



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

JUSTIFICACIÓN DE UNA BASE
DE DATOS HISTÓRICA
PARA PROYECTOS DE SOFTWARE

TESIS QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

HUGO LUNA MERCHAND

DIRECTOR DE TESIS:

M.I. JORGE VALERIANO ASSEM



CIUDAD UNIVERSITARIA, MÉXICO D.F.

ABRIL 2011



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatoria

A las personas que merecen todo mi respeto y admiración, quienes han dado todo por mí, con los que siempre puedo contar, que me han acompañado en los peores y mejores momentos de mi vida, por los que me siento tan orgulloso, quienes me han inculcado tantos valores, les debo todo lo que soy.

Les dedico este trabajo como muestra de mi cariño y amor hacia ustedes.

Carmen y Hugo.

Agradecimientos:

A mis amados padres, agradezco todo su esfuerzo y perseverancia para permitirme dar este paso tan importante, todo lo que soy se los debo a ustedes.

Xochitl, mi amada esposa, porque a lo largo de este camino que hemos recorrido me has ayudado a levantarme y a crecer en todos los sentidos, has completado mi vida, hoy damos un gran paso en nuestras vidas, seguimos contigo.

Lisette, por todos los momentos de tu compañía, porque siempre estas cuando te necesito, por las grandes lecciones que me has dado, tu cariño, pero sobre todo, por fijar un estándar en mis metas.

Josa, Santi y Fati, que son una gran luz en mi vida, me han llenado de alegría y han cambiado mi forma de ver la vida.

Omar por tu amistad y el apoyo que me has mostrado.

A mis abuelos, que siempre me apoyaron a conseguir mis metas y confiaron en mí, hoy, con su gran ayuda, estoy cumpliendo esta meta tan importante.

A mi familia que siempre se ha preocupado por mí bienestar y me han apoyado.

A la UNAM por darme todas las armas necesarias para enfrentarme a la vida como persona y profesional, por teñir mi sangre de azul y mi piel de dorado.

Índice:

Objetivo	1
Introducción	3
1. Manejadores de bases de datos	5
1.1 Bases de datos	5
1.2 El modelo jerárquico	6
1.3 El modelo de red	7
1.4 El modelo Relacional	7
1.5 Bases de datos históricas	8
1.6 Sistemas de administración de bases de datos	9
1.7 Los RDBMS	10
1.7.1 MySQL	11
1.7.2 PostgreSQL	14
1.7.3 Microsoft SQL Server	16
1.7.4 Oracle	17
1.8 Justificación del manejador de bases de datos	20
1.9 Consultas generales sobre las bases de datos	20
1.9.1 Las opciones de tipo de campo	21
1.10 Implementación Casos de uso	22
1.10.1 Caso de uso 1	23
1.10.2 Caso de Uso 2	26
1.10.3 Caso de Uso 3	30
2. Justificación del uso de software	32
2.1 Java como lenguaje de programación	32
2.2 MySQL	36
3. Industria del Software 2009	42

3.1 Industria del software	42
3.1.1 La historia de la industria del software:	43
3.2 Industria del software en México	45
3.3 La industria del software a nivel mundial	47
4. Desarrollo del sistema	52
4.1 Requerimientos y justificación	52
4.2 Creación de la base de datos	52
4.3 Creación del proyecto	53
4.4 Diseño de la aplicación	55
4.5 Código de la aplicación	58
4.5.1 Manejo de eventos jButton	60
4.6 Contenedor Base de datos	67
4.6.1 Diseño del contenedor	69
4.6.2 Calendario	71
4.6.3 Generación de eventos	73
5. Proyectos similares	78
5.1 Perfil actual	78
5.2 Páginas comerciales	78
5.3 Páginas gubernamentales	80
5.3.1 Resultados de las búsquedas	81
5.4 Páginas históricas	84
5.5 Páginas del tipo legislativo	87
5.6 Páginas de transparencia gubernamental	91
5.6.1 Resultados	92
5.7 Páginas de divulgación científica	93
5.7.1 Resultados	95
5.8 Páginas de acceso a la información agrícola	96

5.8.1 Tipos de búsqueda	97
5.8.2 Resultados	102
5.9 Comparación entre los distintos proyectos	103
6. Infraestructura	105
6.1 Base de datos centralizada	105
6.1.1 Sistemas centralizados	106
6.2 Base de datos descentralizada	106
6.2.1 Procesamiento descentralizado de datos	107
6.3 Atributos de los modelos centralizados y descentralizados	107
6.4 Base De Datos Distribuidas	108
Beneficios a la sociedad	111
Conclusiones	113
Mesografía	115
Índice de imágenes	116
Índice de tablas	118
Bibliografía	119

Objetivo:

Para realizar un proyecto de software, de manera general, es indispensable contar con el conocimiento, entender la problemática, tener las tecnologías necesarias, realizar una adecuada planeación, realizar entregables, realizar pruebas y asegurar la calidad.

Reuniendo estas características, nos vemos ante la necesidad de justificar una base de datos que contendrá proyectos de software, proponiendo además una forma de implementar dicha base de manera local y a través de una interfaz de usuario. Con esta implementación se realiza una simulación de las bondades que pueda contener un proyecto futuro, por lo que ofrece un panorama general sobre los errores y aciertos que se pueden tener, para de esta forma en una iteración posterior, tener en cuenta la mayor cantidad de observaciones y mejorar en cada uno de los aspectos posibles.

Para dar una idea del futuro de este proyecto, basta con saber que la mayoría de las empresas tan solo almacenan alrededor del 20% de su información de manera estructurada, mientras que el 80% es información no estructurada, es decir, notas, facturas, planos, esquemas, conceptos, dibujos, talones, recibos, entre otros, es por este motivo, que una empresa consigue optimizar sus recursos. El objetivo primordial consiste en que la Facultad de Ingeniería cuente con un sistema de consulta único en cuanto a desarrollos de software, así como dar una nueva herramienta a los profesores y alumnos, que ayude en la práctica y teoría sobre la Ingeniería de Software.

Realizar todo un análisis como se realiza en las empresas de tecnología, en cuanto a proyectos de software, irá en mejora de la calidad educativa de la Facultad de Ingeniería, ya que los alumnos de la rama de ingeniería en software y en general en aquellas materias en las que se necesite realizar una administración de proyectos de software, tendrán la oportunidad de realizar el mismo proceso en una forma

más práctica que la que se tiene en la actualidad, en la que durante un semestre completo se trabaja en partes para ir realizando todo el proceso de un proyecto de software.

Desarrollar un software con el cual se puedan probar los conceptos más relevantes en cuanto a proyectos de software, dentro de una base de datos y justificar la creación de dicho trabajo, con las herramientas necesarias para tal fin.

Introducción:

En la actualidad, gran parte de la riqueza de los países primer mundistas, se debe sobre todo al fruto de las investigaciones realizadas en los distintos laboratorios y universidades, este fruto se traduce en productos de calidad y servicio para el público en general. En fechas recientes se ha podido comprobar cómo a pesar de haber atravesado por una crisis mundial, el sector tecnológico no se vio afectado, al menos no tanto como otros, sino que de hecho varias empresas vieron un gran crecimiento con respecto a años anteriores, en tanto otras compañías desaparecieron o fueron absorbidas por otras más grandes.

El caso de éxito más grande del que se puede hablar, es la empresa Apple, dedicada totalmente a la tecnología, esta empresa ha podido ver sus ganancias y acciones subir a grandes niveles, han creado productos de nicho y han generado en la población una necesidad por los mismos, su éxito se basa en dar productos de calidad con un soporte muy confiable, pero más allá de los productos físicos, están los servicios de software que se ofrecen, haciendo el círculo del atractivo completo, tanto hardware como software se conjuntan para crear la necesidad entre la población y tener así un enorme éxito alrededor del mundo.

La razón para enfocarse en este caso de éxito radica en lo bien organizada que se encuentra esta empresa, dirigida con puño de hierro, tiene una enorme cantidad de servidores en los que almacenan toda la información necesaria y más, para asegurar la satisfacción total de sus clientes. Estos servidores almacenan bases de datos en los que se encuentran varios de sus desarrollos para su software móvil iOS, o bien su sistema operativo de “escritorio” Mac OS X.

Gran parte de su éxito lo han logrado con la organización en sus datos, es de esta forma en la que nos percatamos de la importancia que es tener estructuras en las cuales almacenar la mayor cantidad de información posible, es decir, bases de datos en las cuales la búsqueda de la información sea lo más precisa y rápida posible.

Se necesitan de bases de datos que sean de utilidad en proyectos en desarrollo y futuros, de tal forma que toda la información contenida en las bases de datos se pueda o vuelva a reutilizar, lo que representa una gran ventaja en cuanto inversión de tiempo en el desarrollo del software. Es de esta forma que, el contar con un sistema de respaldo de cualquier base de datos que contenga datos esenciales para el buen funcionamiento de cualquier proyecto, resulta imprescindible para la mayoría de los sistemas.

1. Manejadores de bases de datos

1.1 Bases de datos

Es frecuente pensar en una base de datos como una variedad de productos y sistemas en los cuales podemos almacenar cualquier cosa, desde una colección de archivos hasta estructuras complejas para interfaces de usuarios.

En vez de pensar en el tipo de datos que se pueden contener dentro de esta, la cantidad de registros que se puede almacenar, como se utilizará o bien un número de elementos que intervienen en una base de datos, la definición resulta un tanto más simple, una base de datos es una colección de datos que se relacionan entre sí de alguna forma. Por ejemplo una aplicación médica requerirá de datos de los pacientes, su nombre, dirección, teléfono, entre otros. Aunque una base de datos es más que simplemente varios datos relacionados. Los datos deben estar organizados y clasificados en un formato estructurado conocido como metadato, el cual es un tipo de datos que describe los datos que serán almacenados. En otras palabras, el metadato indica la forma en la que serán almacenados los datos dentro de la base de datos. De esta forma, los datos junto con el metadato proveen de un entorno lógico que organiza la forma en la cual se podrán acceder y mantener los datos de manera eficiente.

Para entenderlo mejor podríamos pensar en una analogía con la sección amarilla, la cual contiene un gran número de datos (entre nombres, teléfonos, direcciones), la forma en la que se organiza, ya sea por orden alfabético, número telefónico, zonas geográficas, etcétera, será el metadato y en conjunto estos nos permiten encontrar de manera eficiente alguna persona dentro del libro.

Sin embargo no todas las estructuras en las bases de datos tienen un mismo formato. Con el pasar de los años han emergido distintos modelos de datos, de

estos modelos, los tres más comúnmente utilizados son el jerárquico, el de red y el relacional.

1.2 El modelo jerárquico

El modelo jerárquico se basa en los registros individuales y la relación padre-hijo que forman un árbol invertido. Este árbol crea una estructura jerárquica en la cual los datos se descomponen en categorías lógicas y las subcategorías que utilizan los registros para representar unidades lógicas de datos.

Un registro padre puede tener múltiples registros hijos, pero un registro hijo solo puede tener un registro padre. Esta es una estructura similar a la que se puede ver en la estructura de directorios de un explorador de archivos.

Después de su introducción, el modelo jerárquico alcanzó un gran éxito. Una de las implementaciones más populares fue el Sistema de Administración de Información de IBM, el cual fue introducido en 1960 y actualmente se utiliza ampliamente en las computadoras mainframes de IBM.

Sin embargo, a pesar de la popularidad alcanzada por el modelo jerárquico, este modelo ya no ajustaba a las necesidades de las nuevas aplicaciones. Esto debido a la inherente cualidad del modelo, esta es una estructura rígida de la organización padre-hijo vistas que resulta en un proceso de navegación engorroso que requiere del desarrollo de aplicaciones para poder navegar a través de los registros para lograr dar con la información necesaria. Los registros tienen que ser accedidos uno a la vez y de arriba para abajo a través de niveles jerárquicos, lo que hace que la modificación de la base de datos se torne muy laboriosa, lo que provocará un

proceso que consuma mucho tiempo. Además el modelo jerárquico no soporta las relaciones complejas entre los registros.

Incluso con las limitaciones del modelo jerárquico, en la actualidad existen diversos sistemas que ocupan este modelo para organizar sus datos y que han podido sobre llevar las limitantes del modelo, principalmente este modelo se ocupa para que un usuario pueda ir buscando un determinado archivo, llegando a él de modo directo observando los distintos niveles y las distintas iteraciones de los nodos.

1.3 El modelo de red

Para lidiar con las limitaciones del modelo jerárquico, surge el nuevo modelo de bases de datos, construido bajo el modelo jerárquico, emergió en los 70's. El modelo de red mejora al modelo jerárquico permitiendo a los registros entrar en relaciones de múltiples padres-múltiples hijos.

El modelo de red tiene aún muchas desventajas frente al modelo jerárquico, pero también provee mucha mayor flexibilidad al permitir a los programadores navegar a través de los registros. A pesar de la flexibilidad, los desarrolladores deben aún programar una aplicación para navegar por este modelo. Además de que cualquier cambio en la base de datos puede resultar en severas actualizaciones muy complicadas. Una base de datos debe ser bien planeada desde un principio, tomando en cuenta la navegación entre los distintos registros en el nivel de aplicación.

1.4 El modelo Relacional

Debido a las desventajas que presentaba el modelo jerárquico y el de red, un nuevo modelo comienza a ganar fama a principios de los 70's, y para finales de los

80's, surgió como el estándar de la próxima generación de bases de datos. El modelo de datos relacional representa una salida de las estructuras rígidas de los modelos de red y jerárquica. Las aplicaciones que accedan a la base de datos descansan o confían en una implementación definida de las bases de datos, y la estructura de la base de datos debe ser fuertemente codificada dentro de un lenguaje de programación. Si la base de datos cambia, la aplicación debe cambiar.

Sin embargo una base de datos relacional es independiente de la aplicación. Es posible modificar el diseño de la base de datos sin afectar la aplicación, debido a que el modelo relacional reemplaza la relación padre-hijo con una estructura basada en filas y columnas que forman tablas de datos relacionados. Como resultado, se pueden definir relaciones complejas entre las tablas, sin las restricciones de los modelos anteriores.

1.5 Bases de datos históricas

Las bases de datos históricas consisten en el almacenamiento de información que por lo general tiene más de diez años de antigüedad a partir de que se inicia su almacenamiento, principalmente se intenta almacenar documentos que no están digitalizados, para mantener un acervo informativo que contenga la mayor cantidad de documentos posibles y que se mantenga fiel a los documentos originales o bien que mantenga en esencia la misma información que estos. También es válida la información que se encuentra digitalizada, pero que no esté organizada de ninguna forma, más que el almacenamiento común, es así como se va recopilando la información y almacenando en una base de datos, para que se puedan realizar diversas consultas de manera casi inmediata y mantener los datos bien organizados.

1.6 Sistemas de administración de bases de datos

La mayoría de las bases de datos reposan en un sistema de administración de bases de datos para administrar los datos almacenados dentro del sistema de base de datos y hacer los datos disponibles para los usuarios quienes necesitan acceder a tipos específicos de información. Un DBMS (Sistemas de administración de bases de datos) está hecho de un completo arreglo de herramientas cliente-servidor que ayudan con varias tareas administrativas relacionadas con los datos. Por ejemplo, la mayoría de los DBMS proveen de algún tipo de herramienta que permite al cliente interactuar directamente con los datos almacenados en la base de datos.

Un DBMS debe almacenar datos y permitir que estos sean recuperados y modificados de manera que se protejan los datos contra operaciones que pudieran causar inconsistencias en la base o corromper los datos. Aunque por lo general la mayoría de los sistemas proveen muchas más capacidades. En general la mayoría de los DBMS completos proveen los siguientes tipos de funcionalidad:

- ✓ Administración de almacenamiento
- ✓ Mantenimiento de la seguridad
- ✓ Mantenimiento de los metadatos
- ✓ Administración de las transacciones
- ✓ Apoyo en la conectividad
- ✓ Optimización del rendimiento
- ✓ Proveer mecanismos de respaldo y recuperación
- ✓ Procesar peticiones de de recuperación y modificaciones

Estas características se extienden a cualquier DBMS, pero existen muchos más con funciones variadas y de naturaleza específica, para saber las funciones particulares provistas por algún DBMS en particular, hay que referirse a la documentación del mismo.

1.7 Los RDBMS

Al evolucionar los modelos de bases de datos, también lo hacen los productos DBMS que soportan varios tipos de bases de datos. No es de sorprender, entonces, que si existen DBMS, también existan los RDBMS. MySQL es un sistema de este tipo, como lo son Oracle, DB2, SQL Server y PostgreSQL. Estos productos, como cualquier DBMS, permiten acceder y manipular datos dentro de las bases de datos, protegerlos de la corrupción e inconsistencias, y mantener los metadatos necesarios para definir donde los datos que serán almacenados. La diferencia primaria entre los DBMS y los RDBMS es que estos últimos son específicos de las bases de datos relacionales. Soportan no solo el almacenaje de datos en estructuras de tablas, sino también las relaciones entre estas tablas.

1.7.1 MySQL

Surgiendo como uno de los más grandes jugadores en el mercado de los RDBMS está MySQL. Que como otros productos RDBMS, MySQL provee un amplio conjunto de características que soportan un ambiente seguro para almacenar, mantener y acceder a los datos. MySQL es rápido, confiable y una alternativa escalable de los muchos RDBMS comerciales que existen en la actualidad. A continuación se mencionan de manera general algunas de las características que se encuentran en MySQL:

- ❖ **Escalabilidad:** MySQL puede manejar grandes bases de datos, lo cual se ha demostrado con sus implementaciones en organizaciones como Yahoo!, Cox Communications, Google, Cisco, Texas Instruments, UPS, Sabre Holdings, HP y la prensa asociada. Incluso en la NASA y en los censos de Estados Unidos se han implementado soluciones MySQL. De acuerdo a la documentación MySQL, algunas de las soluciones empleadas por MySQL AB, la compañía creadora de MySQL, contiene más de 50 millones de registros, y algunos usuarios de MySQL han reportado que sus bases de datos contienen 60,00 tablas y 5 mil millones de columnas.

- ❖ **Portabilidad:** MySQL corre sobre una variedad de sistemas operativos, incluyendo Unix, Linux, Windows, QS/2, Solaris y MacOS, MySQL puede también correr sobre diferentes arquitecturas, desde las PC de escritorio hasta los grandes Mainframes.

- ❖ **Conectividad:** MySQL está totalmente orientado a las redes, soporta sockets TCP/IP, sockets Unix y las llamadas pipes. En adición, MySQL puede ser

- ❖ acceso desde cualquier lugar en internet, y múltiples usuarios pueden acceder a las bases de datos MySQL simultáneamente. MySQL además provee una gran variedad de interfaces para distintas aplicaciones de programación (APIs) para soportar la conectividad desde distintas aplicaciones escritas en lenguajes tales como C, C++, Perl, PHP, Java y Python.

- ❖ **Seguridad:** MySQL incluye un poderoso sistema de control de acceso a los datos. El sistema utiliza una estructura basada en el anfitrión(host) y el usuario que controla quien puede acceder a la información específica y el nivel de acceso a esa información. MySQL también soporta el protocolo de capa segura de sockets(SSL) para poder permitir conexiones encriptadas.

- ❖ **Velocidad:** MySQL fue desarrollado con la velocidad en mente. El monto de tiempo que toma a las bases de datos MySQL responder una petición de datos es tan rápido o más rápido que muchos de los otros RDBMS comerciales. El sitio

- ❖ **Facilidad de uso:** MySQL es fácil de instalar e implementar. Un usuario puede tener una instalación MySQL lista y corriendo, minutos después de descargar los archivos. Incluso en un nivel administrativo, MySQL es relativamente fácil de optimizar, especialmente comparado con otros productos RDBMS-

- ❖ **Código de fuente abierta:** MySQL hace que el código fuente de MySQL esté disponible para cualquier persona para descargarlo y ocuparlo. La filosofía

- ❖ de código fuente abierto permite a una audiencia global participar en la revisión, pruebas y desarrollo del código.

Como se puede observar MySQL es un RDBMS rápido y confiable que además implementa las ventajas y flexibilidad de los códigos de fuente abierta, es fácil de instalar e implementar, es gratuito y puede ser accesado desde cualquier lugar vía internet.

1.7.2 PostgreSQL

Es un DBMS que incorpora el modelo relacional para sus bases de datos y que se basa en el lenguaje estándar SQL. PostgreSQL ha mostrado ser bastante capaz y confiable, tiene buenas características de rendimiento. Es un manejador multiplataformas, nativamente corre en UNIX, pero es capaz de correr en sistemas como Linux, freeBSD, y Mac OS X, también funciona en sistemas Windows NT/2000/2003 Server, o incluso en sistemas Windows XP. Además utiliza un código de fuente libre.

PostgreSQL puede ser comparado favorablemente contra otros DBMS, pues contiene las mismas características que los demás DBMS comerciales, además de algunos extras que no se encontrarán en otros lados.

Las características de PostgreSQL incluyen:

- Transacciones
- Subselecciones
- Vistas
- Llaves externas con integridad referencial
- Bloqueo sofisticado
- Tipos de usuarios definidos
- Herencia
- Reglas
- Control de concurrencia de múltiples versiones

Desde la versión 6.5, PostgreSQL se ha vuelto bastante estable, con cada gran serie de pruebas de regresión para asegurar una estabilidad muy superior en cada lanzamiento. A partir del lanzamiento 7.x se ha llegado a un acercamiento mayor que en ningún otro hacia lo que es el lenguaje SQL92 y una restricción en el tamaño

de las filas que fue removido. En lanzamiento de la versión 8 se han agregado características tales como:

- Versión nativa de Microsoft Windows
- Espacios de la tabla
- Habilidad para alterar los tipos de columna
- Recuperación en tiempo de punto

PostgreSQL ha demostrado su confiabilidad en el uso. Cada lanzamiento ha sido controlado muy cuidadosamente, y los lanzamientos beta han sido sujetos de prueba al menos una vez al mes. Con una comunidad de usuarios más grande, con acceso al código fuente, los errores y problemas en el funcionamiento son reparados muy rápidamente.

El rendimiento de PostgreSQL ha sido mejorado en cada lanzamiento, y las últimas pruebas de rendimiento contra otras marcas, muestran que, en algunas circunstancias, se compara muy bien contra los productos comerciales.

Una de las fortalezas de PostgreSQL radica en su arquitectura, y es que es debido a esta, que se puede manejar en un entorno cliente/ servidor, el cual beneficia tanto a los desarrolladores como a los usuarios. El corazón de PostgreSQL radica en las instalaciones de bases de datos en los procesos de servidor. El cual corre en un solo servidor. Las aplicaciones que necesitan acceder a los datos almacenados dentro de la base de datos que requieren hacer vía el proceso de bases de datos. Los programas de tipo cliente no pueden acceder a los datos directamente, incluso si están corriendo en la misma máquina como el proceso servidor.

