultas realizadas sobre esas tablas, pues necesita decidir si usará o no los índices, por lo que una forma de aumentar el rendimiento, es eliminar estos índices inactivos.

Optimizar las consultas SQL: El diseño de las consultas es otro aspecto a tener en cuenta debido a algunas particularidades de Postgres, las cuales deben ser manejadas por el equipo de desarrollo.
El planificador de consultas de Postgres (Query Planner) decide como ejecutará determinada consulta. Esto es causa de la forma declarativa del SQL donde el programador especifica qué va a devolver la consulta y no cómo hacerlo. En SQL existen muchas equivalencias por lo que una consulta no trivial puede ser escrita en muchas formas diferentes y además esto trae como equivalencia que son ejecutadas en muchas formas diferentes también por lo que un planificador de consultas de bases de datos es decidir cuál es la manera más eficiente de ejecutar una consulta y hacerlo lo suficientemente rápido pues si el tiempo que se toma en decidir cómo hacer la consulta más el tiempo de ejecución es mayor que el tiempo que se toma en el primer camino que pueda aparecer, entonces no se está resolviendo ningún problema.
La forma de representar un plan de consulta es un árbol, donde cada nodo es una operación del gestor para ejecutar la consulta. El planificador escoge el plan basándose en su costo estimado, asumiendo que lo que más tiempo toma son las búsquedas en disco, escoge el plan que menos búsquedas tenga. Estos estimados están basados en estadísticas que son almacenadas en el catálogo del sistema, las cuales, pueden no siempre estar actualizadas o completas, lo que implica que el planificador puede realizar una mala elección, no saberlo y seguir trabajando bajo el criterio de que el plan escogido fue el correcto para determinada consulta. Es necesario entender los mecanismos de procesamiento de Postgres para escribir sentencias SQL teniendo en cuenta a la hora de escribir una nueva consulta o afinar una consulta con bajo rendimiento en una aplicación que esté en producción, que los dos principales puntos son: el uso del CPU y los procesos de entrada y salida del disco.

Optimizar el fichero Postgresql.conf: En este fichero se encuentran todos los parámetros que determinan el entorno de ejecución del servidor de Postgres, algunos de ellos se pueden fijar mediante las opciones de arranque del servidor y tienen valores por defecto, algunos vinculados al código fuente y otros como una opción de configuración en tiempo de compilación. La configuración por defecto está orientada a la compatibilidad y no al rendimiento. La configuración del fichero Postgresql.conf requiere conocer los tipos de datos que acepta cada parámetro. Las líneas que comiencen con "#" están comentadas por lo que los valores dentro de ellas no tendrán efectos aunque comentar el valor de un parámetro no restaura su valor por defecto , además si el mismo parámetro esta especificado varias veces, el último escrito es el que Postgres tomará como válido.

Optimizar el sistema operativo: A pesar de todas las mejoras que se le pueden introducir en el diseño de las reglas del negocio, de los datos, de la aplicación, en el gestor de bases de datos o el sistema de archivos, si el sistema operativo sobre el que está corriendo el servidor no está apropiadamente configurado será casi imposible que la base de datos que se quiere optimizar presente un buen rendimiento. Para la hora de afinar un sistema operativo es necesario tener en cuenta el sistema de archivos usado, memoria RAM compartida, la versión y los problemas de seguridad y parches. Se recomienda la instalación de Ubuntu Server o cualquier sistema operativo libre para servidores.
Se debe evaluar el paro de los servicios del sistema operativo que no sean necesarios en un momento determinado, esto debe mejorar el rendimiento del sistema pues se puede usar esa porción de memoria para otros procesos y aplicaciones.

Optimizar la estructura física: Antes de la versión 8.1 PostgreSQL usaba un tamaño de página fijo de 8 Kb, ahora esos objetos grandes pueden ser comprimidos o picados en múltiples filas sin que el usuario tenga noción de lo que sucede. Esta técnica es llamada TOAST. Esta operación solo es soportada por algunos datos, además, no hay necesidad de sobrecargar en tipos de datos demasiado grandes.
Los valores out-of-line son divididos (después de la compresión si es usada) en trozos de aproximadamente 2 Mb.
El código TOAST reconoce cuatro diferentes estrategias para el almacenaje en las columnas Toastables:
PLAIN: Impide la compresión o almacenamiento out-line, esta es la única posible estrategia para tipos Toastables.

EXTENDED: Permite la compresión y el almacenamiento out-line, este es el default para la mayoría de los tipos Toastables.
EXTERNAL: Solo permite almacenamiento out-line, hará más rápido las operaciones de substring y bytea (con la penalidad de aumentar el espacio en disco), porque estas operaciones son optimizadas para obtener solo las partes del out-line cuando no está comprimido.
MAIN: Solo permite compresión (realmente el almacenamiento out-line se hará si no queda otra opción para hacer el campo pequeño).
Cada tipo de dato Toastable viene con una estrategia específica pero la misma puede ser modificada. Este esquema tiene ventajas pues permite que el valor de una Tupla tome más de una página y asume que la mayor parte de las consultas están dirigidas a comparación contra valores pequeños y estos datos son solo usados para retornarlos, sin sobrecargar así el shared-buffer-caché.
Dada nuestra situación con las imágenes y texto completo de los documentos, hemos optado por definir para estos campos una estrategia EXTENDED, primero intentando la compresión y luego procediendo a almacenarlo out-of-line en caso de ser necesario.