Esta separación entre el cliente y el servidor permiten que las aplicaciones sean de tipo distribuidos. Se puede utilizar la red para poder realizar la separación de los

clientes de tu servidor y desarrollar aplicaciones de cliente en un entorno que se ajuste a las necesidades de los usuarios. Por ejemplo, se puede implementar la base de datos en UNIX y crear programas cliente que corren sobre Microsoft Windows.

Con PostgreSQL, puedes acceder a tus datos en formas distintas:

Usando una línea de comandos para ejecutar enunciados SQL.

Montar SQL directamente sobre tu aplicación.

Usar llamadas de funciones para preparar y ejecutar enunciado SQL, examinar el ajuste de resultados, y desarrollar actualizaciones de una gran variedad de lenguajes de programación diferentes.

Acceder a la base de datos en PostgreSQL indirectamente utilizando como ODBC o el estándar JDBC, o bien utilizar una biblioteca estándar como PERL DBI.

1.7.3 Microsoft SQL Server

El primer lanzamiento de SQL Server ocurrió en 1989, fue un evento no muy notable en las bases de datos, las demás DBM eran superiores a este.

Microsoft SQL Server 2000 fue, en contraste, el punto de partida o rival a vencer para los DBMS, al final de la década estaba listo para además de su desarrollo, dejar fuera del mercado a numerosos DBMS, considerando sus numerosas características:

Máximo tamaño de sus bases de datos de 1 000 000 de terabytes. Como ejemplo, se podrían almacenar 100 megas de cada mujer, hombre, niño y perro en el planeta en un simple servidor de bases de datos SQL Server.

Hasta 16 instancias simultaneas de SQL Server corriendo en una sola computadora.

Soporta hasta 32 procesadores corriendo sobre una sola instancia.

Soporta hasta arriba de 64 gb en RAM de memoria física.

Otras características son:

- Construido con soporte para Lenguaje extensible de marcas XML
- Vistas indexadas
- Integridad de cascada referencial
- Capacidad mejorada de solicitudes distribuidas
- Soporte de servicios en análisis de minería de datos

1.7.4 Oracle

El servidor Oracle tiene todas las características de un RDBMS y que tiene un soporte amplio para entornos sofisticados cliente/servidor. Muchas de las características internas de Oracle están diseñadas para proveer una alta disponibilidad, máximo rendimiento, seguridad y un uso eficiente de los recursos del cliente. Aunque estas características son arquitectónicamente importantes para un servidor de base de datos, Oracle también incluye características basadas en el lenguaje que aceleran el desarrollo y mejoran el rendimiento del lado del servidor.

- **Lenguaje PL/SQL:** Un gran componente de Oracle es su máquina de procesamiento (Lenguaje de Procedimientos). PL/SQL está diseñado específicamente para procesos clientes/servidor en los que se activa un programa para bloquear la lógica que contiene la aplicación así como los enunciados que serán enviados al servidor en una sola petición.

- **Procedimientos almacenados:** Oracle permite la capacidad de almacenar bloques de PL/SQL como objetos dentro de la base de datos en forma de procedimientos almacenados, funciones, y paquetes de la base de datos. Las porciones lógicas de la aplicación, especialmente aquellas que requieren acceso a la base de datos, pueden residir en donde son procesadas (en el servidor). Usar procedimientos almacenados incrementa la eficiencia de los sistemas cliente/servidor significativamente.

- **Activadores de la base de datos:** Los activadores de la base de datos reensamblan los procedimientos almacenados que residen en los bloques PL/SQL de la base de datos; la diferencia entre los dos radica en que los activadores son disparados automáticamente por el kernel de la RDBMS en respuesta a que se cumpla un evento del tiempo (como alguna operación update, delete o insert)

- **Integridad declarativa:** Cuando se define una tabla en Oracle, se puede incluir una restricción de integridad como parte de la definición de la tabla. Las restricciones son forzadas por el servidor cuando se insertan, actualizan o borran registros. En adición a las restricciones integrales referenciales que fuerzan las relaciones entre las llaves primarias y foráneas, también se pueden definir las propias restricciones del usuario para controlar los valores del dominio de las columnas individuales de la tabla.

- **Funciones definidas por el usuario:** También se encontrarán bloques PL/SQL de funciones definidas por el usuario. Estas son similares a los procedimientos almacenados y también reducen el monto de codificación de la porción del cliente en la aplicación. Estas funciones no solo se pueden

- Llamar desde PL/SQL , sino que también se pueden extender al set estándar de las funciones Oracle SQL. Se pueden colocar funciones definidas por el usuario dentro de sentencias SQL justo como cualquier otra función de Oracle SQL.

1.8 Justificación del manejador de bases de datos

En nuestro caso utilizaremos el manejador de base de datos MySQL, dado que es un manejador del tipo fuente abierta, es decir gratuito y que permite la modificación al código. Ofrece las características generales que cualquier manejador comercial y es un manejador robusto para una gran cantidad de información como ya se mencionó en la introducción, así mismo ofrece gran seguridad y un buen soporte por parte de la comunidad MySQL, por lo que nos será de gran ayuda, es muy fácil de instalar así como de usar.

1.9 Consultas generales sobre las bases de datos

Algunas de las características que utilizaremos de este manejador son:

Mostrar las bases de datos:

```
show databases;
```

Mostrar las bases de datos seleccionadas:

```
select databases();
```

Mostrar las tablas que contiene una base de datos:

```
show tables;
```

Seleccionar una base de datos:

```
use nombre_base;
```

Describir la estructura de campos de una tabla:

```
describe nombre_tabla;
```

Crear una base de datos:

```
create database nombre_base;
```

Creación de una tabla:

```
create [temporary] table [if no exists] nombre_tabla (  
  nombre_campo 1 tipo 2 opciones 3 cláusulas ,  
  " " ,  
  " " ,  
  [último campo] );
```

[temporary] --> la tabla existirá mientras exista la conexión con el cliente actual o hasta que se emita la instrucción drop table.

[if no exist] --> si existe la tabla no se crea una nueva.

1.9.1 Las opciones de tipo de campo

tinyint --> 1 byte

smallint --> 2 byte

mediumint --> 3 byte

int --> 4 byte

bigint --> 8 byte

float --> 4 byte

double --> 8 byte

decimal --> variable

char(n) --> cadena de caracteres de longitud fija

varchar(n) --> cadena de caracteres de longitud variables

tinyblob --> objeto binario largo (muy pequeño)

blob --> objeto binario largo (pequeño)

mediumblob --> objeto binario largo (medio)
longblob --> objeto binario largo (grande)
tinytext --> cadena de texto muy pequeña
text --> cadena de texto pequeña
mediumtext --> cadena de texto media
longtext --> cadena de texto larga
enum --> una enumeración
set --> un conjunto
date --> valor fecha (aaaa-mm-dd)
time --> valor de hora (hh-mm-ss)
datetime --> valor de fecha y hora
timestamp --> valor de lapso de tiempo (aaaammddhhmss)
year --> valor de año

1.10 Implementación Casos de uso

Los actores que se utilizarán a lo largo del desarrollo de los distintos modelos de diseño serán aquellos mencionados en los casos de uso del trabajo anterior, los cuales son:

AC-1 Gestor Superior: Este es el encargado de los aspectos de negocios que tendrán una significancia para el proyecto, sus decisiones afectarán a todo el proyecto, su viabilidad, requisitos, necesidades, identificación de procesos, elaboración de documentación, diseño de flujo de datos, archivos, tareas de diseño.

AC-2 Gestor de proyecto: Conformar, motiva, organiza y controla al equipo de trabajo, además coordina y dirige la totalidad del proyecto.

AC-3 Profesional: Es el encargado de proveer el conocimiento técnico necesario para la ingeniería de un producto o aplicación. Puede especificar los requisitos para la ingeniería de software y otros elementos que tienen una menor influencia en el proyecto.

1.10.1 Caso de uso 1 (tabla CU-01)

ID:	CU-01
Nombre:	Configurar la BD
Descripción:	Este caso de uso permite establecer el ID y el PWD del superusuario, decidir la dirección donde se guarda la BD y el establecimiento de los datos del superusuario.
Evento que inicia el caso de uso:	Ejecución del programa.
Actores:	Gestor superior.
Precondiciones:	<ol style="list-style-type: none"> 1. Conocer el ID, PWD y ubicación de la BD. 2. Conocer la configuración a establecer. 3. Conocer quienes pueden acceder a la BD.
Poscondiciones:	<ol style="list-style-type: none"> 1. Creación del ID y el PWD del superusuario. 2. Conocimiento de la ubicación de la BD.
Flujo normal:	<ol style="list-style-type: none"> 1. Se establece el ID de la BD. 2. Se establece el PWD de la BD. 3. Se establece el nombre y la ubicación de acceso de la BD.
Flujo alternativo 1:	<ol style="list-style-type: none"> 1. Se elige eliminar la BD. 2. Se elimina la BD.
Excepciones:	(No aplica)
Incluye:	(No aplica)
Suposiciones:	(No aplica)
Notas:	(No aplica)

Fig.1.1 Tabla CU-01

Configurar la base de datos

En este caso de uso, lo que se pretende es establecer lo que será el nombre y contraseña del usuario maestro o super usuario.

Para la siguiente acción se asume que el servidor MySQL se instalará por defecto con las opciones clásicas, es decir, una instalación por default en archivos de programa, sin especificar una ruta distinta

Lo que se necesita para poder realizar esta acción es lo siguiente:

A partir del directorio en que se haya realizado la instalación de MySQL se deberá acceder a la base de datos de mediante los siguientes comandos (Utilizando el símbolo del sistema):

```
C:\cd program files
```

```
C:\Program Files\cd MySQL
```

```
C:\Program Files\MySQL\cd mysql server 5.1
```

```
C:\Program Files\MySQL\ MySQL Server 5.1\cd bin
```

```
C:\Program Files\MySQL\ MySQL Server 5.1\bin\mysql -u root -p
```

El password que aparece es por default nulo, esto es para que nosotros lo podamos configurar de acuerdo a nuestras necesidades.

En este caso para crear la contraseña para el usuario root, tecleamos la instrucción:

```
Mysqldadmin -u root password NUEVACONTRASEÑA
```

En donde NUEVACONTRASEÑA será la palabra que utilicemos para entrar a MySQL, sugerimos utilizar "unamfi" como contraseña.

Ahora para crear un nuevo usuario con una nueva contraseña utilizamos:

```
CREATE USER 'usuario1' IDENTIFIED BY 'usuario1';
```

En donde la siguiente palabra de USER será el nuevo usuario y la sentencia que le sigue a la instrucción IDENTIFIED BY será la nueva contraseña, es decir, en esta única sentencia tenemos tanto el nombre del usuario como su contraseña, ahora solo falta controlar los privilegios de nuestro nuevo usuario, para esto ocupamos la sentencia:

```
GRANT ALL TO user@host identified by 'password';
```

```
GRANT ALL TO hugi@host identified by 'user';
```

Para lograr borrar una base de datos se ejecuta la siguiente sentencia:

```
DROP DATABASE nombredelabasededatos;
```

Con esto habremos removido de MySQL la base de datos en cuestión, que es lo que necesitamos en un determinado momento en que se abandone el flujo normal del sistema.

Es de hacer notar que se podría borrar al usuario root para tener únicamente a nuestro usuario, más por razones de seguridad es preferible contar con al menos dos usuarios y en este caso mantendremos al que acabamos de crear, así como el usuario que viene preestablecido.

1.10.2 Caso de Uso 2

ID:	CU-02
Nombre:	Dar generalidades del proyecto
Descripción:	Este caso de uso permite ingresar el nombre del proyecto y sus características esenciales, la declaración de los miembros del equipo y sus características, así como la asignación del gestor del proyecto.
Evento que inicia el caso de uso:	Alta de miembros del equipo
Actores:	Gestor superior.
Precondiciones:	<ol style="list-style-type: none"> 1. Conocer el nombre del proyecto. 2. Conocer la localización del plan del proyecto. 3. Conocer las características principales del proyecto. 4. Conocer el nombre de los miembros de equipo y sus características. 5. Conocer el nombre del gestor del proyecto.
Poscondiciones:	<ol style="list-style-type: none"> 1. Establecimiento del nombre del proyecto y sus características. 2. Establecimiento de los miembros del equipo y del gestor del proyecto, así como la definición de sus características.
Flujo normal:	<ol style="list-style-type: none"> 1. Se establece el nombre del proyecto. 2. Se establecen las características principales del proyecto: inicio y fin del proyecto, tiempo de duración estimado, tiempo de duración real, objetivos, objetivos no logrados, costo estimado del proyecto, costo real del proyecto. 3. Se establece la ubicación de las partes del plan del proyecto o se señala que todo está contenido en un solo documento y se da la ubicación. 4. Se define ID y PWD de cada miembro del staff de la empresa. 5. Se describe cada miembro staff: nombre, horas que trabaja, habilidades, experiencia, temporalidad y monto del pago. 6. Se asigna el proyecto para cada miembro del staff de la empresa y se elige que el gestor del proyecto de entre los miembros del equipo del proyecto.
Flujo alternativo 1:	<ol style="list-style-type: none"> 1. Se establece el nombre del proyecto. 2. Se establecen las características principales del proyecto: inicio y fin del proyecto, tiempo de duración estimado, tiempo de duración real, objetivos, objetivos no logrados, costo estimado del proyecto, costo real del proyecto. 3. Se establece la ubicación de las partes del plan del proyecto o se señala que todo está contenido en un solo documento y se da la ubicación. 4. Se describe cada miembro del staff de la empresa: nombre, horas que trabaja, habilidades, experiencia, temporalidad y monto del pago. 5. Se asigna el proyecto para cada miembro del staff de la empresa y se elige al gestor del proyecto como el gestor superior.
Flujo alternativo 2:	<ol style="list-style-type: none"> 1. Se escribe el nombre del proyecto. 2. Se elimina el proyecto o se modifican los datos del proyecto o del gestor del proyecto.
Excepciones:	(No aplica)
Incluye:	(No aplica)
Suposiciones:	(No aplica)
Notas:	El gestor superior y el gestor del proyecto pueden ser el mismo sólo se tiene que indicar.

Fig.1.2 Tabla CU-02

El caso de uso número 2 CU-02 consiste en que se den las características generales de los proyectos o proyecto a ingresar al sistema.

Para este caso lo que se pretende es que ya se cuenten con varias características del proyecto, se comenzará por ingresar ciertos datos que serán verificados por el sistema mediante los distintos tipos de datos de que dispone MySQL.

Para comenzar, el proyecto necesitará un nombre el cual será del tipo de dato VARCHAR con una longitud de 30, este será el nombre del proyecto.

Para este tipo se creará una tabla especial llamada **CU-02**, la cual controlará algunas características importantes, tales como fecha de inicio del proyecto, fecha del fin del proyecto, tiempo estimado de la duración del proyecto, objetivos, objetivos no logrados, costo estimado del proyecto y costo real del proyecto, así como el mismo nombre del proyecto, por lo tanto nuestra tabla tendría que ser de la siguiente forma:

cu2id	Nombre Proyecto	Fecha inicio	Fecha fin	Tiempo estimado	Duración	Objetivos	Objetivos no logrados	Costo estimado	Costo real

Tabla cu2id

Para generar esta tabla se introduce la siguiente sentencia en Mysql:

```
Create table CU02(cu2id int auto_increment, nombreProyecto varchar(50),
fechalnicio date, fechaFin date, tiempoEstimado time, duracion time,objetivos
varchar(250), objetivosNoLogrados varchar(250), costoEstimado smallint, costoReal
smallint, primary key (cu2id));
```

```
CREATE TABLE `test`.`hola` (
  `Usuaroinicial` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  `ApellidoPaterno` VARCHAR(45) NOT NULL,
```

```
`ApellidoMaterno` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`UsuarioInicial`)  
)  
ENGINE = InnoDB;
```

Se ingresará la ruta del plan del proyecto, dentro de la base de datos, que será un tipo de datos del tipo VARCHAR de longitud de 50, en esta se especifica la localización de los distintos archivos que se utilizarán dentro del proyecto, o bien, se reúnen todo en un solo documento y este es el que se ingresará a la base de datos.

Se da un nombre de usuario y su password para cada uno de los miembros del equipo.

Se realiza la tabla de datos para cada miembro del equipo, con sus características, tales como:

Nombre, número de horas que trabaja, habilidades, experiencia, temporalidad y monto de pago.

De acuerdo a estas características se añade un gestor o administrador de proyectos que será elegido del mismo equipo, de acuerdo a las habilidades y experiencia de los mismos.

```
CREATE USER 'usuario' IDENTIFIED BY 'usuario';
```

```
Create table Usuarios(usrid int auto_increment, nombreusr varchar(50), numhoras  
time, habilidades varchar(100), experiencia varchar(100), temporalidad time,  
montopago int, caracPrinc varchar(200), primary key (usrid));
```

En la tabla se registrarán las características principales del proyecto, que serán del tipo de dato VARCHAR con una longitud de 200.

Se introducirán los nombres de los miembros del equipo así como las características de estos, en este caso también se ocuparán datos del tipo VARCHAR con una longitud de 300, para así poder además seleccionar a los miembros que participarán en determinado proyecto.

Adicionalmente se agrega el nombre del gestor o administrador del proyecto con datos del tipo VARCHAR con longitud de 60.

Para el flujo alternativo se manejan las mismas funciones, con la excepción de que en este caso no se asignarán nombres de usuario y passwords para cada uno de los miembros.

1.10.3 Caso de Uso 3

ID:	CU-03
Nombre:	Establecer Actividades
Descripción:	Este caso de uso permite establecer las partidas en las cuales se divide el trabajo, sus responsables y su status.
Evento que inicia el caso de uso:	(No aplica)
Actores:	Gestor del proyecto.
Precondiciones:	<ol style="list-style-type: none"> 1. Haber llevado a cabo los CU-01 y CU-02. 2. Conocer las actividades a desarrollar de cada miembro del equipo del proyecto. Conocer la duración de la jornada laboral de cada integrante del equipo. 3. Conocer el inicio y término de cada actividad.
Poscondiciones:	<ol style="list-style-type: none"> 1. Establecer las actividades
Flujo normal:	<ol style="list-style-type: none"> 1. Se establecen las actividades y se declara el status de la actividad de acuerdo al nivel en la cual esté. 2. Se establecen las fechas de inicio y término de cada actividad. 3. Se selecciona al responsable principal y al de apoyo.
Flujo alternativo:	<ol style="list-style-type: none"> 1. Se elige modificar datos. 2. Se selecciona que información se va a eliminar o modificar. 3. Se cambian los datos referentes a las actividades y sus encargados o bien se elimina alguna tarea.
Excepciones:	(No aplica)
Incluye:	(No aplica)
Suposiciones:	(No aplica)
Notas:	(No aplica)

Fig.1.3 Tabla CU-03

En el caso de uso número tres CU-03 se permitirán establecer las partidas en las que se divide el trabajo, sus responsables y el estado del mismo.

En este actúa el gestor del proyecto.

Para poder llevar a cabo este caso se necesitan haber realizado los casos CU1 y CU2, saber cuáles son las actividades que desarrollan cada uno de los miembros del equipo, así como su jornada laboral y el inicio y término de cada actividad.

Al terminar este caso de uso se deberán de establecer las actividades a realizar.

Se lleva a cabo una tabla dentro del manejador de bases de datos, que tendrá la siguiente estructura:

idCu3	Actividades	Status	iniAct	finAct	resPrin	resApoy

Tabla idCu3

Create table CU3(idCu3 int auto_increment, actividades varchar(200), status varchar(20), iniAct date, finAct date, resPrin varchar(50), resApoy varchar(50));

En caso de que se requiera modificar alguno de los campos, como el tipo de Status, se utilizará el siguiente comando (encontrándonos dentro de la base de datos):

```
UPDATE cu3 status='Terminado' WHERE status='En proceso';
```

1. INSERT INTO TABLE (field1, field2) VALUES (value1, value2)

Los demás casos de uso que se tienen en el trabajo anterior responden más a las actividades teóricas y de organización para introducir proyectos de software, ya que se refiere a las métricas de software y a la elección de estas por parte de las personas que introducirán los datos en la base, es por esto que esos casos de uso no se utilizarán de manera explícita, ya que están bien indicados en el trabajo anterior.

2. Justificación del uso de software

Para la realización del proyecto de bases de datos históricas, se necesitará del uso de un lenguaje de programación y un manejador de bases de datos.

2.1 Java como lenguaje de programación

Para el lenguaje de programación se sugiere el uso del lenguaje "Java", el utilizar el lenguaje java es importante, ya que, en primer lugar el lenguaje de programación Java no depende del tipo de máquina o bien sistema operativo que se esté utilizando. Si bien es cierto que podría correr en casi cualquier máquina (como una commodore 64), para poder utilizar todo su potencial, se necesita de un sistema operativo que cuente con capacidades multihilos (como Unix, Windows 95, OS/2...etc), ya que en realidad Java es un lenguaje interpretado, esto en principio. Al compilar un programa Java, se genera un pseudocódigo definido por Sun, para una máquina genérica. Posteriormente, al correrse en la máquina, el software de ejecución de Java interpreta las instrucciones, emulando a la máquina genérica. Por supuesto que esto no es eficiente, por lo que la mayoría de los sistemas operativos y navegadores web, al ejecutar por primera vez el código, lo van compilando mediante un JIT (Just In Time compiler), de modo que al crear la segunda instancia del objeto, el código ya se está ejecutando específicamente para la máquina huésped.

Precisamente debido a esta característica es que podemos trasladar la aplicación tanto a una máquina como a otra o bien entre distintos sistemas operativos, sin que se tenga que hacer ningún tipo de modificación a la aplicación, esto siempre y cuando se utilicen las clases más universales del mismo lenguaje, es decir, no incluir clases propias para por ejemplo el sistema operativo MAC, las cuales

partirían con la filosofía de Java, ser universal y portable, este tipo de clases no son el único impedimento en Java, pues además pueden haber ciertas llamadas al sistema que pueden no ser compatibles con los otros sistemas operativos. Por esto, siempre se debe programar de la forma más genérica para hacer las aplicaciones lo más portables posibles.

Otra de las ventajas de utilizar el lenguaje de Java es que está muy difundido en todo el mundo, es uno de los lenguajes que más auge ha tenido y su difusión es de las más importantes, su cuota de mercado es de aproximadamente un 45%, la cual comparte con C#(45%), junto con algunos otros lenguajes que manejan una menor cuota.