Optimizar el hardware: A la hora de afinar el hardware existen cuatro aspectos fundamentales que son necesarios tener en cuenta. Primero es el CPU pues con cuanta más capacidad de memoria se cuente se logrará un mejor funcionamiento, aunque con un presupuesto limitado es necesario priorizar RAM y discos siempre que la base de datos no requiera cálculos complejos. En el caso del CPU es mejor utilizar uno de varios núcleos, con caché L2 y arquitectura de 64 bits, ya que permiten una mayor velocidad de acceso y mayor espacio de la RAM. El segundo punto a tener en cuenta es la RAM pues mientras más capacidad tenga, mejor, pues se podrá contar con mas caché, y esto permitirá minimizar el acceso a disco que es cientos de veces más lento que el acceso a la RAM. El tercer aspecto a tratar es el disco. Los discos SCI son recomendados para tener un servidor de BD, pero son extremadamente caros así que la opción ideal es usar discos Sata 2 que tienen buena velocidad y son mucho más baratos, pudiéndose comprar más para repartir los ficheros de log y tablas en diferentes discos, evitando las congestiones en uno solo. El último aspecto es la red. En su caso lo más común y óptimo es que en el servidor de base de datos no se ejecute ninguna de las aplicaciones clientes por lo que el tráfico de información circulará por la red. Si es una aplicación web lo ideal sería instalar dos tarjetas de red, una para las peticiones de los clientes y otra para la conexión con el servidor, evitando así cuellos de botellas o congestiones.

Otro aspecto importante que se necesita conocer es que la calidad es un aspecto crucial y que un driver o componente inadecuado o de baja calidad pueden destruir el rendimiento de la aplicación.

En el caso de la base de datos de la UNJC se recomienda como otra forma de optimizar el sistema, la sistematización de los VACUUM, en el horario que estadísticamente tenga menos afectaciones para los clientes, además en dependencia de si se le diera de alta a un número grande de documentos sería necesaria la ejecución de esta función a pesar de las afectaciones acarreadas. Otra opción de optimización es el cpu_index_tuple_cost dentro del Postgresql.conf el cual trae un valor por defecto de 0.005 y han sido cambiados a 0.001, además se le incluirá un máximo de conexiones de 100 con un mínimo de cada conexión de 32 Kb pues se va a disponer de 64 Mb al buffer compartido (shared_buffers =64 Mb), se recomienda que sea el 25 %, si se usara más del 40 % sería necesario aumentar el número de checkpoint_segment para compensar el proceso de escribir grandes cantidades de datos a través de un período mayor de tiempo, en el caso de este último, es el número máximo de segmentos de archivos LOG entre los checkpoint automáticos del WAL, por defecto son tres segmentos, si se aumenta este número también lo hace el tiempo para la recuperación ante desastres; pues se están revisando los checkpoints. También es preciso revisar el checkpoint_timeout que no es más que el tiempo entre cada checkpoint WAL. Otra mejora dentro de este fichero es aumentar el maintenance_work_mem, que es la memoria necesaria para operaciones de mantenimiento como VACUUM y el CREATE INDEX, a 32 Mb (tiene 16 Mb por defecto); en el caso del max_stack_depth, que es la pila de memoria que debe ser unlimit pero con un límite inferior para evitar desbordamientos, por defecto su valor es de 2 Mb, pero según el propio manual de Postgres este valor es muy conservador e impide la realización de operaciones de mayor envergadura, es por ello que le pondremos 8 Mb, además el SSL será verdadero (true), el autentication_timeout será de un minuto y se habilitará el password_encryption.

En el mismo archivo se puede revisar además, con el fin de lograr una mejor optimización, el temp_buffers que es el número máximo de buffers temporales para cada sesión de la BD, buffers de sesión local y que son solo usados para el acceso a tablas temporales (por defecto 8 Mb). En el caso de la cantidad de memoria disponible para ordenamientos y tablas HASH, este se utiliza para operaciones de las consultas que pueden necesitar muchos archivos temporales, además de que esta memoria es compartida para todas las sesiones haciendo consultas, por lo cual se estima necesario darle un valor de 10 Mb. Otro valor a optimizar es el de effective_io_concurrency, el cual fija el número de operaciones de entrada\ salida concurrentes (1000 por defecto), si la base de datos está muy frecuentemente ocupada por consultas en

sesiones concurrentes es necesario elevar este valor, pero hay que tener cuidado, un valor más alto de lo necesario ocupará demasiado espacio al CPU, proponemos elevar a 2000 este valor.

Uso de Tablespaces

Estableceremos un Tablespace diferente para la bytea de los documentos y anexos, con el objetivo de fomentar el máximo rendimiento de la base de datos, propiciando mayor estabilidad, escalabilidad y seguridad. Estos Tablespaces deben estar en una unidad de disco diferente y de preferencia de una mayor velocidad. Además consideramos que también los índices podrían ser albergados en este, pero dejamos esta decisión en manos de nuestro cliente.

También contribuyendo al rendimiento dada nuestra situación con las imágenes y texto completo de los documentos hemos optado por definir para estos campos una estrategia EXTENDED (TOAST) primero intentando la compresión y luego procediendo a almacenarlo out-of-line en caso de ser necesario.

2.13 Conclusiones parciales

En este capítulo se analizaron aspectos relevantes del diseño de la base de datos, se explicó las funcionalidades y se definió el diseño final de la base de datos que se le entregará a la UNJC para su utilización.

CAPITULO 3: VALIDACIÓN DE LA BASE DE DATOS.

3.1 Introducción

En este capítulo se estudiará el tema de las validaciones hechas a la base de datos y dentro de este se verán las validaciones teóricas y funcionales,

3.2 Validación teórica

El diseño de una base de datos es una de las etapas más importantes en el desarrollo del sistema que se desea implementar por lo que se debe tener en cuenta una serie de aspectos importante ya que permiten garantizar el mejor diseño. Estos aspectos son la integridad, redundancia y seguridad de los datos, asegurando de esta forma el acceso a los datos y la protección de estos contra situaciones no controladas.