Java forma parte del paradigma orientado a objetos, esto quiere decir que es un lenguaje creado para trabajar desde cero con objetos, no como algunos otros lenguajes, como c++ que fueron modificados para poder ocupar los objetos, en Java todo son objetos.

Un objeto puede ser visto como una pieza de software, la cual cumple con ciertas características, tales como:

Encapsulamiento, el objeto es auto contenido, es decir, la misma definición del objeto incluye tanto los datos que este usa (atributos) así como los procedimientos (métodos) que actúan sobre los mismos. En la programación orientada a objetos se definen clases y la forma en que estas interactúan es a través de mensajes. Cuando se crea un objeto de una clase específica, se dice que se crea una instancia de la clase, que resulta ser un objeto. Es decir, la clase "Autos" y un "Toyota Prius" es una instancia de la clase. Esto es una gran ventaja, ya que al existir los objetos, no hay ningún programa que lo modifique, por lo que en cierta forma permanece independiente del resto de la aplicación. Si el objeto necesita ser modificado, esto se realiza sin tener que tocar el resto de la aplicación, lo cual ahorra mucho tiempo

tanto de desarrollo como de debuggeo, además en Java no existen las variables globales.

La otra característica es la herencia, esto significa que se pueden ir creando otras clases a partir de otras ya realizadas, lo cual reduce la programación, puesto que las clases hijas obtendrán automáticamente los métodos de los padres. Lo que nos lleva a la reutilización del código.

Ofrece un gran respaldo en línea para problemas que pudieran surgir con el propio lenguaje o bien, tips y tutoriales sobre algunos temas comunes de programación y otros más específicos. Además están tanto los foros oficiales de ayuda, como los creados por los usuarios y basta con buscar java en la red, aparecerán miles de millones de páginas con mucha información, consejos y ejemplos sobre el lenguaje. Sun microsystems ofrece capacitación en línea de forma gratuita para universitarios cuyas escuelas estén incritos en la red Sun, esta capacitación consiste en iniciar a los alumnos en el lenguaje Java desde un principio, se hace con ejemplos, videos, presentaciones, cuestionarios y exámenes.

Java además no solo se limita a la programación de computadoras de escritorio, sino que está presente en más de 2600 millones de teléfonos celulares, el cual resulta ser uno de los mercados en mayor crecimiento, lo cual nos habla de la cantidad tan impresionante de mercado que java tiene, por lo que la programación móvil va al alza. Aunque programación móvil no solo es relativo a los celulares, también podemos incluir un gran número de dispositivos como son: auto estéreos, estéreos, televisiones, reproductores blu-rays, netbooks, algunos e-book readers como el Kindle de Amazon e incluso consolas de videojuegos como el Playstation 3.

Algunas características notables del lenguaje Java son:

- Robusto

- Gestiona memoria automáticamente
- No permite técnicas de programación inadecuadas
- Multihilos
- Cliente-servidor
- Mecanismos de seguridad incorporados
- Herramientas de documentación incorporadas

Java es robusto debido a que, por la forma en la cual está diseñado, no permite el manejo directo del hardware ni de la memoria (de hecho no se permite modificar los valores de los apuntadores, entre otros) de modo que se puede decir que es prácticamente imposible colgar un programa java, pues el interprete es siempre el que tiene el control.

Es precisamente el compilador el encargado de impedir que se hagan modificaciones o se genere código inseguro, como utilizar variables sin inicializar, modificar valores de apuntadores directamente, acceder a métodos o variables en forma incorrecta, utilizar herencia múltiple, etc. Además Java implementa mecanismos que limitan los recursos o el acceso a estos en las máquinas donde se ejecuta. Permite acceder a los archivos en forma remota, ejecutándose desde una red. Además cuenta con el JDK (Java Development Kit), que tiene muchas herramientas para facilitar la programación, así como generar prácticamente toda la documentación de un programa automáticamente en formato html.

Uno de los inconvenientes al utilizar java, si es que existe alguno, es el hecho de que para poder ejecutar cualquier aplicación realizada en este lenguaje, es necesario contar con la maquina virtual instalada en la máquina en la que se quiera ejecutar, la ventaja de esto, es que debido al gran crecimiento tanto de internet como del lenguaje Java, prácticamente todas las computadoras con acceso a

internet ya cuentan con este requisito, por lo cual no representa un inconveniente mayor. Más aún, este proyecto está pensado para ser ejecutado en tan solo algunas máquinas, es decir, en una primer etapa, no habrá explotación comercial que nos pueda restringir la instalación en la o las máquinas a utilizar. Además la instalación de la máquina virtual es muy sencilla y en sus versiones más recientes, tiene un tamaño en disco de aproximadamente 16mb.

2.2 MySQL

MySQL, como se vio en la introducción, ofrece una gran cantidad de ventajas, es el manejador de bases de datos que se propone para implementar el sistema, las razones principales son:

- Es un software libre que cuenta con el respaldo de Sun Microsystems y una gran comunidad en línea
- Maneja el lenguaje estándar SQL
- Se compara muy bien contra los principales manejadores comerciales
- Se puede manejar desde una consola de comandos, de forma muy eficiente
- Permite múltiples usuarios con contraseña y distintos niveles de manipulación.
- La conexión con java es muy sencilla y práctica, desde Java se pueden ejecutar comandos para MySQL
- Desde Java se pueden crear tablas y registros, con independencia del mismo programa en Java
- Los requerimientos son mínimos, por lo que se ejecuta bien en casi cualquier máquina
- Es un manejador multiplataforma (Windows, Mac OS, Linux, Solaris, etc.)
- El límite de los registros por tabla es casi infinito, así como las tablas.

- Existen varios IDEs para MySQL
- Los tiempos para mostrar los resultados de las consultas son muy bajos

Estos puntos son los principales a tomar en cuenta para la realización del proyecto, además se debe tener en cuenta que MySQL maneja varias características que son de vital importancia y que pueden servir como referencia y que bien justifican el hecho de utilizar un software tan importante como este, para futuras migraciones o actualizaciones es importante tener en cuenta todas las características que a continuación se mencionan:

1. Escalabilidad y flexibilidad

El servidor de bases de datos MySQL ofrece las mejores características en cuanto a escalabilidad, puesto que es capaz de manejar bases de datos embebidas o empotradas ocupando tan sólo 1MB de memoria en diversas aplicaciones para de esta forma ayudar con la portabilidad y rendimiento, además puede hacer funcionar grandes bases de datos (almacenes de datos) que contengan terabytes de información. La flexibilidad de plataforma es una característica muy importante de MySQL, ya que soporta distintas versiones de Linux, UNIX y Windows, entre otros. Y por tratarse de un manejador de bases de datos de tipo open source, MySQL permite realizar una personalización completa, ya que las personas o empresas interesadas en realizar cambios sobre el programa original, para mejorar ciertas características que encuentren convenientes para sus respectivos intereses, esto debido a que los programas del tipo “código abierto”, permiten realizar modificaciones al liberar el código fuente del programa para después poder ser recompilado y estar listo para trabajar en el con las nuevas modificaciones.

2. Alto rendimiento

Una arquitectura única de motores de bases de datos permite a los profesionales configurar el servidor MySQL para aplicaciones específicas, dando como resultado un rendimiento muy alto, en cuanto a los tiempos de espera para poder generar los resultados. MySQL puede cumplir con las expectativas de rendimiento de cualquier sistema, ya sea un sistema de procesamiento transaccional de alta velocidad, o un sitio web de gran volumen sirviendo a más de un billón de consultas diarias, MySQL ofrece un motor de búsqueda adecuada para sistemas críticos mediante herramientas de carga de alta velocidad, índices full-text y otros mecanismos de mejora del rendimiento, como se mencionó en el capítulo 1.

3. Alta disponibilidad

Solidez y disponibilidad constantes son características distintivas de MySQL, con clientes confiando en ellas para garantizar el uptime en todo momento MySQL ofrece una amplia variedad de soluciones de alta disponibilidad, desde replicación a servidores de cluster especializados, hasta el uso del esquema maestro esclavo, u otros esquemas, los cuales contribuirán a que se mantengan los servicios en alta disponibilidad.

4. Robusto soporte transaccional

MySQL ofrece uno de los motores de bases de datos transaccionales más potentes del mercado y el mejor del tipo open source. Las características incluyen un soporte completo de ACID (atómica, consistente, aislada, duradera), bloqueo a nivel de filas, posibilidad de transacciones distribuidas, y soporte de transacciones con múltiples versiones donde los lectores no bloquean a los escritores y viceversa También se asegura una integridad completa de los datos mediante integridad referencial, niveles de aislamiento de transacciones especializados, y detección de

deadlocks. Por lo que se mantiene como una de las grandes opciones a tomar en cuenta a la hora de elegir un manejador de bases de datos, con las mejores características que se ofrece al mercado.

5. Fortalezas en Web y Data Warehouse

MySQL es el estándar de-hecho para sitios web de gran tráfico por su motor de consultas de alto rendimiento, su posibilidad de insertar datos a gran velocidad, y un buen soporte para funciones web especializadas como las búsquedas fulltext. Estas mismas fortalezas también se aplican a entornos de almacenes de datos (data warehousing), donde MySQL escala hasta el rango de los terabytes tanto para un solo servidor, como para varios. Otras características como las tablas en memoria, índices B-tree y hash, y tablas comprimidas hasta un 80% hacen de MySQL una buena opción para aplicaciones web y de business intelligence, por lo cual en fechas recientes MySQL es de las opciones más utilizadas en entornos Web.

6. Fuerte protección de datos

Porque proteger los datos es el trabajo principal de los profesionales de bases de datos, MySQL ofrece características de seguridad que aseguran una protección absoluta de los datos. En cuanto a autenticación, MySQL ofrece potentes mecanismos para asegurar que sólo los usuarios autorizados tengan acceso al servidor. También se ofrece soporte SSH y SSL para asegurar conexiones seguras. Existe una estructura de privilegios que permite que los usuarios sólo puedan acceder a los datos que se les permite, así como potentes funciones de cifrado y descifrado para asegurarse de que los datos están protegidos. Finalmente, se ofrecen utilidades de backup o respaldo y recuperación por parte de MySQL y terceros, que permiten copias completas, tanto lógicas como físicas, así como recuperación point-in-time, por lo que la protección de los datos se encuentra

garantizada, además de que es muy sencillo el realizar los backups, por lo que no se requiere de una gran capacitación para que el usuario común pueda realizar copias a los datos más esenciales de las empresas o personales.

7. Desarrollo de aplicaciones completo

Uno de los motivos por los que MySQL es la base de datos open source más popular es porque ofrece un soporte completo para cualquier necesidad de desarrollo. En la base de datos se puede encontrar soporte para procedimientos almacenados, triggers, funciones, vistas, cursores, SQL estándar, y mucho más. Existen librerías para dar soporte a MySQL en aplicaciones empotradas. También se ofrecen drivers (ODBC, JDCBC, entre otros) que permiten que distintos tipos de aplicaciones puedan usar MySQL como gestor de bases de datos. No importa si se trata de PHP, Perl, Java, Visual Basic, o .NET, MySQL ofrece a los desarrolladores todo lo que necesitan para conseguir integrar este potente manejador a cualquiera de las aplicaciones que requieran, normalmente la conexión que requieren los lenguajes no lleva más de 5 líneas de código, es por esto, que MySQL es uno de los grandes jugadores en el sector, ya que tiene un uso muy sencillo para los desarrolladores.

8. Facilidad de gestión

MySQL ofrece posibilidades de instalación excepcionales, con un tiempo medio desde la descarga hasta completar la instalación de menos de quince minutos. Esto es cierto sin importar que la plataforma sea Windows, Linux, Macintosh, o UNIX. Una vez instalado, características de gestión automáticas como expansión automática del espacio, o los cambios dinámicos de configuración descargan parte del trabajo de los atareados administradores. MySQL también ofrece una completa colección de herramientas gráficas de gestión que permiten al Administrador de Bases de Datos (DBA) gestionar, controlar y resolver problemas en varios

servidores desde una misma estación de trabajo. Además, hay multitud de herramientas de terceros que gestionan tareas como el diseño de datos, administración, gestión de tareas y monitorización.

9. Open Source y soporte

Una de las razones por las que muchas empresas no se atreven a adoptar software open source, es porque creen que no podrán encontrar el tipo de soporte o servicios profesionales en los que confían con su software propietario actual. Las preguntas sobre indemnizaciones también aparecen. Estas preocupaciones pueden desaparecer con el servicio completo de soporte e indemnización disponibles. MySQL no es un proyecto típico Open Source ya que todo el software es propiedad de MySQL AB, lo que permite un modelo de coste y soporte que ofrece una combinación única entre la libertad del open source y la confianza de un software con soporte, lo cual es una de las razones por las que MySQL es de los manejadores preferidos en el mundo.

10. Costo Total de Propiedad menor

Al migrar aplicaciones actuales a MySQL, o usar MySQL para nuevos desarrollos, las empresas ahorran costos que muchas veces llegan a cifras de millones. Las empresas están descubriendo que, gracias al servidor MySQL y las arquitecturas scale-out que utilizan hardware económico, pueden alcanzar niveles sorprendentes de escalabilidad y rendimiento, y todo a un coste bastante menor que el de los sistemas propietarios. Además, la robustez y facilidad de mantenimiento de MySQL implican que los administradores no pierden el tiempo con problemas de rendimiento o disponibilidad, sino que pueden concentrarse en tareas de mayor impacto en el negocio, lo que contribuye a un enorme ahorro para las empresas.

3. Industria del Software 2009

El software es el aspecto no tangible de una computadora que es necesario para ejecutar alguna función. El software incluye los programas que son los encargados de instruirle a la computadora que hacer. Los archivos de configuración son información importante para el sistema que es almacenada, y los archivos de usuario son los que almacenan los resultados finales de las funciones computacionales. Con frecuencia varias capas del software se combinan para ejecutar tareas computacionales como por ejemplo la transferencia de los archivos en protocolo html mediante TCP/IP, o la operación del reproductor de medios en un sistema operativo.

El software de la computadora, o simplemente software es un término general utilizado para describir el rol en que los programas de computadora, procedimientos y la documentación juegan dentro de un sistema de cómputo.

3.1 Industria del software

La industria del software incluye negocios que envuelven el desarrollo, mantenimiento y la publicación de un software de computadora utilizando un modelo de negocios. La industria incluye servicios de software, tales como entrenamiento, documentación y consultoría. Además es una industria que se dedica a la investigación, el desarrollo, distribución y comercialización del software.

Hay muchos tipos de negocios en esta industria, las empresas más grandes y que tienen mejores ganancias son las que promocionan su software propietario de manera horizontal, tales como Microsoft, Apple, IBM, Adobe, Oracle, entre otras.

Otro software que se desarrolla de manera vertical y que va encaminado a un nicho muy particular del mercado es aquél software que se dirige a la economía por medio de las finanzas, cuidado de la salud, aseguradoras, ventas minoristas, manufactureras automotrices, entre otros. El gran trato consiste en la gran diversidad de software especializado que se produce para los diversos nichos existentes. Algunas otras compañías se dedican a contratar a los programadores para un producto de software único y particular para algún cliente o compañía, también se pueden dedicar a configurar y adaptar los programas de grandes compañías.

3.1.1 La historia de la industria del software:

La historia de la industria del software comenzó en la época de 1960, cuando las industrias y las universidades comenzaron a utilizar las computadoras y utilizar programas que no estaban incluidos en las mismas máquinas para realizar tareas específicas de cómputo. Muchos de estos programas fueron realizados en la casa de programadores de tiempo completo que eran parte del personal de las empresas. Algunos otros programas fueron distribuidos libremente entre los usuarios de alguna máquina en particular sin ningún cargo. Pero otros se basaron en la parte comercial y crearon las primeras firmas independientes de software que comenzaron en los estados unidos entre 1959 y 1960.

La empresa comenzó a volverse muy grande debido al gran crecimiento del personal para las computadoras a mediados de 1970, se creó un mercado muy creciente en lo que fue la industria de los videojuegos, las aplicaciones y las utilidades. Gradualmente esta industria comenzó a volverse un contrincante para

muchas otras. Uno de los primeros ponentes en esta industria fue Microsoft, una empresa creada por Bill Gates.

A principios del siglo veintiuno apareció un nuevo concepto en el modelo de negocios, el cual fue el software como un servicio (SaaS, por sus siglas en inglés). SaaS reduce los problemas de la piratería, ya que estos solamente pueden ser accedidos desde la web, y por definición ningún software es cargado en la máquina huésped del usuario.

Entre 1960 y 1970, solo existían las grandes computadoras que ocupaban varios cuartos, estas se encontraban en corporaciones de alto nivel, los usuarios solo tenían terminales en las que no se contaba con algún tipo de procesamiento.

Durante 1980 se desarrolla la computadora de escritorio, la cual permite la existencia del software personal de escritorio, el cual rápidamente fue propiciando la productividad de los individuos, utilizando software como hojas de cálculo y procesadores de texto. Comienzan a aparecer las redes y servidores, el compartimiento de archivos, impresoras, discos duros, aparecen aplicaciones como el correo electrónico y agendas electrónicas.

Durante los 90's hay un gran dominio de los servidores y las redes, comienza el boom del internet, en principio existen las páginas estáticas las cuales dan información rápidamente, pero comienzan a aparecer las páginas ricas en contenido, que son dinámicas así como interactivas. Las empresas comienzan a enfocarse en los usuarios finales, eliminando de esta forma a los intermediarios y maximizando las ganancias, es entonces cuando aparece y se desarrolla lo que es el comercio electrónico, consistente en ir eliminando a los intermediarios, negociar directamente con los usuarios finales, con negocios tales como empresa a empresa, empresa a usuario o usuario a usuario.

Durante la década del 2000, el internet se convierte en un recurso muy común y casi indispensable, aparecen las redes inalámbricas, los dispositivos celulares permiten una conexión directa al internet, aparecen las características más ricas en internet, videos, blogs, software para su uso en línea, música, buscadores, entre otros.

Anualmente la industria del software genera ganancias por más de 500 mil millones de dólares.

3.2 Industria del software en México

La industria del software en México está basada en los servicios, es decir, el software que se desarrolla en México, en su mayoría está basado en la adaptación de programas más grandes, que es a lo que se le llama software a la medida y se le conoce como servicios, ya que una vez que se termina con el proyecto, todo el software que se desarrolló, por lo general no se vuelve a utilizar, sobre todo porque al terminarse, se termina con la necesidad inicial de un software a la medida y es entonces cuando se pierde la capacidad de solventar una necesidad universal o bien darle un mejor uso a esta aplicación a la medida. El desarrollo del software también se basa en realizar desarrollos innovadores como sistemas operativos y aplicaciones, software que se vende por miles y millones, como los paquetes de aplicaciones, producen grandes cantidades de dinero y es de esta forma que el software es considerado como un bien.

En México el porcentaje destinado a la tecnología es de apenas un 0.7%, mientras que en países desarrollados el PIB para la tecnología es de hasta el 4%, lo cual deja a México en una franca desigualdad en contra de otros países y nos habla de la falta de inversión en el país para la industria de la tecnología.

Aunque se asegura que esta posición no es del todo mala, ya que nos habla de que al no tener una gran inversión por parte del gobierno, las industrias privadas tienen un amplio mercado en México, pues se puede explotar muy bien al no haber un gran desarrollo tecnológico. Según estimaciones por parte de los líderes de la industria (entre ellos IBM) y basados en estudios realizados, México será una de las cinco principales economías para el 2050, lo que lo vuelve muy atractivo para las empresas de IT (Tecnologías de la información). Y aunque para muchas empresas el principal problema con México a la hora de invertir es la corrupción, es un riesgo que se acepta debido en gran parte a que también, un estudio realizado muestra que el crecimiento que ocupa México lidera sobre otros 11 países de alto crecimiento desde hace dos años. Otra de las ventajas de invertir en México es que la penetración de las tecnologías no ha llegado más que a un 30%, esto debido a que alrededor del 70% de las PyMES se rehusa a utilizar la tecnología como un medio para encontrar nuevos planes de negocio y mejorar las condiciones de venta y distribución entre otros.

Así mismo lo logrado por México durante los últimos tres años en el “Indicador de la Sociedad de Información”(ISI), se ha volcado a la misma cifra que tenía durante el 2006, lo que significa un retroceso del 3.4% en el progreso logrado anteriormente. Los resultados del reciente reporte del ISI en Latinoamérica, en el mes de agosto, advierten que la posición de México ha bajado una posición y pasó a ocupar el cuarto lugar de la región justo detrás de Chile, Argentina y Perú.

De acuerdo a los resultados obtenidos durante este año y en parte provocado por la crisis mundial, es de esperarse que México tenga una baja en el porcentaje del ISI y no solo México, sino también todos los países evaluados, por lo cual resulta relativamente satisfactoria esta evaluación, México tendrá un repunte y comenzará

a subir este indicador hasta el segundo semestre del 2010, sin embargo la caída será moderada, esto según los estudios de la consultora Everis.

En los servicios de TIC (Tecnologías de la Información y Comunicación) se espera que los dominios de internet sean los que ayuden al crecimiento de forma positiva para esta rama y que crezcan en un 18%, además se estima que durante el segundo trimestre del 2009 ha habido una expansión en el número de computadoras, con un incremento del 13.9%(200 aparatos por cada mil personas). En cuanto al acceso a internet, también se ha registrado un incremento, esto debido en gran parte a la amplia oferta de los distintos proveedores, para el segundo trimestre del 2009 ha aumentado el número de usuarios de banda ancha (uno de cada cinco), aumentando en 2.7 puntos porcentuales las cifras del año pasado y duplicando en el mismo trimestre las cifras del 2005.

Para el mes de junio de 2010 se espera que las cifras para las tecnologías de la información y comunicación (TIC) presenten un aumento, esto debido a que las empresas del ramo recuperarán su nivel de ventas, además se espera que los usuarios adquieran nuevo equipo, impulsado por los nuevos sistemas operativos que verán la luz a finales del 2009, así como por la recuperación económica que se espera.

3.3 La industria del software a nivel mundial

A nivel mundial, la industria del software ha sufrido de los embates de la crisis económica que se ha vivido durante el 2009, y aunque el crecimiento de la industria no es perceptible, tampoco se trata de un crecimiento negativo, el crecimiento económico que se ha generado en lo que va del año es conocido como un crecimiento plano, es decir, prácticamente se mantiene en el mismo nivel que el

año pasado, pues su crecimiento fue de tan solo el 0.3% en comparación con el mismo periodo en el año pasado.