3.2.1 Integridad de la información

La integridad de la información en una base de datos se refiere a la corrección y exactitud de toda la información almacenada. Las bases de datos sin importar su tipo pueden estar sometidas a un grupo de restricciones de integridad con complejidad diversa, garantizando que los datos no se vuelvan corruptos.

Integridad de datos: La integridad de los datos se refiere a la corrección y completamiento de los datos y puede afectarse cuando se realizan modificaciones al contenido de la base de datos, ya sea por adición de datos incorrectos o por modificación de datos existentes haciendo que tomen valores incorrectos o causando su eliminación. La integridad de los datos se puede observar a diferentes niveles. Las restricciones de dominio, transacciones y entidades son el pilar principal para definir las reglas para el mantenimiento de la integridad en las relaciones individuales. Las relaciones de integridad referencial aseguran la mantención de la integridad en las relaciones. Las restricciones de integridad en una base de datos gobierna la misma como una sola unidad y las restricciones de integridad de las transacciones controlan la forma en que se manejan los datos ya sea dentro de la base de datos o entre varias de ellas. En una base de datos relacional la integridad de los datos es una de las funciones más importantes.

Integridad de entidad: La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices o restricciones.

Integridad de dominio: Este tipo de integridad viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante el tipo de datos, además del formato y el intervalo de valores posibles mediante restricciones, definiciones y reglas.

Integridad referencial: esta integridad protege las relaciones entre las tablas cuando en estas son creadas o eliminadas sus filas. Se basa en las relaciones de las claves externas con las claves principales o exclusivas. La integridad referencial garantiza que los valores de claves sean coherentes en las distintas tablas, para obtener esa coherencia es imprescindible que no haya referencias a valores inexistentes, y que si cambia el valor de una clave, todas las referencias a ella en la base de datos cambien a su vez.

3.2.2 Redundancia de los datos

La redundancia es la repetición innecesaria de una información determinada dentro de la base de datos lo cual podría causar dificultades a la hora de realizarle un cambio a los datos siendo uno de los elementos más frecuentes de las inconsistencias. Además el poseer redundancia en la base de datos produce una necesidad adicional de espacio que puede llevar a aumentar los costes de almacenamiento y acceso a esos datos. Un modelo de datos normalizado no debe contener redundancia pues esto implicaría la necesidad de usar triggers o disparadores para asegurar la consistencia cuando se gestionan los datos. Algunas veces la redundancia puede ser a su vez favorable como a la hora de evitar joins para búsquedas y campos calculables o para evitar subconsultas correlativas, lo mejor sería analizar bien si es mejor la utilización o no de la redundancia en la base de datos.

3.2.3 Normalización de la base de datos

Las formas normales son aplicadas a las tablas de una base de datos con el objetivo de eliminar la redundancia, es decir, que se repitan los datos. Decir que una base de datos está en la forma normal N es decir que todas sus tablas se encuentran en la forma normal N. En general las tres primeras formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos.

Primera Forma Normal:

Una tabla estaría en una forma normal si y solo si todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, o sea, mínimos. Otra característica que define una tabla en primera forma normal es el hecho de poseer una clave primaria única además de no contener atributos nulos, no debe existir variación en el numero de columnas, los campos no claves deben identificarse por la clave y, por ultimo debe existir una independencia del orden tanto en las filas como en las columnas o sea el cambio de orden no debe cambiar su significado. El resultado será una base de datos libre de variables repetida.

La base de datos no se puede llevar a la primera forma normal puesto que es necesario mantener los valores nulos en algunos atributos.

Segunda Forma Normal:

Una relación esta en segunda forma normal si esta en primera forma normal y los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal, o sea, que no existan dependencias parciales. Se podría decir que la segunda forma normal está basada en el concepto de dependencias completamente funcionales.

La base de datos no se puede llevar a segunda forma normal puesto que no se pudo llevar a primera forma normal.

Tercera Forma Normal:

Para llevar una tabla a tercera forma normal es necesario que ella este en segunda forma normal y no debe existir ninguna dependencia funcional transitiva entre los atributos que no son clave.

La base de datos no se podrá llevar a tercera forma normal puesto que no cumple con la condición de estar en segunda forma normal.

3.3 Validación funcional

La validación funcional no es más que el conjunto de pruebas que se le realizan a la base de datos con el objetivo de ver su comportamiento en un entorno de trabajo que simula el real comprobando la integridad

de los datos con el fin de determinar si la base de datos en cuestión cumple los requisitos funcionales solicitados inicialmente.

3.3.1 Prueba de volumen

Las pruebas de volumen son realizadas con el fin de conocer si la base de datos aguanta la cantidad de datos para la que fue diseñada o si el sistema falla, además de realizar un llenado de la base de datos que será útil para realizar las pruebas de carga.