Los analistas se muestran positivos a que la crisis por la que atraviesa el mundo durante el 2009 se superará de mejor forma que en 2001 y 2002, sobre todo en la industria del software, se muestran confiados en que las estrategias tales como la mayor penetración en el mercado, la maduración, la confianza en las TI ayuden a salir adelante prácticamente toda la industria mundial del software, parte de esta recuperación se logrará mediante técnicas como la compra o actualización de nuevas licencias, soporte y mantenimiento del software, al venta del software como servicio, el código abierto o el outsourcing así como las nuevas tecnologías y nuevos sistemas operativos. Las empresas que se mantienen con un panorama favorable son aquellas que han conseguido sustentar una gran reputación y poseen un equilibrio balanceado en cuanto a su geografía de expansión por todo el mundo. Es debido a esto que varias empresas pueden arriesgarse en modificar el precio de sus productos para volverlos más competitivos en el mundo del software y de la misma forma alcanzar una audiencia todavía mayor que la lograda hasta ahora.

De una forma u otra todas las regiones del mundo se han visto afectadas debido a la economía, que ha influido en las previsiones de ventas y ganancias para este año, sin duda los países a los que más ha afectado son los países desarrollados, pero no son los únicos, los países en vías de desarrollo también han sentido el peso de la crisis ya que sus proporciones de crecimiento se han visto afectadas en forma negativa, esto se hace presente sobre todo al este de Europa, en donde las previsiones de crecimiento han sido bastante perturbadas.

De acuerdo a la consultora Forrester para la industria de TI, durante el 2009:

La industria mejorará por más de la media. Sus proyecciones auguran compras de productos por \$388 000 millones de dólares, lo mismo que en 2008, lo cual resulta en una mejoría en comparación con otros sectores del TI que declinarán.

La inversión en equipos de comunicación se contraerá. Compras de routers, switches, equipos de conferencia, y equipo de comunicaciones unificadas caerá a los \$353 000 millones de dólares con respecto a los \$364 000 millones de dólares en 2008.

La inversión en equipo de cómputo verá su más grande caída en el crecimiento que ha mantenido. Las compras de computadoras personales, servidores, dispositivos de almacenamientos y periféricos caerán un 4% a \$434 000 millones de dólares en 2009, de los \$450 000 millones de dólares en 2008.

Los servicios globales de TI y el outsourcing también tendrán una caída. Los gobiernos y las empresas comprarán \$484 000 millones de dólares de consultoría TI, sistemas de integración y servicios de outsourcing en 2009, 3% menos que en el 2008. Los servicios de outsourcing de TI tendrán un panorama un poco mejor que los servicios de consultoría y sistemas de integración, con la última caída de compras en la implementación e integración del software.

Además de las proyecciones para las TI provee datos sobre los 15 países que serán los mercados más grandes para bienes y servicios en la industria de TI. Los grandes países que hacen uso extensivamente de su tecnología y la han sabido aplicar a los ámbitos económicos nacionales son aquellos a los que se les conoce como “BRIC” Brasil, Rusia, India y China

Otro consultor, Gartner, a través de una encuesta realizada a mil CEO de distintas y grandes compañías de TI alrededor del mundo, revela que en lo que va del año y hasta terminarlo se seguirán reduciendo los presupuestos para la TI, por lo que, las

inversiones en cuanto las áreas tecnológicas se mantendrán prácticamente sin cambios, sin embargo, para el 2010, se tiene planeado aumentar los presupuestos para el software en torno al 1.53%.

En cuanto a las expectativas de gasto en el software para el 2010 se espera que en la región de Norte América haya una disminución del 2.06%, mientras que en Europa, el medio este y África (EMEA), haya un aumento del 0.45% en comparación con el 2009, por otra parte se espera que en Latinoamérica haya un incremento en el gasto en un 2.54%, mientras en Asia/Pacífico aumentará en un 4.34%, mostrando en general tendencias positivas para con este sector económico.

En la India, a pesar del año que vivimos en plena desaceleración económica, la industria del software es la industria con mayor crecimiento y que mayores ganancias aporta a esta nación, se espera que para el 2010 el software y los servicios de la industria crezcan en un 17% alcanzando ganancias por 71 000 millones de dólares.

A pesar de que la crisis mundial aun no ha terminado, para la industria del software y tecnología en China, no ha sido un mal año, ya que sus reportes de mitad de año han demostrado que, lejos de disminuir económicamente hablando, se mantienen en crecimiento, arrojando datos de aproximadamente un 20% de crecimiento con respecto al año anterior. Año tras año la tasa de crecimiento en este sector en China había registrado un crecimiento de aproximadamente 35%, y en especial durante este ciclo que ha sido tan retador, un crecimiento del 20% para la primera mitad del 2009 es un gran logro el que la industria del software y la tecnología mantengan este gran crecimiento y por sobre las otras industrias de ese país, la del Software representa un gran orgullo así como ganancias para China. Este

crecimiento puede significar 2 cosas, que la industria del Software tienen una gran demanda y fuerza vital, o bien, la sociedad le tiene confianza al desarrollo de la industria del software como servicio. A pesar de la crisis mundial, la industria del Software en china es la que más rápido crecimiento tiene entre las demás de ese país.

En China las ganancias propiciadas por el sector de la tecnología han llegado a los 62 500 millones de dólares en tan solo la primera mitad del año.

Se tiene previsto que para el año de 2010 la inversión mundial en Software sea mayor a los 5 700 000 millones de dólares

4. Desarrollo del sistema

4.1 Requerimientos y justificación

Necesitamos tener un IDE(Entorno de Desarrollo Integrado), para este proyecto recomendamos NetBeans 6.5 por ser de software libre y tener un apoyo por parte de Sun Microsystems, además de tener las librerías para realizar las conexiones con MySQL que será el manejador de bases de datos que se propone utilizar.

Como ya tenemos el planteamiento realizado en el trabajo anterior, únicamente cabe destacar algunas modificaciones de tipo funcional y ajustándolo a las herramientas utilizadas para el desarrollo.

En el trabajo anterior se ha propuesto la utilización de tecnologías PHP con MySQL, Linux CentOS y Samba. Ahora se propone el uso de Java con MySQL por ser ambas herramientas multiplataformas, lo cual facilitará su desarrollo en cualquier entorno o sistema operativo.

4.2 Creación de la base de datos

En principio se tiene que crear la base de datos que contendrá toda la estructura (tablas y relaciones entre los distintos datos).

Mediante el mysql command line client, al abrirlo pide una contraseña, para nuestro caso será nula es decir, con un solo enter estará listo el programa para comenzar a trabajar, iniciamos con la sentencia:

```
Create database TOOLBDH;
```

Con lo que se creará una base de datos llamada toolbdh como se propone en principio, con esta sentencia, únicamente se estará reservando un espacio en memoria para ir llenando la base de datos, es decir, la base recién creada, se encuentra vacía, por lo que ahora se creará la tabla que contendrá los registros, mediante las siguientes sentencias:

Use TOOLBDH;

Con lo que indicamos que estaremos trabajando sobre esta base de datos.

```
mysql> create table ingreso(nombreProyecto varchar(50), fechaInicio date,
fechaFin date, tiempoEstimado varchar(10), Duracion varchar(10), Objetivos
varchar(250), objetivosNoLogrados varchar(250), costoEstimado varchar(20),
costoReal varchar(20));
```

Con esta sentencia creamos la estructura que llevará nuestra tabla que contiene los registros, tales como: nombre del proyecto, fecha de inicio, fecha de finalización, duración, objetivos y costos, entre otros.

4.3 Creación del proyecto

Una vez con la base de datos creada, abrimos el IDE Netbeans, seleccionamos File->New Project-> y a continuación aparecerá una ventana como en la figura 4.1

Dentro de la categoría Java elegimos el tipo de proyecto Aplicación Java de Escritorio (Fig.4.1), damos clic en Siguiente.

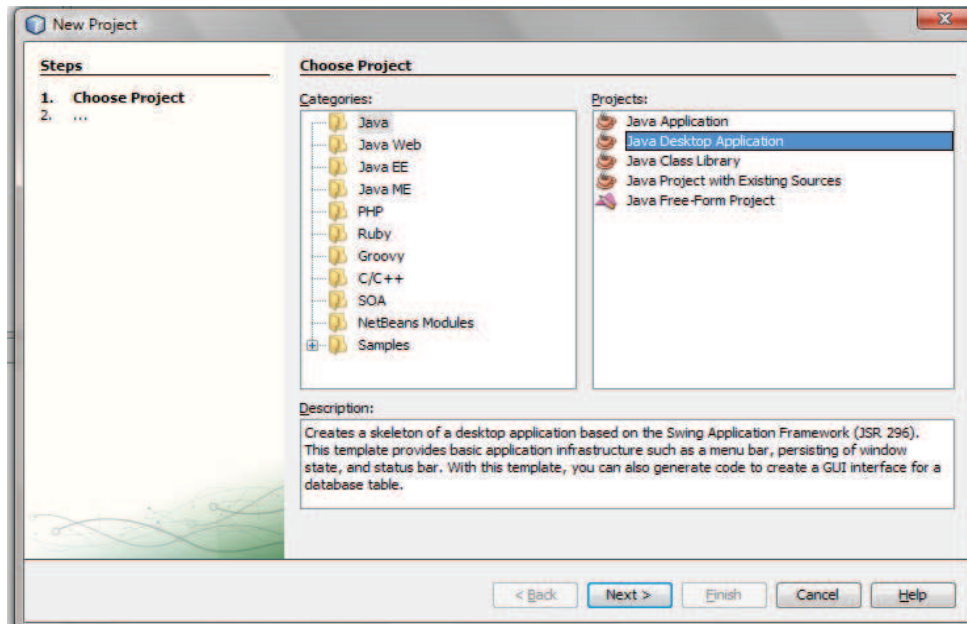


Fig.4.1 Selección de proyecto en Netbeans

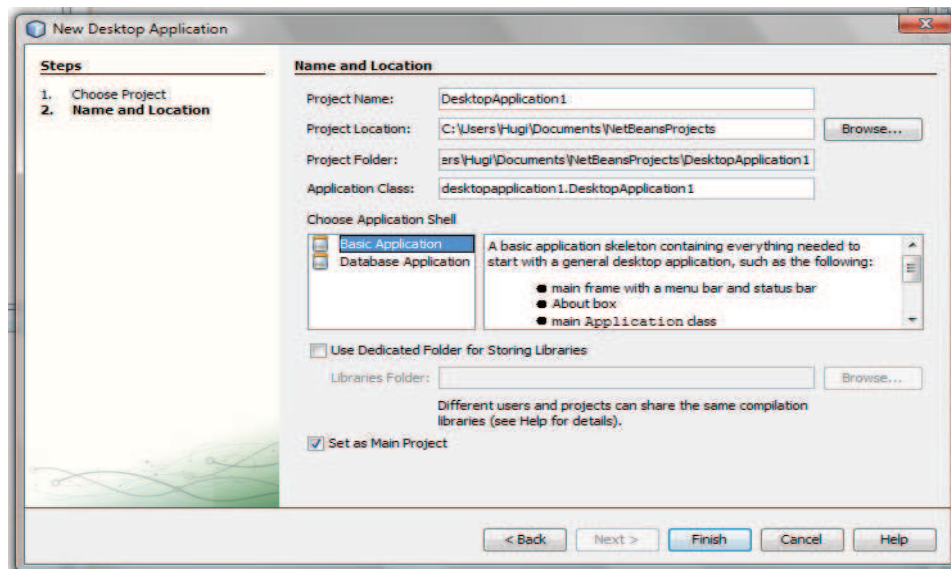


Fig.4.1.1 Nombre y localización del proyecto

En la nueva ventana (Fig.4.1.1) se introducirá el nombre de nuestro proyecto, así como la ubicación en el disco duro de nuestro sistema y para finalizar seleccionamos la opción de Aplicación Básica y damos clic en Finalizar.

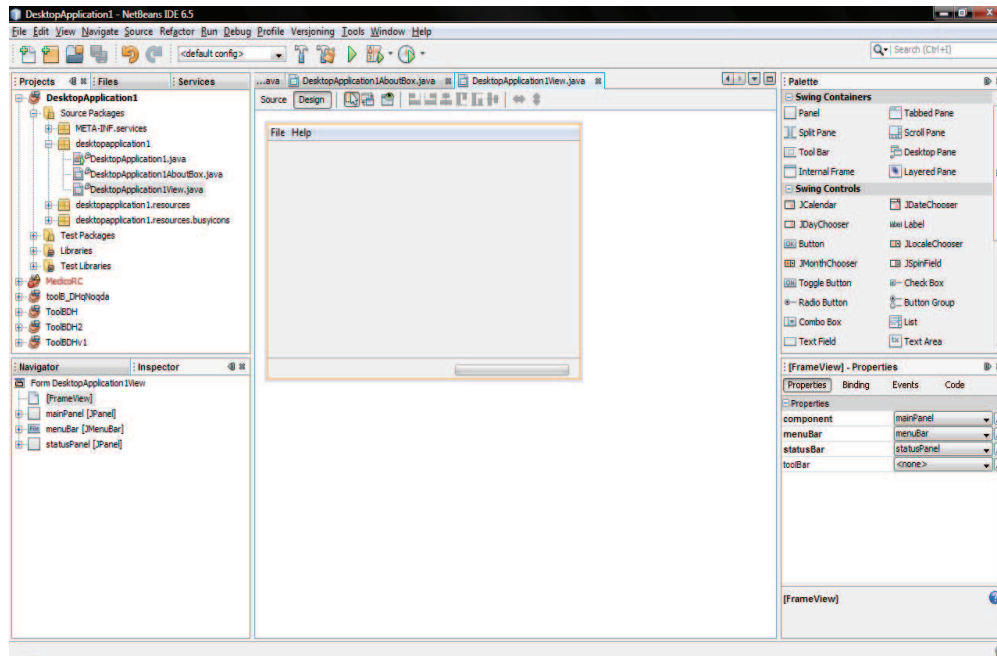


Fig.4.1.2 Presentación básica del proyecto

4.4 Diseño de la aplicación

Se nos muestra una ventana sin ninguna decoración (Fig.4.1.2), esta será el lienzo sobre el cual comenzaremos nuestro programa, ahora iremos implantando cada uno de los siguientes elementos, arrastrándolos, desde el Palette Manager(Lado superior derecho de la Fig.4.1.2):

jLabel “Usuario” acompañado por un jTextField

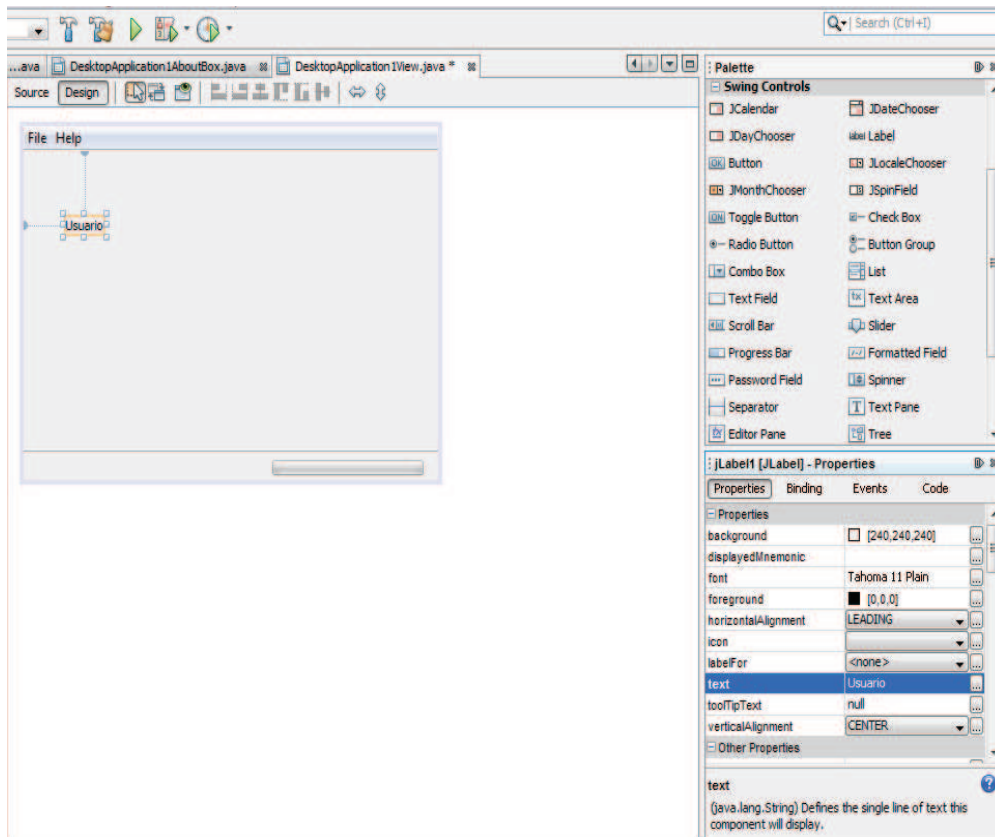


Fig.4.1.3 Propiedades de los controles Swing

Arrastramos desde el palette manager un JLabel (En swing controls dentro de palette) hasta la ventana sin decoración, una vez colocado, lo seleccionamos y en el menú de properties (Debajo del palette manager), buscamos por la propiedad text, aquí cambiamos el predeterminado por "Usuario", ahora de la misma manera, dentro del palette manager, buscamos y arrastramos a la ventana un JTextField, de tal forma que quede colocada junto a nuestro JLabel que acabamos de ubicar. Al igual que con la JLabel, dentro de las propiedades del JTextField, buscamos por text y borramos el contenido para que el interior de este quede en blanco.

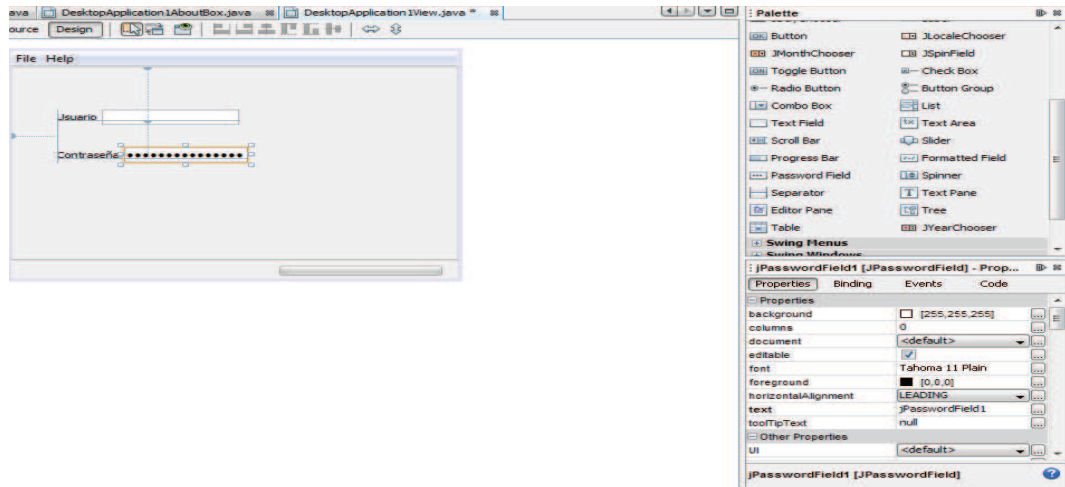


Fig.4.1.4 jLabel “Usuario” junto con un jPasswordField “Contraseña”

De la misma forma que en el paso anterior, arrastramos desde el palette manager un jLabel hacia la ventana de trabajo y cambiamos en sus propiedades el text por el de “Contraseña”, ahora arrastramos el componente de password field hacia la ventana, lo ubicamos y en sus propiedades, cambiamos lo que tenga en text, para dejarlo nulo y dejar libre el campo.