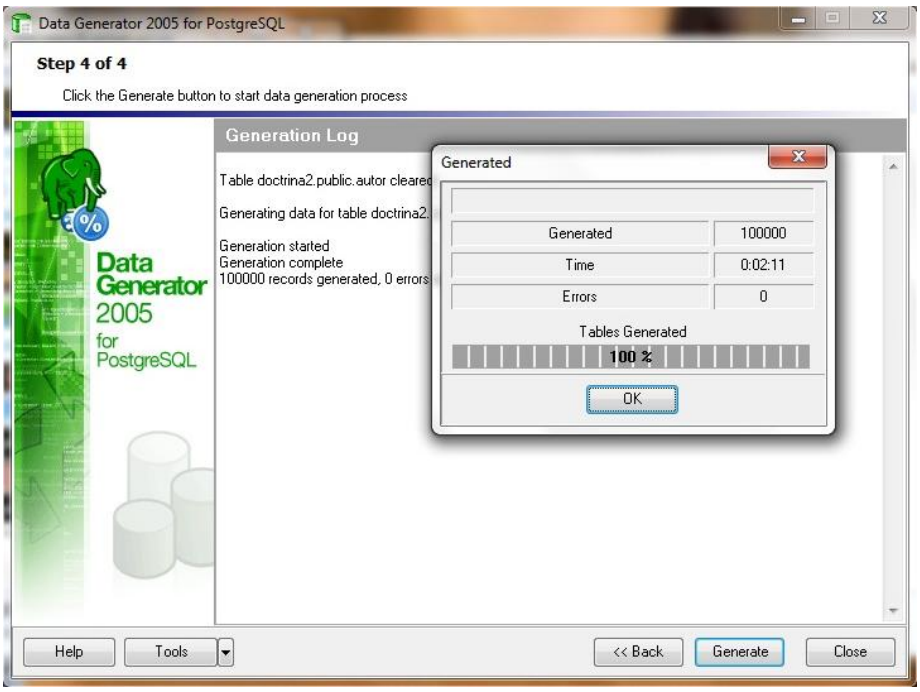


Figura 3: Prueba de volumen para la tabla autor.

Para esta prueba se utilizó la herramienta Data Generator 2005 para PostgresSQL insertando una cantidad de datos equivalente a lo que sería llenado en no menos de un año quedando como más probable a voluminosas las tablas fechasMencionadas, identificadoresTopónimos y autor las cuales llenamos con 100 000 cada una demorando en llenarse 1:27, 1:33 y 2:12 minutos respectivamente.

Mientras se añadían datos a las tablas no se presentaron problemas de desbordamiento de atributos, columnas o tipo de datos, ni de límite de capacidad. Utilizando esta herramienta se pudo verificar la

integridad de los datos, además de garantizar que el diseño de las estructuras de la BD y el gestor utilizado para su desarrollo, soportan el cúmulo de información requerido para su funcionamiento.

3.3.2 Prueba de carga

Con este examen se controlan las consultas críticas del sistema y su tiempo de respuesta.

Listar todos los documentos de un mismo autor

Consulta que, dado el autor, devuelve todos los documentos escritos por este.

Tablas relacionadas

autor con 500 registros.

documento con 500 registros.

Consulta

select * from (documento join documentoautor on(documento.kdocumento=documentoautor.kdocumento)) join autor on(documentoautor.kautor = autor.kautor) where autor.nombre='yandy4'

Resultado

La siguiente imagen muestra el resultado de la prueba de concepto listar documentos dado un autor:

Panel de Salida

Salida de datos | Comentar | Mensajes | Historial

	QUERY PLAN text
1	Nested Loop (cost=27.48..37.71 rows=1 width=153) (actual time=0.242..0.392 rows=1 loops=1)
2	-> Hash Join (cost=27.48..37.33 rows=1 width=94) (actual time=0.224..0.373 rows=1 loops=1)
3	Hash Cond: (documentoautor.kautor = autor.kautor)
4	-> Seq Scan on documentoautor (cost=0.00..7.98 rows=498 width=14) (actual time=0.012..0.052 rows=498 loops=1)
5	-> Hash (cost=27.45..27.45 rows=2 width=80) (actual time=0.199..0.199 rows=2 loops=1)
6	-> Seq Scan on autor (cost=0.00..27.45 rows=2 width=80) (actual time=0.010..0.195 rows=2 loops=1)
7	Filter: ((nombre)::text = 'yandy4'::text)
8	-> Index Scan using documento_pkey on documento (cost=0.00..0.36 rows=1 width=59) (actual time=0.014..0.015 rows=1 loops=1)
9	Index Cond: (documento.kdocumento = documentoautor.kdocumento)
10	Total runtime: 0.465 ms

Figura 4: Resultado de listar documentos dado un autor.

Listar todos los documentos dado el idioma y el autor

Consulta que, dado el autor y el idioma, devuelve todos los documentos que respondan a esta clasificación.

Tablas relacionadas

autor con 500 registros.

documento con 500 registros.

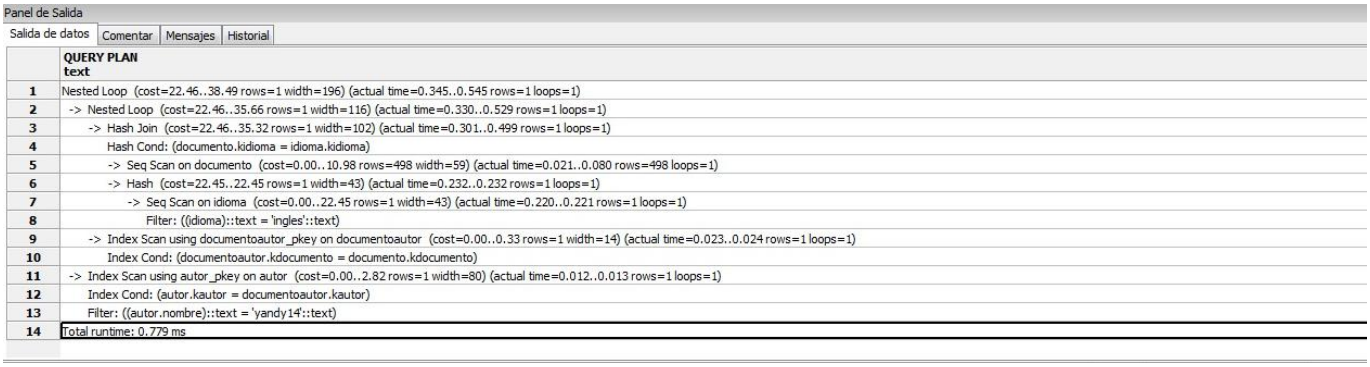
Idioma con 500 registros

Consulta

explain analyze select * from ((documento join idioma on (documento.kidioma=idioma.kidioma)) join documentoautor on (documento.kdocumento=documentoautor.kdocumento))join autor on (documentoautor.kautor=autor.kautor) where idioma.idioma='ingles' and autor.nombre='yandy14'

Resultado

La siguiente imagen muestra el resultado de la prueba de concepto listar documentos dado un autory un idioma:



Panel de Salida

Salida de datos | Comentar | Mensajes | Historial

	QUERY PLAN text
1	Nested Loop (cost=22.46..38.49 rows=1 width=196) (actual time=0.345..0.545 rows=1 loops=1)
2	-> Nested Loop (cost=22.46..35.66 rows=1 width=116) (actual time=0.330..0.529 rows=1 loops=1)
3	-> Hash Join (cost=22.46..35.32 rows=1 width=102) (actual time=0.301..0.499 rows=1 loops=1)
4	Hash Cond: (documento.kidioma = idioma.kidioma)
5	-> Seq Scan on documento (cost=0.00..10.98 rows=498 width=59) (actual time=0.021..0.080 rows=498 loops=1)
6	-> Hash (cost=22.45..22.45 rows=1 width=43) (actual time=0.232..0.232 rows=1 loops=1)
7	-> Seq Scan on idioma (cost=0.00..22.45 rows=1 width=43) (actual time=0.220..0.221 rows=1 loops=1)
8	Filter: ((idioma)::text = 'ingles'::text)
9	-> Index Scan using documentoautor_pkey on documentoautor (cost=0.00..0.33 rows=1 width=14) (actual time=0.023..0.024 rows=1 loops=1)
10	Index Cond: (documentoautor.kdocumento = documento.kdocumento)
11	-> Index Scan using autor_pkey on autor (cost=0.00..2.82 rows=1 width=80) (actual time=0.012..0.013 rows=1 loops=1)
12	Index Cond: (autor.kautor = documentoautor.kautor)
13	Filter: ((autor.nombre)::text = 'yandy14'::text)
14	Total runtime: 0.779 ms

Figura 5: Resultado de listar documentos dado un autor y un idioma.

Listar idioma y autor de todos los documentos con más de 20 páginas

Consulta que devuelve el autor y el idioma de todos los documentos que tengan más de 20 páginas.

Tablas relacionadas

autor con 500 registros.

documento con 500 registros.

Idioma con 500 registros.

Consulta

explain analyze select idioma.idioma, autor.nombre from ((documento join idioma on documento.kidioma=idioma.kidioma))join documentoautor on(documento.kdocumento=documentoautor.kdocumento)) join autor on(documentoautor.kautor=autor.kautor)where documento.cantidadpaginas >20

Resultado

La siguiente imagen muestra el resultado de la prueba de concepto listar autor e idioma de los documentos de más de 20 páginas

Panel de Salida

Salida de datos | Comentar | Mensajes | Historial

	QUERY PLAN text
1	Hash Join (cost=67.56..108.08 rows=498 width=16) (actual time=1.531..2.508 rows=498 loops=1)
2	Hash Cond: (documentoautor.kdocumento = documento.kdocumento)
3	-> Hash Join (cost=14.21..47.88 rows=498 width=15) (actual time=0.248..0.900 rows=498 loops=1)
4	Hash Cond: (autor.kautor = documentoautor.kautor)
5	-> Seq Scan on autor (cost=0.00..24.96 rows=996 width=15) (actual time=0.012..0.132 rows=996 loops=1)
6	-> Hash (cost=7.98..7.98 rows=498 width=14) (actual time=0.218..0.218 rows=498 loops=1)
7	-> Seq Scan on documentoautor (cost=0.00..7.98 rows=498 width=14) (actual time=0.008..0.064 rows=498 loops=1)
8	-> Hash (cost=47.13..47.13 rows=498 width=15) (actual time=1.271..1.271 rows=498 loops=1)
9	-> Hash Join (cost=18.45..47.13 rows=498 width=15) (actual time=0.446..1.079 rows=498 loops=1)
10	Hash Cond: (idioma.kidioma = documento.kidioma)
11	-> Seq Scan on idioma (cost=0.00..19.96 rows=996 width=15) (actual time=0.007..0.218 rows=996 loops=1)
12	-> Hash (cost=12.23..12.23 rows=498 width=14) (actual time=0.432..0.432 rows=498 loops=1)
13	-> Seq Scan on documento (cost=0.00..12.23 rows=498 width=14) (actual time=0.011..0.244 rows=498 loops=1)
14	Filter: (cantidadpaginas > 20::numeric)
15	Total runtime: 2.665 ms

Figura 6: Resultado de listar autor e idioma de documentos con más de 20 páginas.

Además se preparó un caso de prueba en Apache jmeter 2.4, con un controlador de peticiones random conteniendo las tres consultas planteadas anteriormente.

Condiciones iniciales:

Número de hilos: 100.

Máximo de conexiones concurrentes 80.

Edad máxima de conexión: 5000ms.

Resultando:

0% de errores.

Un tiempo máximo de conexión de 4399 ms.