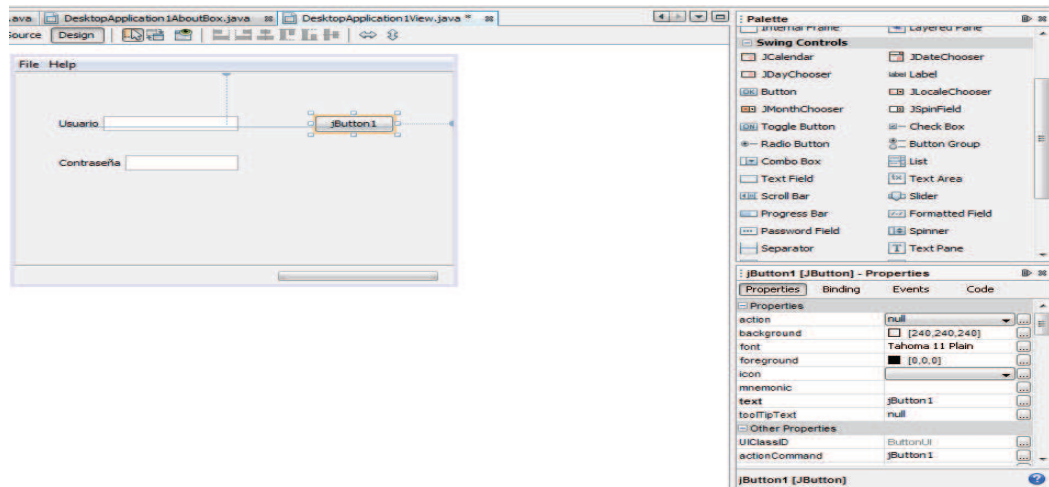


Fig.4.1.5 jButton “Crear”

Para la colocación de los jButton o botones, necesitamos arrastrar desde el palette manager el componente ok button, dentro de los swing controls, una vez colocado el componente dentro de la ventana, lo seleccionamos y dentro de las propiedades de este buscamos el text y cambiamos su contenido por el de ingresar.

jButton “Ingresar”

Se procede de la misma forma que con el botón ingresar, solo que en este caso se cambia el contenido de “text” por el de “Crear”

jButton “Aceptar”

Lo mismo que en los casos anteriores, el text contendrá “Aceptar”

jButton “Cancelar”

En este caso el text será “Cancelar”

Con esto hemos terminado la ventana principal de nuestro programa, la ventana debería tener un aspecto similar al de la siguiente figura:

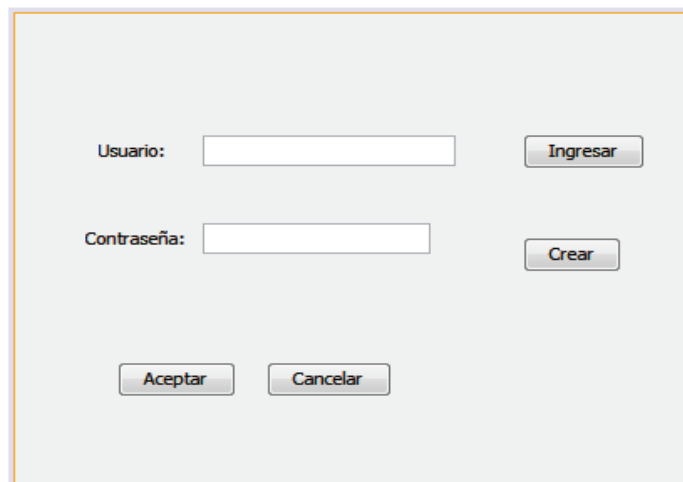


Fig.4.1.6 Ventana principal

4.5 Código de la aplicación

Ahora lo que tendremos que agregar es un método para poder cifrar nuestros datos, para tal efecto, damos clic en Source como en la Fig.4.1.7

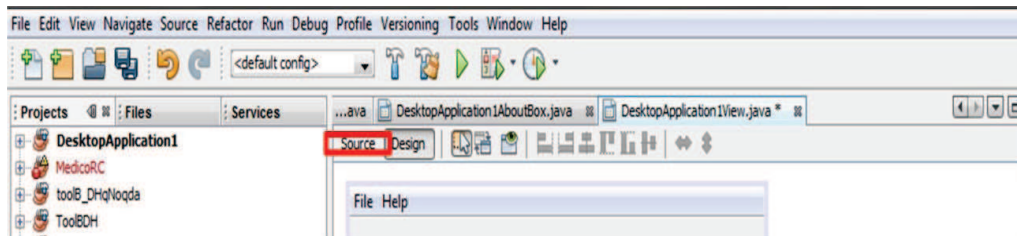


Fig.4.1.7 Pestaña Source

Con lo que se abrirá una nueva ventana que contiene el código fuente de nuestra aplicación, una vez dentro del código, buscamos la línea:

// Variables declaration - do not modify

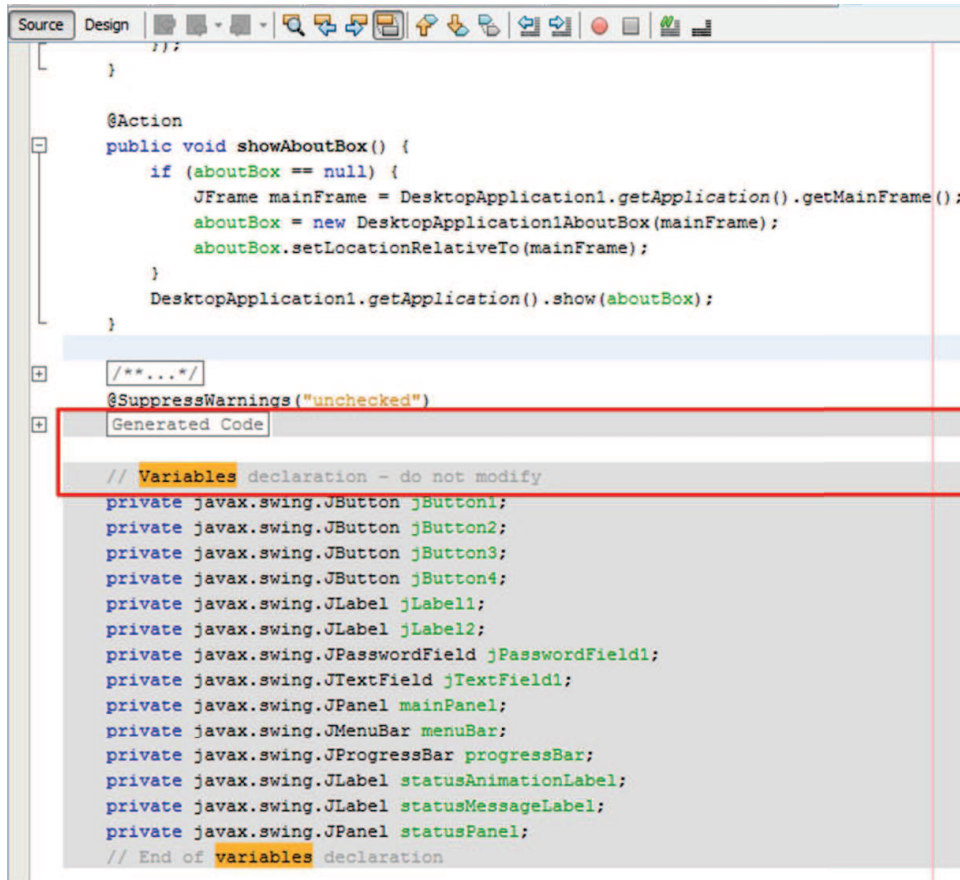


Fig.4.1.8 Localización del código

Y copiamos el siguiente código antes de esta línea, en el renglón en blanco de la fig-4.1.8

```
private static final char[] HEXADECIMAL = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'b', 'c', 'd', 'e', 'f'};

public String hash(String stringToHash) {
    try {
        MessageDigest md = MessageDigest.getInstance("SHA");
        byte[] bytes = md.digest(stringToHash.getBytes());
        StringBuilder sb = new StringBuilder(2 * bytes.length);
        for (int k = 0; k < bytes.length; k++) {
            int low = (int) (bytes[k] & 0x0f);
            int high = (int) ((bytes[k] & 0xf0) >> 4);
            sb.append(HEXADECIMAL[high]);
            sb.append(HEXADECIMAL[low]);
        }
        return sb.toString();
    } catch (NoSuchAlgorithmException e) {
        //exception handling goes here
        return null;
    }
}
```

4.5.1 Manejo de eventos JButton

Aceptar: Damos clic derecho sobre este botón, a continuación seleccionamos la opción **Events->Action->actionPerformed** , con esto se abrirá el código fuente

justo en el lugar donde tendremos que ajustar las acciones de este botón, ahora procedemos a agregar el siguiente código:

```
String user = hash(tfUsuario.getText());
    String contra = hash(tfContraseña.getText());
    File filK = new File("Mu.md5");
    File filL = new File("Mc.md5");
    File filA = new File("Uu.md5");
    File filB = new File("Uc.md5");
    if (tfUsuario.getText().isEmpty() || tfContraseña.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "No pueden ser datos nulos");
    } else {
        if (filA.exists() && filB.exists()) {
            FileReader leebe = null;
            try {
                String clv = "";
                leebe = new FileReader("Uu.md5");
                BufferedReader leebe = new BufferedReader(leebe);
                String lector;
                while ((lector = leebe.readLine()) != null) {
                    clv += lector;
                }
                FileReader leeb = null;
                try {
                    String cl = "";
                    leeb = new FileReader("Uc.md5");
                    BufferedReader lee = new BufferedReader(leeb);
                    String reactor;
```

```

while ((reactor = lee.readLine()) != null) {
    cl += reactor;
}

if (clv.equals(user) || cl.equals(contra)) {
    jButton3.setVisible(true);
    //jButton4.setVisible(true);
}

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "" + ex);
}

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "" + ex);
}
}

if (filK.exists() && filL.exists()) {
    FileReader leebe = null;
    try {
        String clv = "";
        leebe = new FileReader("Mu.md5");
        BufferedReader leebi = new BufferedReader(leebe);
        String lector;
        while ((lector = leebi.readLine()) != null) {
            clv += lector;
        }
    }
}

```

```

    }
    FileReader leeb = null;
    try {
        String cl = "";
        leeb = new FileReader("Mc.md5");
        BufferedReader lee = new BufferedReader(leeb);
        String reactor;
        while ((reactor = lee.readLine()) != null) {
            cl += reactor;
        }

        if (clv.equals(user) || cl.equals(contra)) {
            jButton3.setVisible(true);
            jButton4.setVisible(true);
        }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "" + ex);
    }

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "" + ex);
}
} else {
    try {
        FileWriter fw = new FileWriter("Mu.md5");
        BufferedWriter bw = new BufferedWriter(fw);

```

```

        PrintWriter salida = new PrintWriter(bw);
        salida.print("" + user);
        salida.close();

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "" + e);
    }
    try {
        FileWriter fw = new FileWriter("Mc.md5");
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter salida = new PrintWriter(bw);
        salida.print("" + contra);
        salida.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "" + e);
    }
}
tfUsuario.setText(null);
tfContraseña.setText(null);
}

```

Cancelar: Se procede en la misma forma que con el botón aceptar, pero en esta ocasión el código es distinto:

```

tfUsuario.setText(null);
tfContraseña.setText(null);

```

Ingresar: De igual forma que con los botones anteriores, en este agregaremos un código que nos permitirá ya ingresar a la base de datos para su consulta y edición:

```
JFrame mainFrame = ToolWk2App.getApplication().getMainFrame();
N4 db = new N4(mainFrame, false);
db.setVisible(true);
```

Crear: El código que agregaremos para este botón, será el siguiente:

```
String user = hash(tfUsuario.getText());
    String contra = hash(tfContraseña.getText());
    File filA = new File("Uu.md5");
    File filB = new File("Uc.md5");
    if (tfUsuario.getText().isEmpty() || tfContraseña.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "Introduzca los datos de usuario y
oprima de nuevo este botón");
    } else {
        if (filA.exists() && filB.exists()) {
            FileReader leebe = null;
            try {
                String clv = "";
                leebe = new FileReader("Uu.md5");
                BufferedReader leebebi = new BufferedReader(leebe);
                String lector;
                while ((lector = leebebi.readLine()) != null) {
                    clv += lector;
                }
                FileReader leeb = null;
```



```

try {
    String cl = "";
    leeb = new FileReader("Uc.md5");
    BufferedReader lee = new BufferedReader(leeb);
    String reactor;
    while ((reactor = lee.readLine()) != null) {
        cl += reactor;
    }
    if (clv.equals(user) && cl.equals(contra)) {
        JOptionPane.showMessageDialog(mainPanel, "Ya existe ese
usuario");
    }
    else{JOptionPane.showMessageDialog(mainPanel, "Verifique los
datos");}
} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "" + ex);
}

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "" + ex);
}
} else {
    try {
        FileWriter fw = new FileWriter("Uu.md5");
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter salida = new PrintWriter(bw);
        salida.print("" + user);
    }
}

```

```

        salida.close();

        FileWriter fw1 = new FileWriter("Uc.md5");
        BufferedWriter bw1 = new BufferedWriter(fw1);
        PrintWriter salida1 = new PrintWriter(bw1);
        salida1.print("" + contra);
        salida1.close();
        if (filA.exists() && filB.exists()) {
        JOptionPane.showMessageDialog(null, "Usuario Creado");}
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "" + e);
    }
}

tfUsuario.setText(null);
tfContraseña.setText(null);
}

```

4.6 Contenedor Base de datos

Ahora hace falta crear el contenedor de nuestra base de datos; creamos un nuevo Formulario de ejemplo Maestro/Detalle, dando click derecho en nuestra carpeta de fuente principal y seleccionando esta opción.

El nombre podría ser TOOLBDH como se sugiere en el trabajo anterior.

Se nos pedirá elegir una conexión de base de datos, aquí entrará la base que se creó anteriormente, se selecciona la conexión junto con las tablas a utilizar. O bien se creará una nueva conexión con la base de datos, con los siguientes datos:

name: MySQL/JConnector
host: localhost
port:3306
database: TOOLBDH
User Name: root
Password: nulo
jdbc:mysql://localhost:3306/TOOLBDH

Si se obtiene algún error, habrá que modificar la tabla recién creada, pues netbeans requiere de al menos una llave primaria para poder utilizar la base de datos, es por esto que se ejecuta la siguiente sentencia desde la línea de comandos MySQL:

```
alter table ingreso modify nombreProyecto varchar(50) unique primary key;
```

Realizados estos cambios, elegimos todas las columnas posibles para incluir y por último finalizamos el asistente de netbeans para la conexión con la base de datos. Ahora nos aparecerá una ventana ya con campos y botones configurados, así como una tabla en la que aparecerán los datos de la Base de Datos (Fig. 4.2).

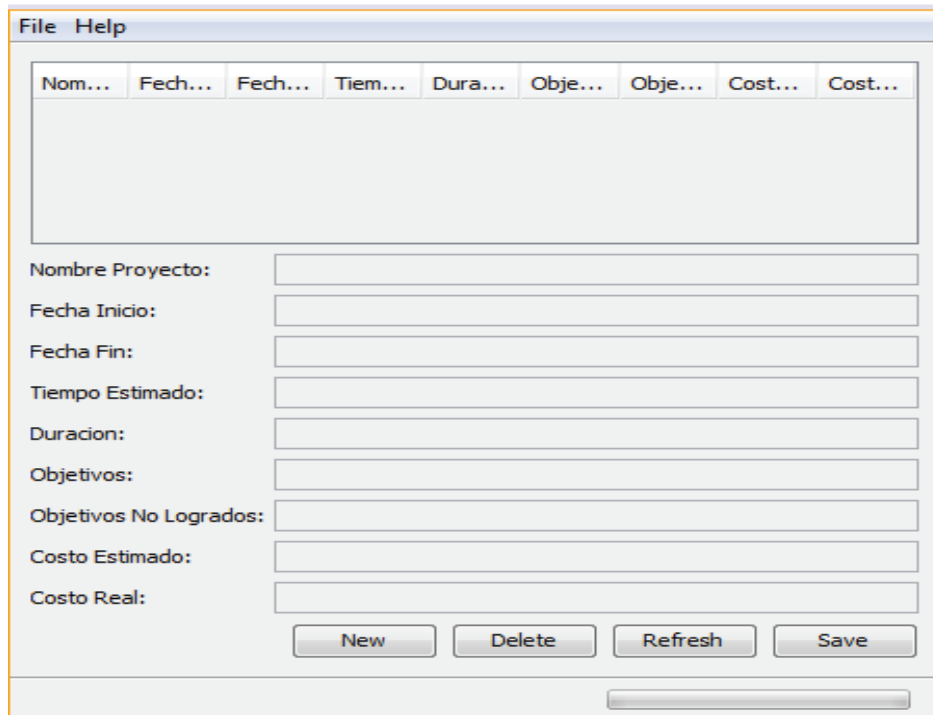


Fig.4.2 Ventana automatizada para bases de datos

4.6.1 Diseño del contenedor

Para la introducción de los datos para el proyecto, primero necesitaremos definir algunos de los campos a utilizar:

Una vez que aparece la página con las opciones automáticas, se necesita ir cambiando los campos de texto según se requiera. Para empezar ajustaremos el tamaño de los campos de texto, arrastrando cada uno de los campos de texto o las etiquetas, aumentándolos o disminuyéndolos según se necesite:

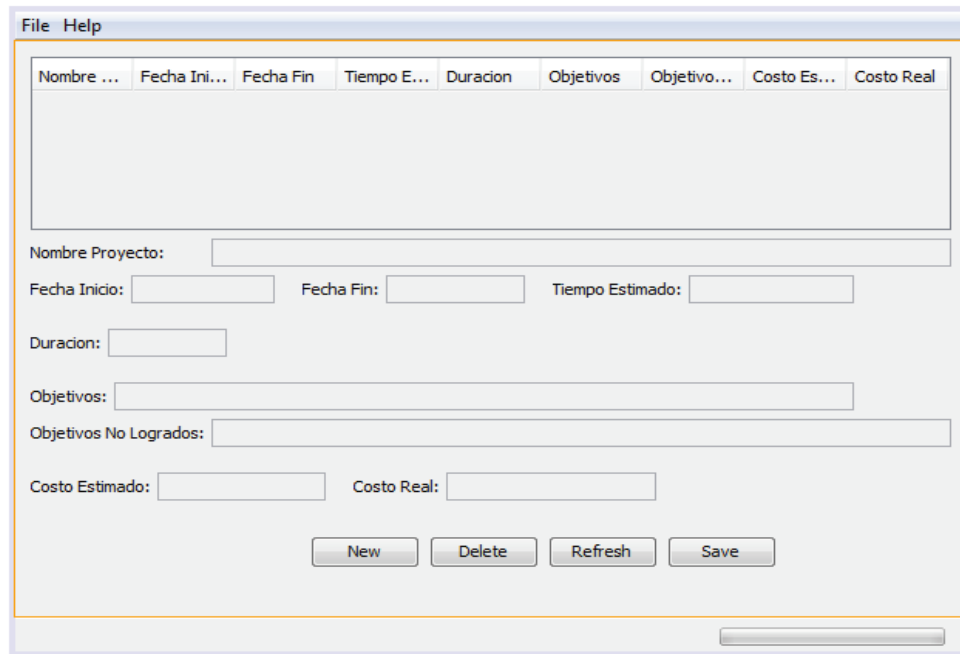


Fig. 4.3 Ventana modificada

Ahora cambiaremos el campo de texto de "Objetivos" por un campo del tipo área de texto, para que de esta manera se vea en su totalidad la descripción de los objetivos del proyecto. Para realizar este cambio lo que se necesita hacer es tomar desde la paleta de herramientas la opción de `textArea`, la seleccionamos y arrastramos a la ventana, cuidando que no salga demasiado de la ventana o que no modifique la posición de cada campo. Ahora, damos click sobre el `textField` previo que se encontraba, buscamos en sus propiedades la etiqueta de `binding`, en esta copiamos las propiedades que tenga el `textField` (`text` y `enabled`) y las pegamos en esas mismas propiedades pero ahora en el `textArea`. Una vez que la copia de estas propiedades esta lista, ya podemos borrar el `textField` previo para "Objetivos". Hacemos lo mismo para el `textField` de "Objetivos no logrados", para cambiarlos también al área de texto (Fig.4.4).

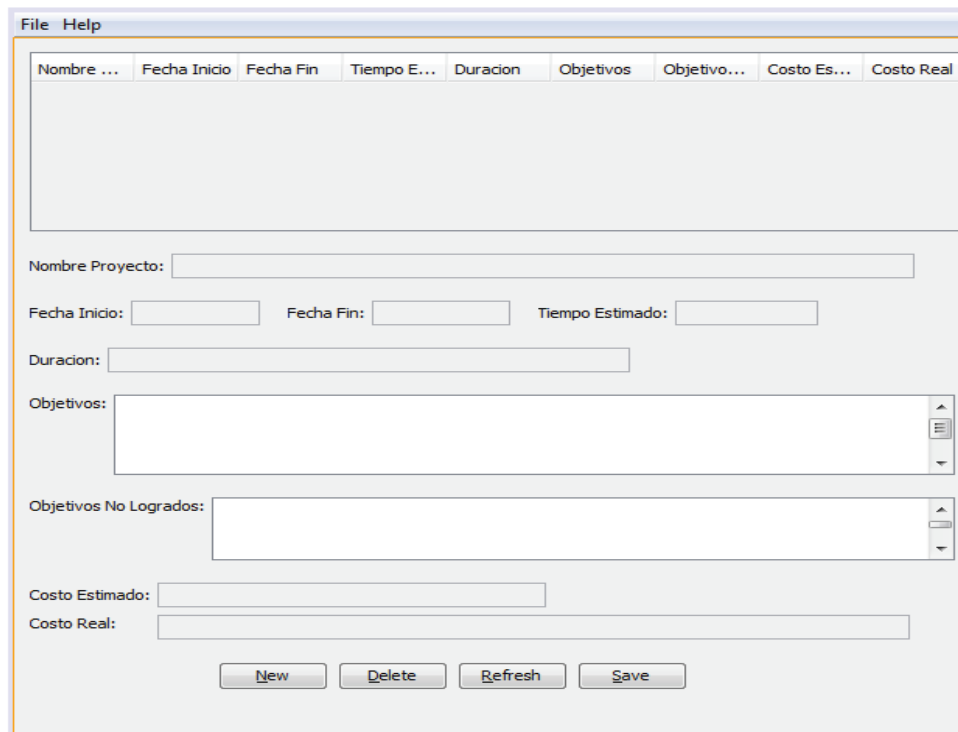


Fig. 4.4 Cambios de jTextField por JTextArea

4.6.2 Calendario

Por lo pronto conviene dejar los botones y la tabla tal cual se encuentran. Ahora se realizará otro cambio a la ventana generada automáticamente, este cambio consiste en utilizar un calendario que haga a nuestra aplicación tener un estilo más profesional, además de evitarnos el desarrollo de un calendario y hacer las validaciones correspondientes a las fechas, para utilizar este calendario es necesario hacer uso de los recursos para descargar un archivo desde la página: <http://www.toedter.com/en/jcalendar/index.html>, o bien, utilizar nuestro buscador favorito y teclear "jcalendar" y descargar el archivo jcalendar -1.3.3.zip, en el cual se encuentra los archivos que utilizaremos. Descomprimos los archivos

que se encuentran en la carpeta lib, esta carpeta la debemos tener bien localizada, pues será la que utilizaremos más adelante.

Para utilizar este paquete es necesario ir a la paleta de herramientas en netbeans, dar click derecho sobre esta y elegir la opción 'palette manager...' posteriormente en 'Add from jar...'

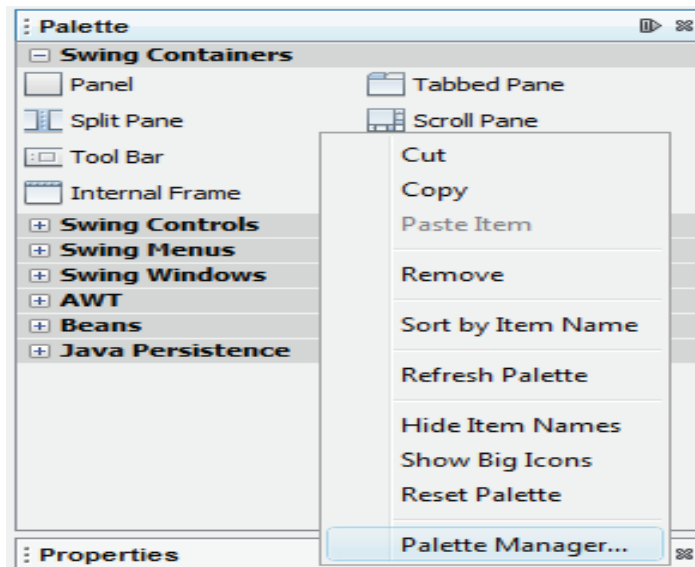


Fig. 4.5 Administración de componentes

Aquí localizaremos el archivo descomprimido 'jcalendar-1.3.3.jar', seleccionamos todas las opciones que nos aparezcan, elegimos la carpeta de la paleta en la que aparecerán las nuevas herramientas y finalizamos el proceso para poder utilizar el calendario.

Ahora tendremos que introducir los calendarios en lugar de los campos de texto "fecha inicio" y "fecha fin".

Simplemente arrastramos el componente JDateChooser, uno de los que acabamos de introducir, a nuestra ventana, lo ajustaremos a las medidas deseadas para la

ventana. Los campos de texto de las fechas no se eliminarán, simplemente los haremos más chicos y apenas visibles, ya dentro del código los haremos invisibles, para esto damos clic en la opción source dentro del marco de nuestro proyecto (Fig.4.6), localizamos la línea en la que aparece initComponents(); y justo debajo de ella colocaremos las dos líneas que harán nuestros campos de texto invisible, estas son:

```
fechaFinField.setVisible(false);
```

```
fechaInicioField.setVisible(false);
```

```
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import org.jdesktop.beansbinding.AbstractBindingListener;
import org.jdesktop.beansbinding.Binding;
import org.jdesktop.beansbinding.PropertyStateEvent;

/**
 * The application's main frame.
 */
public class ToolBDHView extends JFrameView {

    public ToolBDHView(SingleFrameApplication app) {
        super(app);

        initComponents();
        fechaInicioField.setVisible(false);
        fechaFinField.setVisible(false);

        // status bar initialization - message timeout, idle icon and busy animation, etc
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                statusMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
```

Fig.4.6 Fuente del programa

4.6.3 Generación de eventos

Regresamos a la parte gráfica dando clic en Design en el marco de nuestro proyecto. Ahora hay que programar los eventos que generará el calendario y que

pasarán la fecha a las cajas de texto que hemos desaparecido, para realizar tal acción, debemos dar clic derecho encima de la forma de calendario que recién agregamos, en cualquiera de las dos, pues el proceso es el mismo, a continuación elegir la opción Events->PropertyChange->propertychange y agregamos el siguiente código dependiendo del calendario sobre el que estemos trabajando:

```
try {
    SimpleDateFormat formato = new
SimpleDateFormat(this.jDateChooser1.getDateFormatString());
    String fechaRe[] = {formato.format(this.jDateChooser1.getDate())};
    fechaInicioField.setText("" + fechaRe[0]);
    } catch (Exception e) {
    }
```

Y para el calendario fechaFin, las sentencias quedan como:

```
try {
    SimpleDateFormat formato = new
SimpleDateFormat(this.jDateChooser2.getDateFormatString());
    String fechaRe[] = {formato.format(this.jDateChooser2.getDate())};
    fechaInicioField.setText("" + fechaRe[0]);
    } catch (Exception e) {
    }
```

Con esto, lo que se está logrando, es que al cambiar la fecha de los calendarios, este cambio se refleje en los campos de texto y estos a su vez en la base de datos, se tiene que realizar esto, debido a que los componentes para ocupar los calendarios no son nativos de netbeans y al ser realizados por terceros, no cumplen

con todas las propiedades necesarias para un manejo más práctico y sencillo que el aquí presentado. Ahora bien, el siguiente paso sugerido es que para el campo del tiempo estimado agreguemos 3 combo box, para ser utilizados como la fecha en la que se estime sea el final del proyecto, uno de estos será utilizados como los meses, otro como las semanas y el final como los días, el proceso para realizar estos es el siguiente:

-Se toma de la paleta de herramientas un combo box y se arrastra hasta la ventana de desarrollo.

-Se cambia el nombre de la variable por otro, en este caso cbMes, lo mismo el contenido por default, este se cambia por 48,36,24,12,11,10,9,8,7,6,5,4,3,2,1,0, que será la cantidad de meses que se podrán elegir , no se agregan más, debido a que los proyectos de software por lo general no son tan largos, de la misma forma que para otros componentes, se ajusta el tamaño de este componente a uno acorde a las medidas de la ventana.

-De la paleta se toma nuevamente un combo box, se cambia el nombre de la variable por cbSemana, el contenido por default se cambia por 0,1,2,3 que será el número de semanas que se podrán elegir.

-Por último, de la paleta tomamos un jcombo box y lo ajustamos a nuestra ventana, cambiamos su contenido previo por 1,2,3,4,5,6 y 7 que serán el número de días que se podrán elegir.

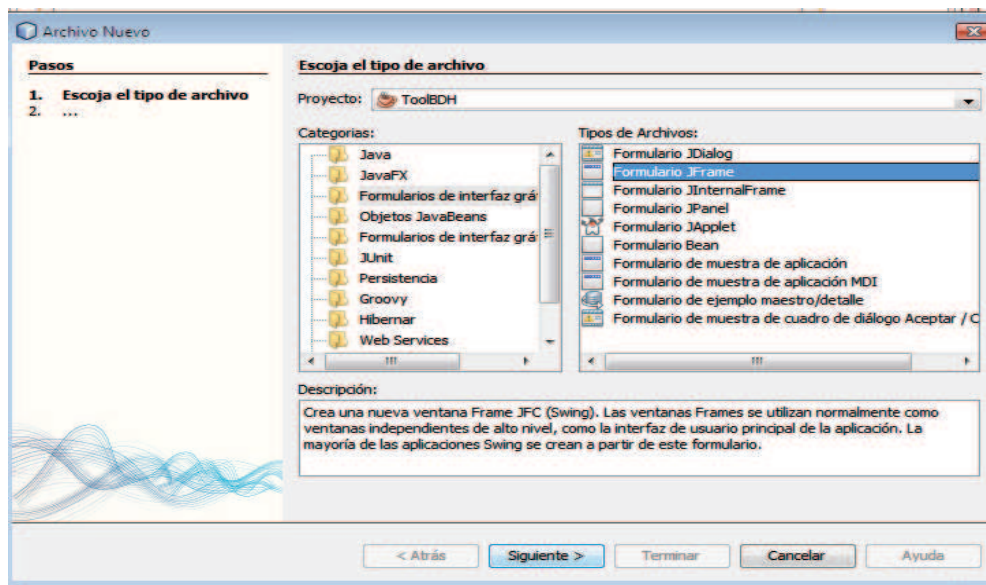
-En el modo Design damos clic derecho sobre cada uno de los jcomboBox y elegimos la opción de Event->Item->ItemStateChange, NetBeans creará en automático cada uno de los eventos, y para asegurarnos que se actualice se tiene que realizar en cada uno de los eventos la misma sentencia, la cual será:

```
tiempoEstimadoField.setText(cbMes.getSelectedItem().toString()+" meses  
"+cbSemana.getSelectedItem().toString()+" semanas  
"+cbDia.getSelectedItem().toString()+" dias");
```

y ahora en cada ocasión en que se cambie la JComboBox de cada uno de los elementos se determinará el nuevo tiempo estimado.

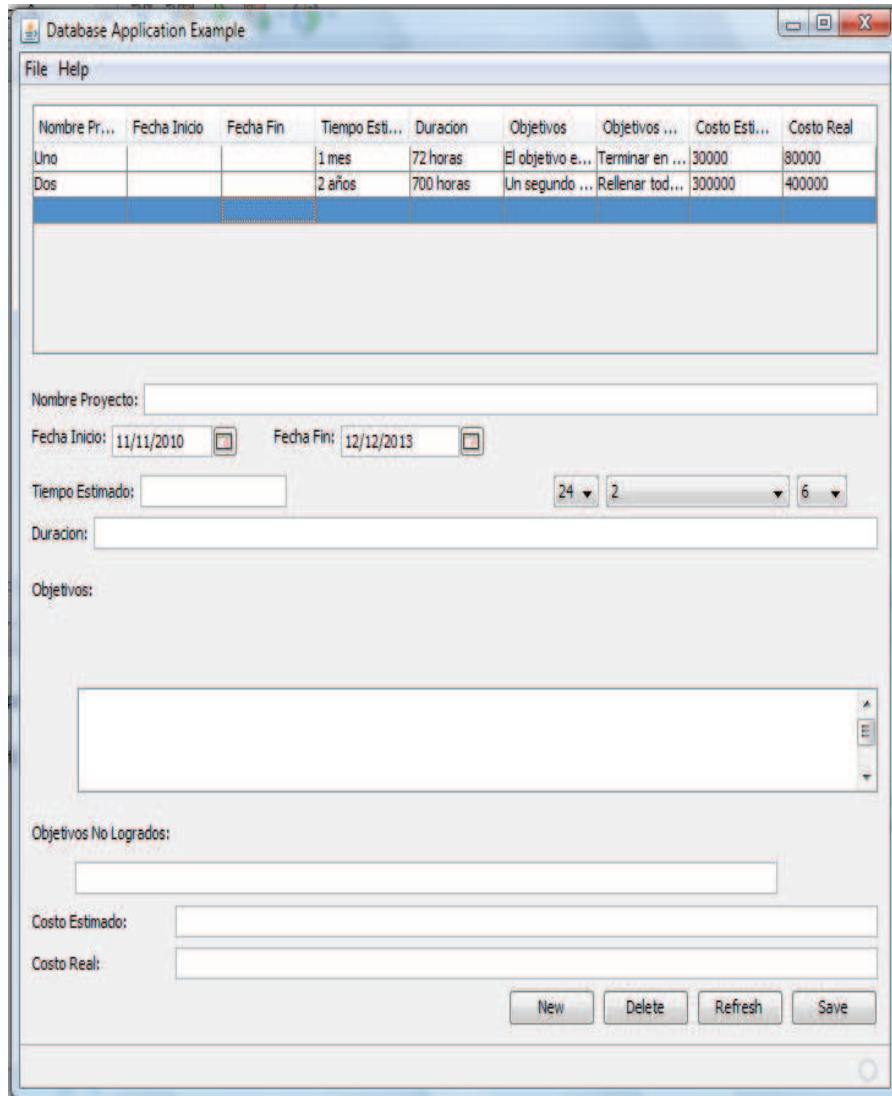
Realizados estos cambios, se verifica que no exista ningún error en el programa, compilándolo y corriéndolo dentro del mismo NetBeans.

Ahora con los nuevos cambios, procedemos a crear un nuevo JFrame que será la forma en la que nuestra ventana principal se comunicará con la base de datos, lo que debemos hacer es simplemente darle click derecho a la carpeta fuente principal y seleccionar un formulario JFrame con el nombre N4 y finalizar el asistente.



4.7 Selección de un JFrame

Nos aparecerá una ventana sin ningún decorado u objeto, a esta ventana simplemente arrastramos el archivo que creamos con la base de datos a su interior y nos aparecerá ya nuestra interfaz para la base de datos lista para ser utilizada, únicamente se ajustará el tamaño del JDialog para que empate con este archivo.



4.8 Programa de pruebas

Es así como ya terminado este proyecto, se puede proceder con las pruebas que tendrán como fin el dar una vista previa de lo que se pretende que sea el proyecto final, y con el cual nos percataremos de las acciones a emprender para el trabajo definitivo.

5. Proyectos similares

5.1 Perfil actual

En la actualidad las bases de datos juegan un papel muy importante en las empresas y sobre todo en el mundo del software, ya que es una gran cantidad de aplicaciones las cuales se pueden utilizar o realizar con las bases de datos, así como las conexiones que facilitan el uso de las mismas.

Las bases de datos están presentes en las organizaciones donde se maneja gran cantidad de información y en las que se necesita llevar a cabo un control sobre los datos. Son utilizadas para simplemente almacenar desde los empleados de un comercio, hasta para procesar datos de todo un país.

Una de las metas de tener bases de datos es tener toda la información posible referente a un tema y que toda esta información sea ordenada de acuerdo a ciertas características, pero sobretodo que cada uno de los registros de que se dispongan sea único e irreplicable así sea en un solo carácter.

En este proyecto, las bases de datos se utilizan para almacenar proyectos de software que deberán ser identificados por los parámetros ya indicados, de tal forma que cada proyecto sea único y fácilmente localizable. Y así como este proyecto tiene por finalidad el facilitar la entrada de distintos proyectos de software, existen varios proyectos similares en todo el mundo, a continuación se dan a conocer los atributos y características de cada uno de ellos, al final de esta sección se tiene un cuadro comparativo entre estos proyectos.

5.2 Páginas comerciales

Revisemos la siguiente página web

<http://www.proyectosysoluciones.es/>

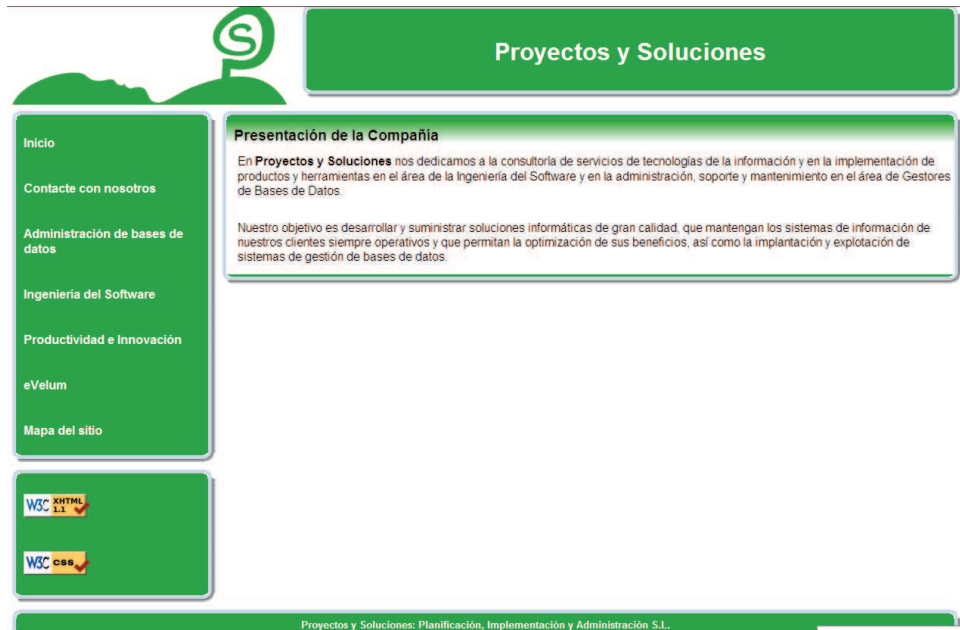


Fig. 5.1 Proyectos y soluciones

En la Fig. 5.1 se muestra cómo es que algunas empresas se dedican a la administración de bases de datos, es decir, existe una gran oportunidad de negocio en el mercado de bases de datos, así mismo, se muestra que en esta página hay también una oportunidad de negocios en cuanto a la ingeniería de software, por lo que podemos deducir que nuestro proyecto tiene características que son útiles para las empresas en general. La Fig. 5.2 se muestra el negocio sobre cómo se debe ir realizando paso a paso el desarrollo de los sistemas de gestión de bases de datos, el tipo de hardware que se debe de utilizar y las consideraciones finales una vez que se ha terminado el desarrollo del proyecto.

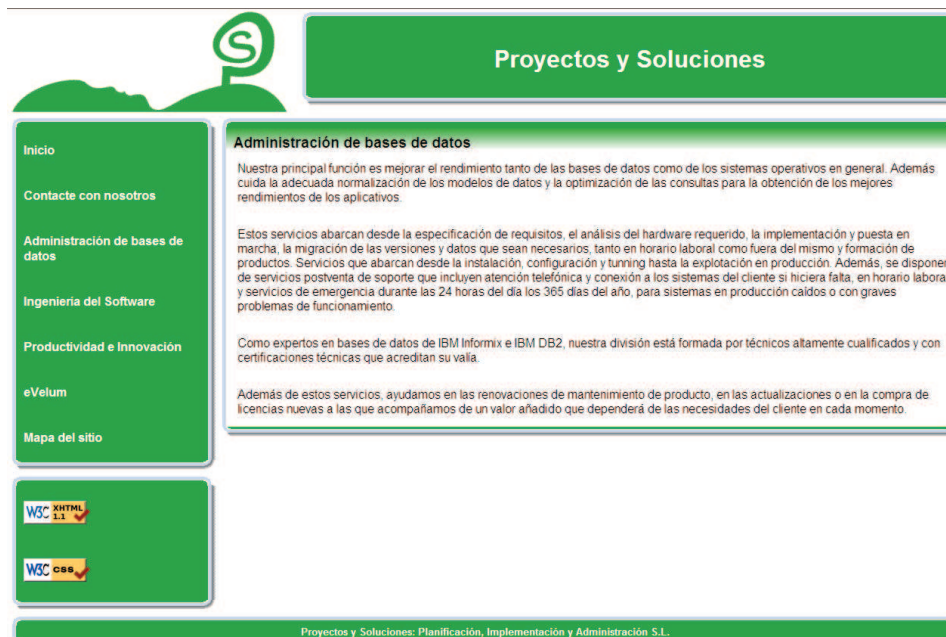


Fig. 5.2 Proyectos y soluciones, administración BD

5.3 Páginas gubernamentales

<http://www.hcdn.gov.ar/dependencias/dip/grvprov2.html>

En la Fig. 5.3 nos muestra una interfaz desarrollada para la entrada de datos, la cual tiene, varios campos de entrada, pudiendo ser cajas de texto, cajas combo y cajas de chequeo, para poder tener una mayor facilidad para el manejo de datos y de esta forma evitar hasta donde sea posible las entradas de datos erróneas, o bien, omisión en los datos. Para poder utilizar de forma adecuada la interfaz, se requiere una guía previa, ya que los datos o bien las etiquetas de cada uno de los campos no resultan ser muy claras, por lo que, presumimos existe una guía o manual en el cual se habla de la forma en la que se deben de introducir los datos y cual es un rango de valores aceptados por parte del sistema, ya que de esta forma se tiene una seguridad pasiva en la base de datos, pues al no saber cuales son los

datos correctos que se pueden utilizar, el sistema rechazará la entrada de datos y seguramente el usuario optará por abandonar sus intentos por introducir datos en el sistema.

Esta es básicamente una forma de consulta, ya que no se pueden introducir nuevos datos, únicamente consultar los datos históricos que están almacenados dentro de la base de datos, lo cual nos permite visualizar una forma eficiente de buscar información, ya que en esta forma, un usuario común y corriente, no necesitará conocer de sentencias especializadas para poder encontrar los datos que requiere, sino que únicamente bastará con datos clave para comenzar a buscar y utilizar este sistema.

5.3.1 Resultados de las búsquedas

Secretaría Parlamentaria
Dirección de Información Parlamentaria

Base de Proyectos presentados a partir de 1999
DATOS DE ACTUALIZACIÓN

Cámara iniciadora: senado
Expediente: 73862267 Orden del Día: 686
Tipo de proyecto: declaración Período de ingreso: 122 (01/03/04 al 28/02/05)
Resultado: aprobado Ley N°: 45637
Firmante: Matias Bloque: Demóstrata Progres Dist: Entre Ríos
Cofirmante: Lopes Bloque: Compromiso para el Cambio Dist: Mendoza
Comisión Diputados: Altas Comisión Senado: Baja
Búsqueda general: Base ingeniería
ejecutar borrar
 Mostrar sólo registros con aciertos Mostrar sólo encabezados con aciertos
Efectúe las búsquedas sin colocar acentos en las palabras
No es necesario completar todos los casilleros

Fig. 5.3 Secretaría Parlamentaria

En la Fig. 5.4 se muestra además cual es el resultado de una consulta realizada sin un conocimiento de cuáles son los parámetros adecuados para que se genere una búsqueda exitosa. En el final de la búsqueda se muestra cual es la forma en la que

los resultados aparecerán, es decir, las etiquetas y el orden que seguirán los resultados para su mejor apreciación por parte del usuario. Aparece además un link, con el cual podemos ver los documentos que aparecieron después de la ejecución de la consulta.



Fig. 5.4 Proyectos encontrados

La siguiente figura(Fig. 5.5) muestra una sub búsqueda, en la cual no se necesita conocer los tipos de palabras a buscar, pues realiza una búsqueda más general dentro de todos los componentes de la base histórica.

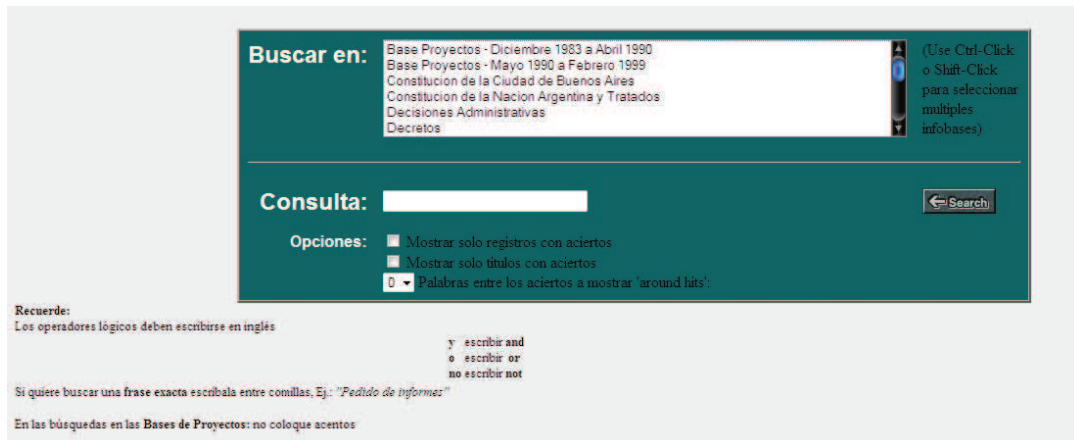


Fig. 5.5 Búsqueda general

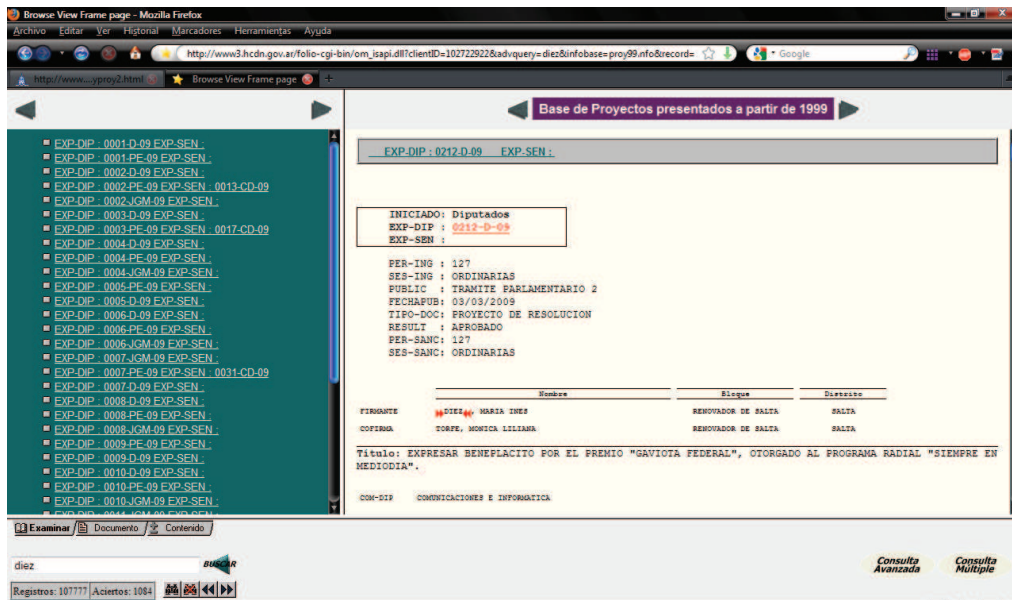


Fig. 5.6 Documentos encontrados

http://www3.hcdn.gov.ar/folio-cgi-bin/om_isapi.dll?clientID=102722922&advquery=diez&infobase=proy99.nfo&record={F9}&softpage=Browse Frame Pg42&x=25&y=12&zz=

La Fig. 5.6 muestra que también existe una búsqueda más general, por medio de la cual se puede navegar por todos los documentos de la base de datos, los cuales están organizados por categorías de acuerdo al número de expediente o documento que se está buscando, así mismo dentro de cada una de estas categorías se encuentran ordenadas las propuestas realizadas por una o varias personas, organizadas de acuerdo a la fecha en que se propusieron, agrupación a la que pertenecen, orden alfabético, así como el bloque. Estas búsquedas son muy generales, ya que no se introduce ningún dato para su búsqueda, consiste básicamente en estar navegando a través de los distintos expedientes .