Rendimiento 22,7/sec a 3kb/s.

(Ver Anexo 1 y 2).

Luego incrementamos (100+500=600 muestras):

Número de hilos:500.

Máximo de conexiones: 300.

3.4 Seguridad de la base de datos

La seguridad en una base de datos es básicamente el conjunto de acciones que toma el diseñador al momento de crearla , tomando en cuenta el volumen, el volumen de las transacciones y las restricciones que debe especificar en el acceso a los datos, esto permitirá que el usuario adecuado sea quien visualice la información adecuada. Además, es también el conjunto de acciones que se toman para proteger la información contenida dentro de la bd contra fallos de software o hardware que puedan provocar su corrupción parcial o total. La seguridad de una base de datos atiende también los aspectos dedicados a proteger la información de una eliminación o modificación por parte de usuarios no autorizados.

Se puede brindar seguridad de los datos ante una falla o corrupción del sistema haciendo copias de respaldo, la más recomendable sería semanalmente pues el flujo de documentos en la base de datos no va a ser ni tan abundante como para una copia de respaldo diaria ni tan pequeña como para un tiempo mayor.

3.5 Conclusiones parciales

En este capítulo se analizaron un grupo de pruebas hechas a la base de datos que permiten su validación teórica y funcional, arrojando resultados que dan por sentado el pleno funcionamiento de la base de datos y el pleno cumplimiento de esta con los requisitos establecidos.

CONCLUSIONES

Habiendo concluido el presente trabajo se puede afirmar que el mismo ha cumplido con su objetivo a través del diseño e implementación de una base de datos que permitió guardar toda la información doctrinal de la UNJC. Durante la etapa completa de la investigación se realizó un estudio acerca de elementos teóricos claves que estaban relacionados con el diseño de la base de datos favoreciendo el desarrollo del presente trabajo. La BD propuesta cumple con los requerimientos propuestos, también cumple con los aspectos de validez de datos creando restricciones que lo garanticen, por lo que la inconsistencia es nula. Además, la base de datos fue expuesta a diferentes pruebas que garantizarán su validez obteniendo un sistema con un mayor control y organización de la información.

RECOMENDACIONES

Luego de haber concluido el presente trabajo es recomendado:

- ✓ Que sea tenido en cuenta el diseño propuesto para futuras versiones de la base de datos documental de la Unión Nacional de Juristas de Cuba.

- ✓ Que se le de continuidad al proyecto realizado en pos de fortalecer el diseño del mismo teniendo en cuenta cualquier eventualidad que pueda presentarse o alguna modificación que se desee implementar.

BIBLIOGRAFIA

- Knosys [consultado abril 15, 2011]. Disponible en: http://www.softwareseleccion.com
- Papiro Gestión documental [consultado abril 15, 2011]. Disponible en: http://www.hsmsoft.com
- Inmagic DB/TextWorks Sistema de gestión de bases de datos documentales, 2010. Disponible en: http://www.doc6.es
- MongoDB, Daniel Vigueras, Diciembre 7, 2010 [consultado abril 15, 2011]. Disponible en: http://www.mongodb.org
- Bases de Datos, José Valle [consultado abril 15, 2011]. Disponible en: http://www.monografias.com
- La recuperación de la información en bases de datos jurídicas: Evaluación de Aranzadi y La Ley, Luisa Alvite Diez [Consultado abril 15, 2011]
- Cap18 de PostgreSQL 8.4.4 Documentation, The PostgreSQL Global Development Group [Consultado abril 26, 2011]
- Tema I: Una metodología para el desarrollo de BD, Grupo BD Avanzadas [consultado abril 18, 2011]. Disponible en: http://ocw.uc3m.es
- Entrevista a... Lluís Codina, Julián Marquina, mayo 24, 2009 [consultado abril 18, 2011]. Disponible en: http://www.recbib.es
- http://www.mysql.org [Consultado mayo 6, 2011]
- http://www.postgresql.org [Consultado mayo 6, 2011]
- http://www.mongodb.org [Consultado mayo 6, 2011]
- Modelo de datos, Javier Fernández Rivera, www.aurea.es [Consultado mayo 7, 2011]
- Visual Paradigm, una herramienta de lo más útil, Eduardo Lion, [consultado mayo 13, 2011]. Disponible en: http://slion2000.blogspot.net
- Enterprise Architect-Herramienta de Diseño UML, [consultado mayo 13, 2011]. Disponible en: http://www.sparxsystems.com.ar
- Rational Rose Enterprise, [Consultado mayo 13, 2011]. Disponible en: http://www.rational.com.ar
- Herramientas CASE para BD, José del Valle, [Consultado mayo 13, 2011]. Disponible en: http://www.monografias.com
- Descargar Navicat for Mysql gratis, [Consultado mayo 13, 2011]. Disponible en: http://downloads.phpnuke.org

- Descargar EMS PostgreSQL Manager for PostgreSQL Lite gratis, [Consultado mayo 13, 2011]. Disponible en: http://descargar.traducegratis.com
- PGAdmin III, [Consultado mayo 13, 2011]. Disponible en: http://www.guia-ubuntu.org
- Oracle – Considerando la introducción de redundancia (Desnormalización), [Consultado mayo 18, 2011]. Disponible en: http://bdatos.wordpress.com
- Integridad de los datos, [consultado mayo 18, 2011]. Disponible en: http://msdn.microsoft.com
- Marlon Ruiz, Introducción a los Sistemas de Base de Datos, [Consultado mayo 18, 2011]. Disponible en: http://www.monografias.com
- Concepto de seguridad de Base de Datos, Prof. Lauro Soto, [consultado mayo 18, 2011]. Disponible en: http://www.mitecnologico.com
- **Luis Codina:** Profesor titular de la Universidad Pompeu Fabra (UPF) y director de la Unidad de Soporte a la Calidad y a la Innovación Docente (USQUID) de la Facultad de Periodismo. Docencia en Comunicación Audiovisual y Periodismo. Participación en Programas de Máster y de Doctorado del Departamento de Comunicación (UPF). Colaboración con programas de máster de la Universidad de Barcelona y de la Universidad Politécnica de Valencia.

GLOSARIO DE TÉRMINOS

Linux: Es un sistema operativo creado por Linus Torvald y el cual es totalmente libre. Su código fuente puede ser accedido, modificado y redistribuido por cualquiera bajo los términos de la GPL.

Tesauro: Derivado del neolatín que significa *tesoro*, se refiere al listado de palabras o términos empleados para representar conceptos. El término proviene del latín *thesaurus*, el cual tiene su origen del griego clásico θησαυρός (thesauros), *almacén*, *tesorería*. Como neologismo del latín es acuñado a principios de la década de 1820.

Taxonomía: La taxonomía (del griego ταξις, *taxis*, "ordenamiento", y νομος, *nomos*, "norma" o "regla") es, en su sentido más general, la ciencia de la clasificación. Habitualmente, se emplea el término para designar a la taxonomía biológica, la ciencia de ordenar a los organismos en un sistema de clasificación compuesto por una jerarquía de taxones anidados.

Ontología: La Ontología (Informática) es una herramienta para la organización, explotación y reutilización de información. Según Gil Leiva (2008, p. 224) desde la década de 2000 se están probando su utilidad en distintos procesos como la recuperación de información (Castells, Fernández y Vallet, 2007; Song et al. 2007), la clasificación de documentos (Yun, Guiyi y Jun, 2006; Weng, 2006) o para la extracción de información (Endres-Niggmeyer et al., 2006; Cimiano, Reyle y Saric, 2005).

Motor de búsqueda: Un motor de búsqueda, también conocido como buscador o *browser* es un sistema informático que busca archivos almacenados en servidores web gracias a su «*spider*» (o Web crawler). Un ejemplo son los buscadores de Internet (algunos buscan sólo en la Web pero otros buscan además en noticias, servicios como Gopher, FTP, entre otros mas.) cuando se pide información sobre algún tema. Las búsquedas se hacen con palabras clave o con árboles jerárquicos por temas; el resultado de la búsqueda es un listado de direcciones Web en los que se mencionan temas relacionados con las palabras clave buscadas.

Browser: Consultar “Motor de búsqueda”.

Tupla: una Tupla se define como una función finita que *mapea* (asocia unívocamente) los nombres con algunos valores. Su propósito es el mismo que se definió en las matemáticas.

Clase: construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten. Un objeto de una

determinada clase se denomina una instancia de la clase. La clase que contiene (y se utilizó para crear) esa instancia se puede considerar como del tipo de ese objeto

Software: equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

UML: Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

GNU GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios

Sun Microsystems: Sun Microsystems es una empresa informática recientemente adquirida por Oracle Corporation, anteriormente parte de Silicon Valley, fabricante de semiconductores y software.

Oracle Corporation: Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer, Oracle JDeveloper y Oracle Developer Suite. Su CEO actual es Larry Ellison. Hoy Oracle es el estándar de oro para la tecnología de base de datos y aplicaciones en las empresas en todo el mundo. La compañía es el proveedor líder mundial de software de gestión de información y la segunda mayor compañía de software independiente. La adquisición de Sun le da a Oracle un papel de liderazgo en el campo del software.

Array: En programación, una matriz o vector (llamados en inglés arrays) es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo, los elementos de la matriz. Desde el punto de vista lógico una matriz se puede ver como un conjunto de elementos ordenados en fila (o filas y columnas si tuviera dos dimensiones).

Trigger: Un trigger (o disparador) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos.

Apache: Apache Software Foundation (ASF) es una organización no lucrativa (en concreto, una fundación) creada para dar soporte a los proyectos de software bajo la denominación *Apache*, incluyendo el popular servidor HTTP Apache. La ASF se formó a partir del llamado *Grupo Apache* y fué registrada en Delaware (Estados Unidos), en junio de 1999.

Nosql: Nosql es un término usado en informática para agrupar una serie de almacenes de datos no relacionales que no proporcionan garantías ACID. Normalmente no tienen esquemas fijos de tablas ni sentencias "Join".

Google: Google Inc. es la empresa propietaria de la marca Google, cuyo principal producto es el motor de búsqueda del mismo nombre. Dicho motor es resultado de la tesis doctoral de Larry Page y Sergei Brin (dos estudiantes de doctorado en Ciencias de la Computación de la Universidad de Stanford) para mejorar las búsquedas en Internet. La coordinación y asesoramiento se debieron al mexicano Héctor García Molina, director por entonces del Laboratorio de Sistemas Computacionales de la misma Universidad de Stanford.[3] Partiendo del proyecto concluido, Page y Brin fundan Google Inc. el 4 de septiembre de 1998. Contaban con un servidor con 80 CPUs, y dos routers HP. Este motor de búsqueda superó al otro más popular de la época, AltaVista, que había sido creado en 1995.

Facebook: Facebook es un sitio web de redes sociales creado por Mark Zuckerberg y fundado por Eduardo Saverin, Chris Hughes, Dustin Moskovitz y Mark Zuckerberg. Originalmente era un sitio para estudiantes de la Universidad Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica.