5.4 Páginas históricas

<http://www.asasve.es/portal/index.php?mod=article&cat=biblioteca2>



Fig. 5.7 Página principal

Esta es una base de datos históricos para la consulta por parte de cualquier usuario que se encuentre interesado en la consulta de los temas de la historia militar española. Esta página (Fig. 5.7) tiene la peculiaridad de que sus búsquedas ya están todas previamente almacenadas en los componentes de la misma , en este caso, las consultas se podrán realizar únicamente seleccionándolas a través de la caja combo, o los links en forma de flecha que aparecen junto a esta, con las opciones de búsqueda que nos interesen, así mismo, lo cual no facilita las búsquedas, pues el componente combo box no siempre resulta atractivo para el usuario, además de que en ocasiones, con tantas opciones, el usuario puede llegar a perderse en sus búsquedas. La presentación de los datos es muy original y atractiva, ya que en casi todas las opciones de búsqueda, aparecen imágenes que asemejan a carteles de la época en cuestión(Fig. 5.8), con un buen diseño tanto gráfica como tipográficamente, en unos cuantos resultados, aparece únicamente texto sin imágenes, para el caso de por ejemplo una bibliografía.

ica 2010 ; La Armada participa en la Africa Partnership Station 2010 Palabra: Buscar [Búsqueda avanzada]

DE INTERÉS » BASE DE DATOS HISTÓRICOS

BIBLIOTECA DE DATOS HISTÓRICOS
Viernes, 01.01.2010, 10:00 (GMT+1)

Estadillo de Aspirantes ascendidos de la IPS

Por Francisco Ángel Cañete Páez
Profesor Mercantil y Licenciado en Ciencias Económicas,
Comandante de Infantería de la Escala de Complemento.

Instrucción Premilitar Superior
Universitarios españoles que pasaron por los
Campamentos de la IPS en sus 30 Promociones
(1942-1972)

• Caballeros Aspirantes Universitarios.....	167.887
• Caballeros Aspirantes (Cuerpo Armado)..	1.193
Total Caballeros Aspirantes	169.080
• Promovidos a Alféreces.....	121.096
• Promovidos a Sargentos.....	42.910
• Bajas en la I.P.S.....	5.074
Total Aspirantes (1942-1972)	169.080

De los 121.096 Alféreces, fueron promovidos, tras la superación de los Cursos de Aptitud correspondientes y dentro de la Escala de Complemento:

• A Tenientes.....	5.779
• A Capitanes.....	791
• A Comandantes.....	136

"Nunca la Pluma embotó la Espada"

© Portal ASASVE

Se autoriza su publicación mencionando al autor del Estadillo, D. Francisco Ángel Cañete Páez, y a la fuente que lo publica, ©Portal ASASVE.

Portal ASASVE

BIBLIOTECA DE DATOS HISTÓRICOS

Comentarios (3) Imprimir Enviar a un amigo Arriba

AGENDA
Marzo 2010

Lu	Ma	Mi	Ju	Vi	Sa	Do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

BREVES EN IMÁGENES

Pulse sobre la imagen y el titular

ENFOQUE ESTRATÉGICO

Boletín on-line sobre Defensa, Seguridad y Relaciones Internacionales Iberoamericanas

DESTACADO

BIBLIOTECA DE DATOS HISTÓRICOS

el Gran Capitán
Historia Militar
NINGUN AÑO!

Minerva Hosting
29€
¡Reservado con garantía!

¡CLASIFICADO MEJOR HOSTING DEL 2009 Y 2008 por MEJOR SERVIDOR, MEJOR SERVIDOR ESPAÑOL!

Minerva Hosting
29€
¡Reservado con garantía!

¡Foto y su disco duro Backup Remoto!

Minerva Hosting
¡Reservado con garantía!

¡Reservado con garantía!

0.012€/min

ENCUESTA
¿Qué opinión le merece Portal ASASVE?

Excelente

Fig. 5.8 Ejemplo de documentos encontrados

5.5 Páginas del tipo legislativo

http://faolex.fao.org/cgi-bin/faolex.exe?search_type=entry&lang=spa&page_header=SPHALL&database=HIST&table=hist

Base de datos histórica

Correspondencia exacta de las palabras: [busque] [salvar] [isis.qi] [despeje]

Base de datos histórica FAOLEX

Presentación de resultados: Cronológico

País/Territorio [O] [lista]

Palabras del título [O] [lista]

Año desde hasta

Legislación vigente	<input checked="" type="checkbox"/> Excluir	
Tipo de texto	<input type="checkbox"/> Constitución	<input type="checkbox"/> Acuerdo
	<input type="checkbox"/> Legislación	<input type="checkbox"/> Reglamento
	<input type="checkbox"/> Miscelánea	

[busque] [salvar] [isis.qi] [despeje]

Fig. 5.9 Búsqueda general Faolex

En este sistema de consultas (Fig. 5.9), nos podemos percatar que también se muestran componentes comunes, tales como las etiquetas de texto, las cajas de chequeo, botones de aceptación, caja de texto y las cajas de combo, cada uno de estos componentes va limitando la consulta para de esta forma realizar un proceso de búsqueda más preciso y eficiente, ya que al ir eliminando componentes de búsqueda, el tiempo de espera por los datos va disminuyendo.

Este sistema resulta particularmente útil, ya que, al únicamente permitir cuatro campos para que el usuario introduzca datos de consulta, los posibles errores disminuyen y se limitan a estos cuatro campos.

Los cuatro botones que se utilizan tienen cada uno una función en particular, para empezar:

Botón **busque**: Con este botón, una vez que se han introducido los campos requeridos comienza la búsqueda de los datos y se cambia de página a una que contiene los resultados de la búsqueda, si no se llenan todos los campos se ejecuta la instrucción y muestra un error. Dependiendo de la búsqueda, el mismo sistema nos indica si se trata de una formulación muy general o no. Como se puede observar (Fig. 5.10), los resultados aparecen en un listado, sin un orden aparente, en el cual, al seleccionar alguno de los links, nos envía a una vista del documento, para posteriormente, si el usuario está interesado en el documento, solo selecciona la caja de chequeo y con el botón **bajar** se descarga el documento de su interés para su posterior revisión, o bien para mantener una copia del mismo.

5.5.1 Resultados obtenidos



Fig. 5.10 Resultados Faolex

Botón salvar: este botón, una vez introducidos todos los datos, pregunta si es que quiere guardar su consulta dentro de la página en un marcador del navegador utilizado para su posterior consulta, es decir, esta opción agregará la página a sus favoritos ya con el texto ingresado antes de la ejecución de la búsqueda, lo cual, si la búsqueda genera una amplia gama de resultados y el usuario se encuentra interesado en varios de ellos, puede resultar práctico, aunque sería más práctico utilizar esta opción para cada uno de los resultados y no para el paso anterior a la consulta.

Botón isis ql: La función de este botón es la de abrir una nueva ventana (Fig. 5.11), para hacer una consulta en crudo, es decir, nos muestra la sentencia que ejecutará el programa y nos pide el número máximo de los registros que aparecerán, es de notar, que esta sentencia que aparece en automático, puede ser modificada por el usuario, siempre y cuando se utilice el mismo lenguaje y sintaxis que en el programa.

5.5.2 Búsquedas Isis ql

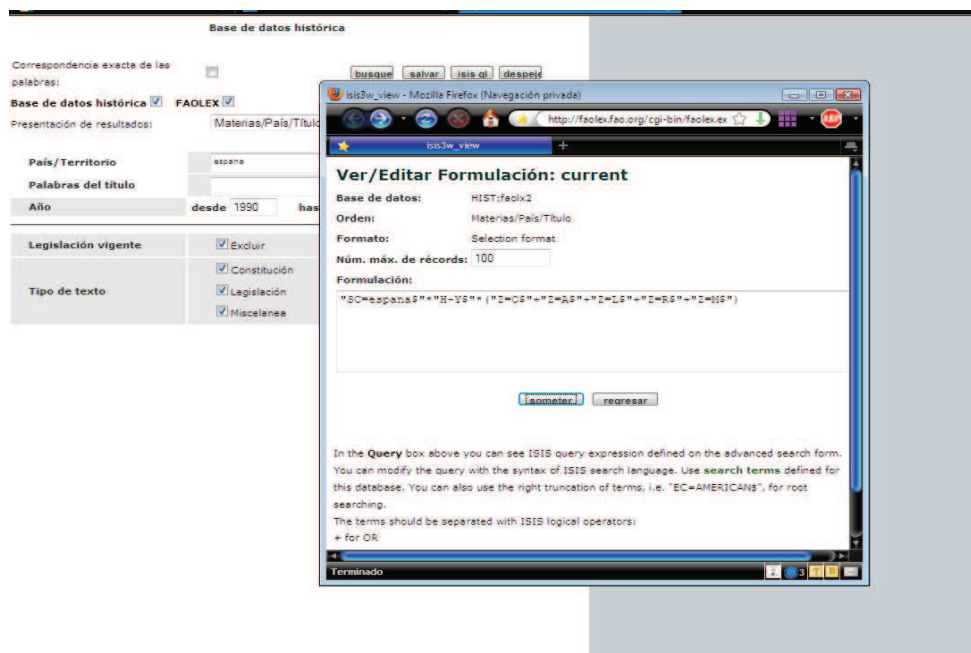


Fig. 5.11 Búsqueda Isis ql

Botón despeje: Este botón realiza un reset a los datos previos para generar una consulta desde cero, esto por facilidad, ya que a veces se quiere hacer una nueva consulta y resulta impráctico estar borrando campo por campo, o bien puede generar confusiones que desembocarían en una consulta no exitosa de datos.

5.6 Páginas de transparencia gubernamental

Estás en: [Información al ciudadano](#) > [Ayudas y Subvenciones](#) > Base de datos histórica de convocatorias

INFORMACIÓN
AL
CIUDADANO
[Empleo](#)
[Legislación](#)
[Ayudas y subvenciones](#)
[Carnés de Industria](#)
[Becas](#)
[Premios](#)
[Transporte](#)
[Quejas y sugerencias](#)
[Direcciones útiles](#)
[Publicaciones](#)
[Impresos](#)
[I.T.V. Inspección Técnica de Vehículos](#)

Base de datos histórica de convocatorias de ayudas y subvenciones

Vigentes y no vigentes

Ayuda / Subvención:

Administración convocante:

Boletín: Boletín Oficial de la Provincia de Zaragoza ▾

Fecha de publicación: Desde: 2004 2005 2006 2007 2008 2009 2010
Hasta: 2004 2005 2006 2007 2008 2009 2010

[Aviso legal](#) | [Política de privacidad](#) | [Mapa web](#)

© Gobierno de Aragón. Todos los derechos reservados.

[Reducir](#) [Normal](#) [Ampliar](#)

[Accesibilidad](#)

Fig. 5.12 Portal Aragon

<http://portal.aragon.es/portal/page/portal/IAD/AYUDAS/Base%20de%20datos%20hist%C3%B3rica%20de%20convocatorias>

Este es un sistema dedicado a la consulta de datos históricos, como se puede observar (Fig. 5.12), no tiene algún elemento gráfico atractivo para el usuario, lo que incide en el gusto de los usuarios por utilizarlo, lo que a la larga será un factor determinante para el uso del sistema.

Las opciones de búsqueda están limitadas un poco, pues existen tres elementos en los que el usuario no puede intervenir en cuanto a la modificación de estos, lo que en dos elementos si se puede realizar, es decir, el usuario debe saber precisamente que es lo que se debe buscar, pues de otra forma la consulta no será exitosa, o bien arrojará resultados muy generales.

Al igual que en los anteriores sistemas, este cuenta con varios elementos típicos, tales como las cajas de texto, en las que el usuario puede introducir información,

las etiquetas de texto, que dan al usuario la idea de lo que se debe introducir en la búsqueda.

Este sistema tiene solamente dos botones, buscar y borrar, el primero ejecuta la búsqueda de las palabras y fechas que el usuario haya introducido, el segundo realiza un reset a los datos, para que tanto las cajas de texto como los botones radio regresen a su estado original.

A continuación se muestra una búsqueda en la que los campos se quedan en blanco y los años de búsqueda van desde el 2004 al 2010.

5.6.1 Resultados

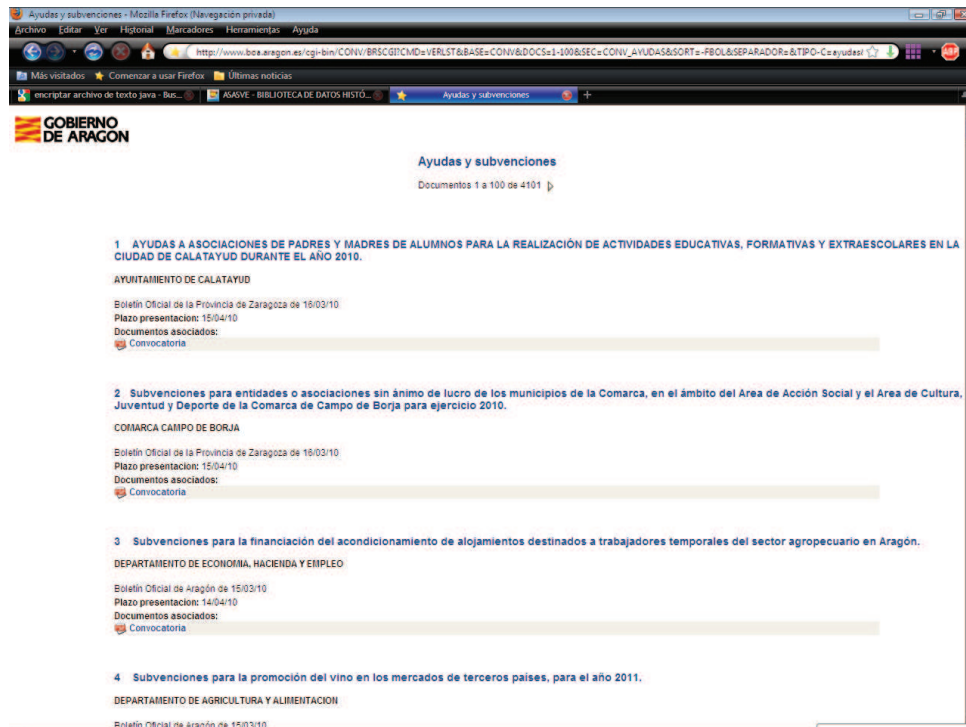


Fig. 5.13 Resultados Aragon

En la Fig. 5.13 se muestra el modo en que se presentan los resultados es en documentos que tienen además otros asociados que aquí aparecen como

convocatorias, cada uno con un número, el título de la misma y se organizan de acuerdo al plazo de presentación en forma descendente. Se aprecia que en la página de resultados, ya se tienen elementos gráficos más atractivos para el usuario, tales como la imagen junto a la leyenda gobierno de aragon, los distintos colores del texto, el ícono junto a los documentos asociados, el sombreado de estos mismos y el espacio entre los textos. También es de notar que se muestra en el resultado, el número total de documentos encontrados en la consulta, por ejemplo, en esta imagen, se muestra que existen un total de 4101 documentos en la base de datos, ya que como se mencionó esta es la consulta más general posible, pues en las cajas de texto no se introdujo dato alguno y se tomaron los límites de los años para la consulta.

5.7 Páginas de divulgación científica

http://www.vhfdx.net/propindex/index_e.html

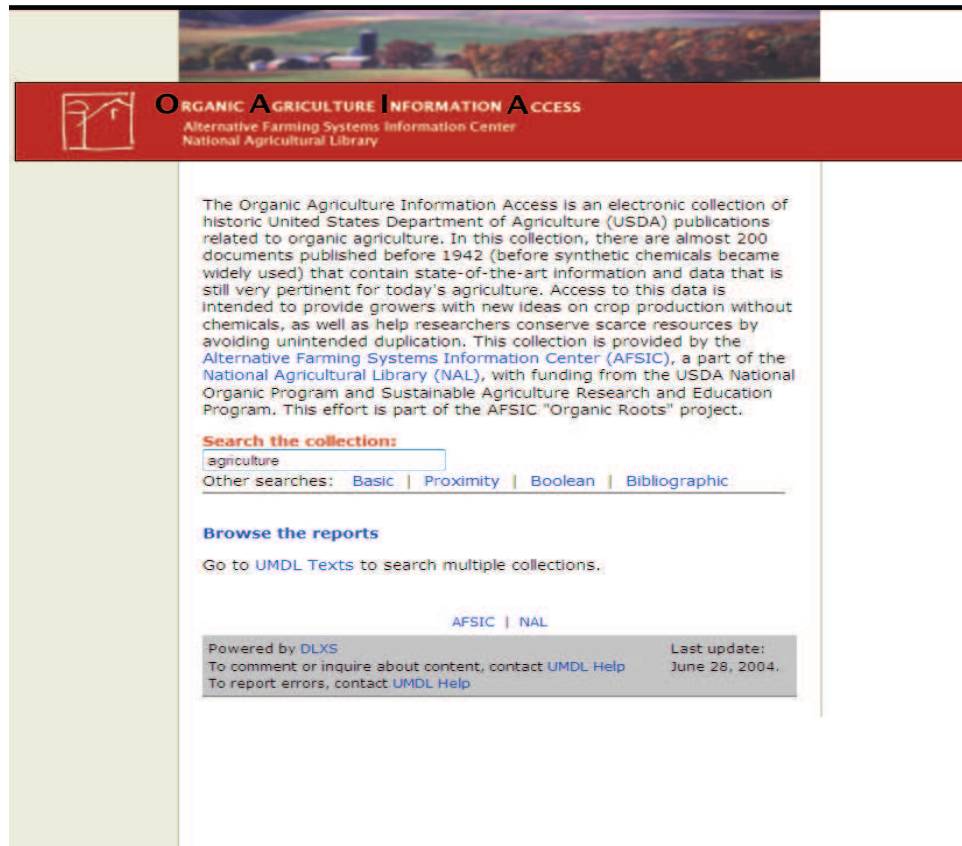


Fig. 5.14 Índices de propagación

En este sistema (Fig. 5.14), como nos podemos dar cuenta, tiene únicamente como elementos de control, etiquetas de texto, cajas combo y un botón para la consulta, es así como la consulta en general está totalmente controlada, debido a que no es posible introducir datos no válidos, es decir, datos que no se encuentren dentro de la misma base de datos histórica. Se observa además como es que este sistema cuenta con un mayor tipo de diseño que otros, ya que si bien no se puede decir que el sistema esté diseñado para impactar, es cierto que ofrece un diseño entretenido y que facilita el uso del mismo por parte del usuario al ser más atractivo, al tener colores más variados tanto en el fondo como en el texto, lo mismo que varios íconos que llaman la atención de del usuario.

5.7.1 Resultados

<http://quod.lib.umich.edu/n/nal/>



The screenshot shows the homepage of the Organic Agriculture Information Access system. At the top, there is a red banner with the title "ORGANIC AGRICULTURE INFORMATION ACCESS" and the subtitle "Alternative Farming Systems Information Center National Agricultural Library". Below the banner, there is a paragraph of introductory text explaining the collection's purpose and origin. The main content area features a search bar with the text "agriculture" entered, and a list of search options: "Basic", "Proximity", "Boolean", and "Bibliographic". There is also a "Browse the reports" section with a link to "UMDL Texts". At the bottom, there is a footer with contact information and a last update date of June 28, 2004.

The Organic Agriculture Information Access is an electronic collection of historic United States Department of Agriculture (USDA) publications related to organic agriculture. In this collection, there are almost 200 documents published before 1942 (before synthetic chemicals became widely used) that contain state-of-the-art information and data that is still very pertinent for today's agriculture. Access to this data is intended to provide growers with new ideas on crop production without chemicals, as well as help researchers conserve scarce resources by avoiding unintended duplication. This collection is provided by the Alternative Farming Systems Information Center (AFSIC), a part of the National Agricultural Library (NAL), with funding from the USDA National Organic Program and Sustainable Agriculture Research and Education Program. This effort is part of the AFSIC "Organic Roots" project.

Search the collection:
agriculture
Other searches: [Basic](#) | [Proximity](#) | [Boolean](#) | [Bibliographic](#)

Browse the reports
Go to [UMDL Texts](#) to search multiple collections.

AFSIC | NAL

Powered by DLXS
To comment or inquire about content, contact [UMDL Help](#)
To report errors, contact [UMDL Help](#)

Last update:
June 28, 2004.