Blog: Un blog, o en español también una *bitácora*, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. El nombre *bitácora* está basado en los cuadernos de bitácora, cuadernos de viaje que se utilizaban en los barcos para relatar el desarrollo del viaje y que se guardaban en la bitácora. Aunque el nombre se ha popularizado en los últimos años a raíz de su utilización en diferentes ámbitos, el cuaderno de trabajo o bitácora ha sido utilizado desde siempre.

Software propietario: El software privativo (también llamado propietario, de código cerrado o software no libre) es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo (esto último con o sin modificaciones).

Software libre: El software libre (en inglés *free software*, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término en el idioma inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la *Free Software Foundation*, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

Consulta: En base de datos, una consulta es el método para acceder a los datos en las bases de datos. Con las consultas se puede modificar, borrar, mostrar y agregar datos a una base de datos. Para esto se utiliza un lenguaje de consultas. El lenguaje de consultas más utilizado en bases de datos es el SQL (*Structured Query Language*).

JPG: (del inglés *Joint Photographic Experts Group*, Grupo Conjunto de Expertos en Fotografía), es el nombre de un comité de expertos que creó un estándar de compresión y codificación de archivos de imágenes fijas. Este comité fue integrado desde sus inicios por la fusión de varias agrupaciones en un intento de compartir y desarrollar su experiencia en la digitalización de imágenes. La ISO, tres años antes (abril de 1983), había iniciado sus investigaciones en el área.

PDF: (acrónimo del inglés *portable document format*, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

WORD: Microsoft Word es un software destinado al procesamiento de textos. Fue creado por la empresa Microsoft, y actualmente viene integrado en la *suite* ofimática Microsoft Office. Originalmente fue

desarrollado por Richard Brodie para el computador de IBM bajo sistema operativo DOS en 1983. Se crearon versiones posteriores para Apple Macintosh en 1984 y para Microsoft Windows en 1989, siendo para esta última plataforma las versiones más difundidas en la actualidad. Ha llegado a ser el procesador de texto más popular del mundo.

CPU: La unidad central de procesamiento o CPU (por el acrónimo en inglés de *central processing unit*), o simplemente el procesador o microprocesador, es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos. Los CPU proporcionan la característica fundamental de la computadora digital (la programabilidad) y son uno de los componentes necesarios encontrados en las computadoras de cualquier tiempo, junto con el almacenamiento primario y los dispositivos de entrada/salida. Se conoce como microprocesador el CPU que es manufacturado con circuitos integrados. Desde mediados de los años 1970, los microprocesadores de un solo chip han reemplazado casi totalmente todos los tipos de CPU, y hoy en día, el término "CPU" es aplicado usualmente a todos los microprocesadores.

RAM: La memoria de acceso aleatorio (en inglés: *random-access memory*, cuyo acrónimo es RAM) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

UPS: Un sistema de alimentación ininterrumpida, SAI (en inglés *Uninterruptible Power Supply*, *UPS*), es un dispositivo que gracias a sus baterías, puede proporcionar energía eléctrica tras un apagón a todos los dispositivos que tenga conectados. Otra de las funciones de los UPS es la de mejorar la calidad de la energía eléctrica que llega a las cargas, filtrando subidas y bajadas de tensión y eliminando armónicos de la red en el caso de usar corriente alterna.

SQL: El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *structured query language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella.

Caché: Un caché es un sistema especial de almacenamiento de alta velocidad. Puede ser tanto un área reservada de la memoria principal como un dispositivo de almacenamiento de alta velocidad independiente. Hay dos tipos de caché frecuentemente usados en las computadoras personales: memoria caché y caché de disco. Una memoria caché, llamada también a veces almacenamiento caché o RAM caché, es una parte de memoria RAM estática de alta velocidad (SRAM) más que la

lenta y barata RAM dinámica (DRAM) usada como memoria principal. La memoria caché es efectiva dado que los programas acceden una y otra vez a los mismos datos o instrucciones. Guardando esta información en SRAM, la computadora evita acceder a la lenta DRAM.

Sata: Serial ATA o SATA (acrónimo de *Serial Advanced Technology Attachment*) es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como puede ser el disco duro, lectores y regrabadores de CD/DVD/BR, Unidades de Estado Sólido u otros dispositivos de altas prestaciones que están siendo todavía desarrollados. Serial ATA sustituye a la tradicional Parallel ATA o P-ATA. SATA proporciona mayores velocidades, mejor aprovechamiento cuando hay varias unidades, mayor longitud del cable de transmisión de datos y capacidad para conectar unidades al instante, es decir, insertar el dispositivo sin tener que apagar el ordenador o que sufra un cortocircuito como con los viejos Molex

Driver: Un controlador de dispositivo, llamado normalmente controlador (en inglés, *device driver*) es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz -posiblemente estandarizada- para usarlo. Se puede esquematizar como un manual de instrucciones que le indica al sistema operativo, cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware.

Mb: El megabyte (MB) o megaocteto (Mo) es una unidad de medida de cantidad de datos informáticos. Es un múltiplo del byte u octeto, que equivale a 10^6 bytes.

Bitmap: Una imagen rasterizada, también llamada mapa de bits, imagen matricial o bitmap, es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada raster, que se puede visualizar en un monitor, papel u otro dispositivo de representación.

Join: La sentencia join en SQL permite combinar registros de dos o más tablas en una base de datos relacional. En el Lenguaje de Consultas Estructurado (SQL), hay tres tipos de *JOIN*: interno, externo, y cruzado.

Buffer: Un *buffer* (o búfer) en informática es un espacio de memoria, en el que se almacenan datos para evitar que el programa o recurso que los requiere, ya sea hardware o software, se quede sin datos durante una transferencia.