Fig. 5.16 Acceso a la información de agricultura orgánica

En este sistema (Fig. 5.16), se muestran opciones de búsqueda organizadas en una forma distinta a las ya vistas, pues cuenta con elementos nuevos, tales como los links en sustitución a los botones ya ocupados, ofrece un poco de información sobre lo que se podría encontrar en las consultas a la base de datos a manera de resumen, así como el objetivo del desarrollo del sistema. Las opciones de búsqueda son muy abiertas, es decir, el usuario introduce cualquier carácter y se tiene que arrojar algún resultado, dado que lo que se está buscando es un contenido dentro de los textos almacenados, esta es la única búsqueda válida, ya

que el poner las búsquedas como una caja combo, generaría opciones de la misma caja que resultarían casi interminables y muy tediosas para el usuario, la única desventaja sería el tiempo de procesamiento de las ordenes de la búsqueda. Las opciones para buscar son los siguientes links:

5.8.1 Tipos de búsqueda

Organic Agriculture Information Access - Mozilla Firefox (Navegación privada)

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://quod.lib.umich.edu/cgi/t/text/text-idx?page=simple;c=na1

Más visitados Comenzar a usar Firefox Últimas noticias

encriptar archivo de texto java - Bus... Organic Agriculture Information Access

ORGANIC AGRICULTURE INFORMATION ACCESS
AFSIC, NAL

Home Search Browse Bookbag Help

Basic Search

Basic Search Boolean Proximity Bibliographic History

Search in: full text

Find: organic

Tip : work* finds "worker," "working," etc.
war and peace finds "war and peace"

▼ More tips

Limit to: and in author

and in author

Search

Powered by DLXS
To comment or inquire about content, contact [UMDL Help](#)
To report errors, contact [UMDL Help](#)

Fig. 5.17 Búsqueda básica

Basic(Fig. 5.17): Realiza una búsqueda básica, la cual estará basada en el contenido de cada documento, es decir, es la búsqueda más general y por tanto la más prolongada en tiempo de procesamiento, a no ser que esta se vea limitada por los campos inferiores que tienen que ver con los autores, o bien limitados por la caja

combo superior, que limita la búsqueda en texto completo, título del texto o bibliográficamente.

Organic Agriculture Information Access - Mozilla Firefox (Navegación privada)

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://quod.lib.umich.edu/cgi/t/text/text-idx?cc=na1;cc=na1;page=proximity

Más visitados Comenzar a usar Firefox Últimas noticias

encriptar archivo de texto java - Bus... Organic Agriculture Information Access

ORGANIC AGRICULTURE INFORMATION ACCESS
AFSIC, NAL

Home Search Browse Bookbag Help

Proximity Search

Basic Boolean Proximity Search Bibliographic History

Search in: full text

Find: (near < 40)

near < 40

Tip: work* finds "worker," "working," etc.
war and peace finds "war and peace"

▼ More tips

Limit to: in author

and in author

Search

Powered by DLXS
To comment or inquire about content, contact [UMDL Help](#)
To report errors, contact [UMDL Help](#)

Fig. 5.18 Búsqueda por Proximidad

Proximity(Fig. 5.18): En esta búsqueda se encontrará una palabra que coincida con algunos de los caracteres introducidos en la búsqueda, no precisamente con la totalidad de estos, lo que genera una búsqueda aún más lenta en cuanto a tiempo de procesamiento.

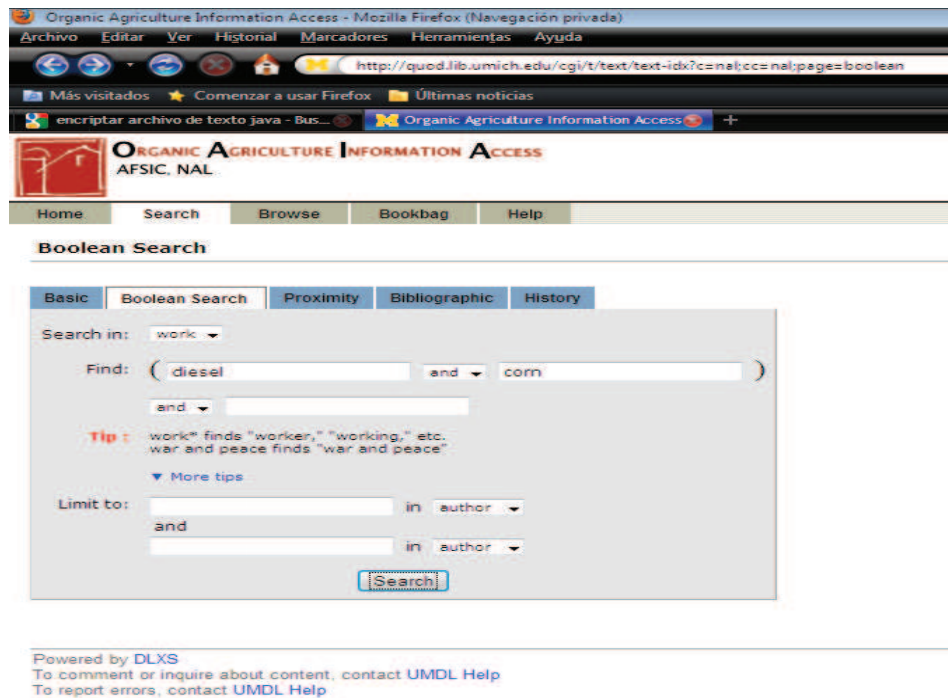


Fig. 5.19 Búsqueda booleana

Boolean(Fig. 5.19): Coincidencia exacta de los caracteres introducidos contra los encontrados en el documento.

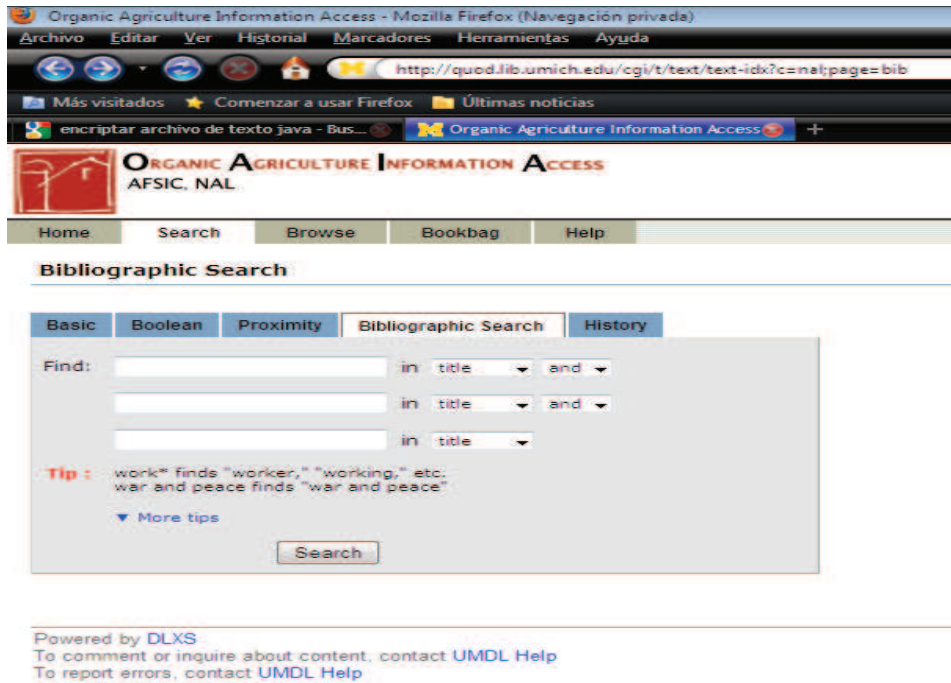


Fig. 5.20 Búsqueda bibliográfica

Bibliographic(Fig. 5.20): Se realiza una búsqueda bibliográfica, en la que únicamente se buscará por los autores.

Organic Agriculture Information Access - Mozilla Firefox (Navegación privada)

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://quod.lib.umich.edu/cgi/t/text/text-idx?c=naif:pages/history

Más visitados Comenzar a usar Firefox Últimas noticias

encriptar archivo de texto java - Bus... Organic Agriculture Information Access

ORGANIC AGRICULTURE INFORMATION ACCESS
AFSIC, NAL

Home Search Browse Bookbag Help

Search History

Basic Boolean Proximity Bibliographic Search History

Click on any search query in the history to resubmit that search.

	Search Query	Databases Selected	Total Results
1	title including "mccain", not sorted	Organic Agriculture Information Access	0 matching records
2	work including "diesel", and including "corn", not sorted	Organic Agriculture Information Access	match in 1 record
3	"organic" within full text, not sorted	Organic Agriculture Information Access	1013 matches in 68 records
4	"McCain" within full text, not sorted	Organic Agriculture Information Access	0 matches in 0 records

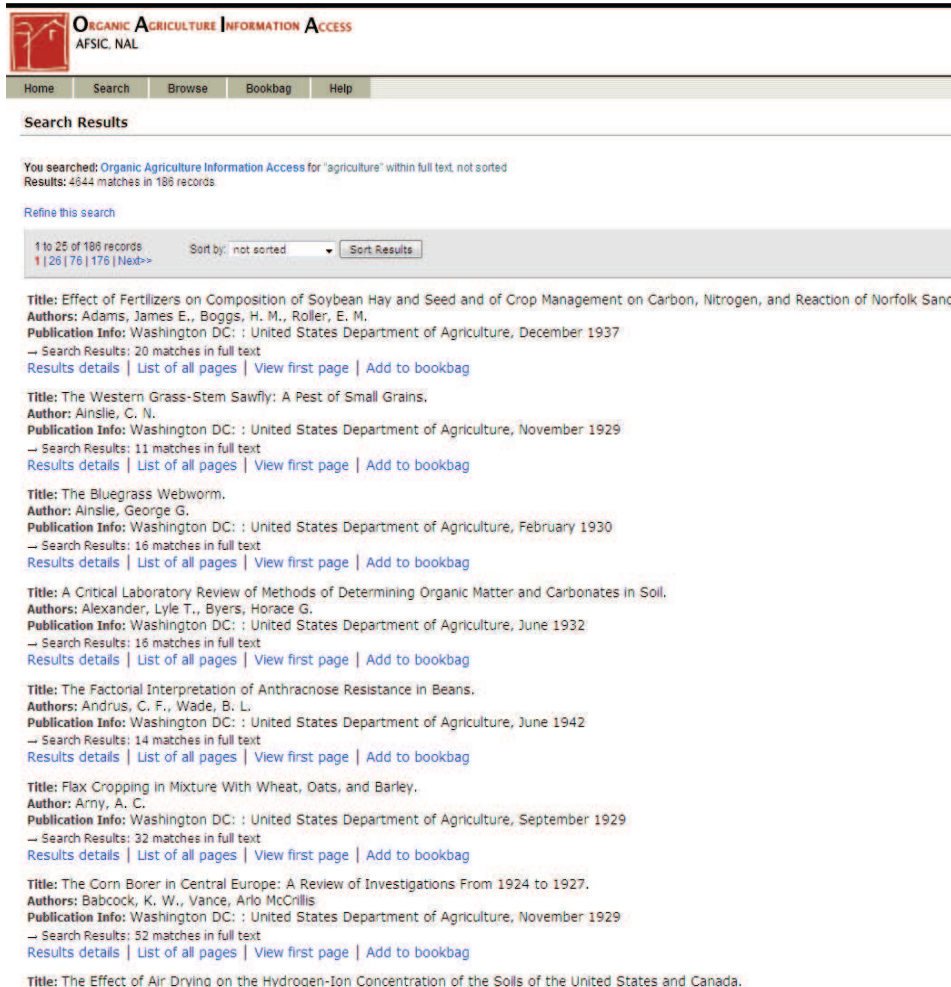
Powered by DLXS
To comment or inquire about content, contact UMDL Help
To report errors, contact UMDL Help

5.21 Historial de búsquedas

Aún cuando no aparece como link en la página inicial, dentro de las búsquedas vistas, encontramos otra opción más, la cual es la de búsqueda histórica (Fig. 5.21). Esta búsqueda proporciona los resultados de todas las consultas realizadas en este sistema, dando como resultado el texto buscado, las bases de datos seleccionadas y los resultados totales obtenidos en cada una.

A continuación observamos el resultado que arroja la búsqueda de la palabra agricultura (Fig. 5.22), se presentan varios documentos, todos ordenados por autor alfabéticamente, conteniendo el título del documento, los autores, lugar de la publicación, fecha y el número de coincidencias que tiene la búsqueda en el documento.

5.8.2 Resultados



ORGANIC AGRICULTURE INFORMATION ACCESS
AFSIC, NAL

Home Search Browse Bookbag Help

Search Results

You searched: Organic Agriculture Information Access for 'agriculture' within full text, not sorted
Results: 4844 matches in 188 records

Refine this search

1 to 25 of 188 records
1 | 26 | 76 | 176 | Next->

Sort by: not sorted

Title: Effect of Fertilizers on Composition of Soybean Hay and Seed and of Crop Management on Carbon, Nitrogen, and Reaction of Norfolk Sand.
Authors: Adams, James E., Boggs, H. M., Roller, E. M.
Publication Info: Washington DC: : United States Department of Agriculture, December 1937
→ Search Results: 20 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: The Western Grass-Stem Sawfly: A Pest of Small Grains.
Author: Ainslie, C. N.
Publication Info: Washington DC: : United States Department of Agriculture, November 1929
→ Search Results: 11 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: The Bluegrass Webworm.
Author: Ainslie, George G.
Publication Info: Washington DC: : United States Department of Agriculture, February 1930
→ Search Results: 16 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: A Critical Laboratory Review of Methods of Determining Organic Matter and Carbonates in Soil.
Authors: Alexander, Lyle T., Byers, Horace G.
Publication Info: Washington DC: : United States Department of Agriculture, June 1932
→ Search Results: 16 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: The Factorial Interpretation of Anthracnose Resistance in Beans.
Authors: Andrus, C. F., Wade, B. L.
Publication Info: Washington DC: : United States Department of Agriculture, June 1942
→ Search Results: 14 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: Flax Cropping in Mixture With Wheat, Oats, and Barley.
Author: Army, A. C.
Publication Info: Washington DC: : United States Department of Agriculture, September 1929
→ Search Results: 32 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: The Corn Borer in Central Europe: A Review of Investigations From 1924 to 1927.
Authors: Babcock, K. W., Vance, Arlo McGillis
Publication Info: Washington DC: : United States Department of Agriculture, November 1929
→ Search Results: 52 matches in full text
[Results details](#) | [List of all pages](#) | [View first page](#) | [Add to bookbag](#)

Title: The Effect of Air Drying on the Hydrogen-Ion Concentration of the Soils of the United States and Canada.

Fig. 5.22 Resultados Agricultura Orgánica

5.9 Comparación entre los distintos proyectos

	Componentes de control	Componentes gráficos	Opciones de búsqueda	Actividad por parte del usuario	Resultados de las búsquedas
ToolBDH	Cajas combo, cajas de texto botones ok,	Fondo de colores, iconos acorde al programa, texto de distintos colores	Navegación a través de la tabla de datos	Dependiendo de los privilegios puede introducir o no datos mediante los componentes de control	Aparece en los mismos componentes de control en los que se introduce la información
Secretaría Parlamentaria	Cajas combo, cajas de texto, listas de chequeo, botones ok	Fondo de color, texto en colores, subrayado del texto	Navegación a través de links o mediante la ejecución en los componentes de control	Introducción de la información para la búsqueda	Aparecen en una nueva página con forma de documento, enlistando las características

Asasve	Caja combo, links	Varios colores, diseño muy vistoso de los documentos, enlaces, publicidad, calendario	No existen	Navegar por la caja combo	Carteles bien realizados y texto informativo en distintas páginas
Faolex	Listas de chequeo, cajas combo, cajas de texto, botones ok	Sombreado de algunos componentes y texto en negritas	Búsqueda en los documentos y ejecución de comandos mediante "isis ql"	Introducción de texto y selección de opciones en las cajas combo	Aparición de los títulos de los documentos, con opción a ser descargados

Gobierno de Aragón	Cajas de texto y botones radio	Texto en negritas, una vez en los resultados, cambio de color del texto, imágenes e íconos	Búsqueda del texto en los documentos almacenados	Introducción de texto	Títulos de los documentos, con opción a ver las convocatorias
Índices de propagación	Cajas combo, botones ok	Uso de colores de fondo así como íconos y color en los textos	Búsqueda de resultados por fechas	Selección de fechas en cajas combo	Desplegado de resultados por fecha, identificados mediante el cambio de color en los textos
Agricultura Orgánica	Caja de texto, links de búsqueda especializada	Imágenes, texto en colores, fondo en dos colores	Basic, proximity, boolean, bibliographic	Introducir texto de búsqueda	Colección de resultados organizados de forma alfabética por autor, mostrando título

Tabla de comparación entre los distintos proyectos

6. Infraestructura

El tipo de infraestructura o bien arquitectura, se debe de elegir pensando en el sistema a ejecutar, las necesidades que este tendrá, el tipo de usuario que lo utilizará, la seguridad que se le asignará, en donde se ejecutará y la distribución que tendrá, es por eso que se tiene que tener en cuenta todos los tipos de arquitectura, o al menos las fundamentales.

Las conexiones en redes de las computadoras permiten que diversas tareas sean ejecutadas en sistemas servidores y que otras tantas se ejecuten en sistemas clientes. Lo cual ha desembocado en el desarrollo de sistemas de bases de datos cliente-servidor.

- El procesamiento paralelo en una computadora permite acelerar las actividades del sistema de base de datos, proporcionando a las transacciones unas respuestas más rápidas así como la capacidad de ejecutar más transacciones por segundo. Las consultas pueden procesarse de manera que se explote el paralelismo ofrecido por el sistema informático subyacente. La necesidad del procesamiento paralelo de consultas ha conducido al desarrollo de los sistemas de bases de datos paralelos.

6.1 Base De Datos Centralizada

Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas. Por otro lado, los sistemas cliente-servidor tienen su funcionalidad dividida entre el sistema servidor y múltiples sistemas clientes.

6.1.1 Sistemas centralizados

Una computadora de propósito general consiste en una o unas cuantas unidades centrales de procesamiento y un número determinado de controladores para los dispositivos que se encuentran conectados a través de un bus común, el cual proporciona acceso a la memoria compartida (unidades centrales de procesamiento) poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos.

Se distinguen dos formas de utilizar las computadoras:

En sistemas monousuario o multiusuario. En la primera categoría están las computadoras personales y las estaciones de trabajo.

Un sistema monousuario típico es una computadora de escritorio utilizada por una única persona que dispone de un solo CPU, de uno o dos discos fijos y que trabaja con un sistema operativo que sólo permite un único usuario. Por el contrario, un sistema multiusuario típico tiene más discos y más memoria, puede disponer de varios CPU y trabaja con un sistema operativo multiusuario. Se encarga de dar servicio a un gran número de usuarios que están conectados al sistema a través de terminales.

6.2 Base de datos descentralizada

La administración de bases de datos centralizada presenta problemas como:

*Degradación del desempeño provocado por un número creciente de ubicaciones remotas a mayores distancias.

* Costos altos asociados con el mantenimiento y operación de grandes sistemas de

bases de datos centrales (mainframe).

* Problemas de confiabilidad creados por la dependencia en un sitio central.

El ambiente de negocios dinámico y las deficiencias de las bases de datos centralizados demandaron aplicaciones que permitieran el acceso a los datos de diferentes fuentes de múltiples ubicaciones. Ese ambiente de base de datos de fuentes y ubicaciones múltiples, conocido como base de datos distribuida, es manejado por un sistema de administración de base de datos distribuidos.

6.2.1 Procesamiento descentralizado de datos

* El desarrollo de las mini y las microcomputadoras; la tecnología de bases de datos, y el surgimiento de los sistemas de Procesamiento Distribuidos de Datos (DDP). Estos tres fenómenos están interrelacionados porque las mini y las microcomputadoras son una buena medida de la factibilidad económica del DDP, y las bases de datos pueden "distribuirse" entre los sistemas de cómputo para mejorar la efectividad de los sistemas DDP.

6.3 Atributos de los modelos centralizados y descentralizados

Centralizadas

Uniformidad del sistema

Uniformidad de hardware

Disponibilidad de información siempre completa

Centralización de seguridad

Mantenimiento más barato

Descentralizadas

Incremento de seguridad para parcelación

Minimiza la dependencia tecnológica

Evita la selección adversa d proveedores

Menor inversión inicial

Disponibilidad de información en función de su utilidad

6.4 Base De Datos Distribuidas

En un sistema distribuido de bases de datos se almacena la base de datos en varias computadoras. Varios medios de comunicación, como las redes de alta velocidad o las líneas telefónicas, son los que pueden poner en contacto las distintas computadoras de un sistema distribuido. No comparten ni memoria ni discos. Las computadoras de un sistema distribuido pueden variar en tamaño y función pudiendo abarcar desde las estaciones de trabajo a los grandes sistemas.

Los sistemas distribuidos de bases de datos consisten en sitios débilmente acoplados que no comparten ningún componente físico. Además, puede que los sistemas de bases de datos que se ejecutan en cada sitio tengan un grado sustancial de independencia mutua. Lo que ocasionará que existan datos en un sitio que en otro no.

Hay varias razones para construir sistemas distribuidos de bases de datos, incluyendo el compartimiento de los datos, la autonomía y la disponibilidad.

- Datos compartidos. La principal ventaja de construir un sistema distribuido de bases de datos es poder disponer de un entorno donde los usuarios puedan acceder desde una única ubicación a los datos que residen en otras ubicaciones.

Autonomía. La principal ventaja de compartir datos por medio de distribución de datos es que cada ubicación es capaz de mantener un grado de control sobre los datos que se almacenan localmente.

En un sistema centralizado, el administrador de bases de datos de la ubicación central controla la base de datos. En un sistema distribuido, existe un administrador de bases de datos global responsable de todo el sistema.

Disponibilidad. Si un sitio de un sistema distribuido falla, los sitios restantes pueden seguir trabajando.

En particular, si los elementos de datos están replicados en varios sitios, una transacción que necesite un elemento de datos en particular puede encontrarlo en varios sitios.

El principal inconveniente de los sistemas distribuidos de bases de datos es la complejidad añadida que es necesaria para garantizar la coordinación apropiada entre los sitios. Esta creciente complejidad tiene varias facetas:

- Costo de desarrollo del software: La implementación de un sistema distribuido de bases de datos es más difícil y, por lo tanto, más costoso.

- Mayor probabilidad de errores: Como los sitios que constituyen el sistema distribuido operan en paralelo es más difícil asegurarse de la corrección de los

algoritmos, del funcionamiento especial durante los fallos de parte del sistema así como de la recuperación. Son probables errores extremadamente sutiles.

- Mayor sobrecarga de procesamiento: El intercambio de mensajes y el cómputo adicional necesario para conseguir la coordinación entre los distintos sitios constituyen una forma de sobrecarga que no surge en los sistemas centralizados.

Beneficios a la sociedad

Mediante la realización de este proyecto, se propone que la búsqueda de información para los estudiantes y profesores, en cuanto a proyectos y desarrollo de software, se vea optimizada, al desarrollar un sistema que permita a cada uno de los usuarios tener acceso libre para la consulta de información sobre algún proyecto en específico. Mediante las características que se proponen, el sistema ofrecerá, en primer lugar a los administradores (que pueden ser profesores, alumnos, o personal de confianza), poder ir depurando en cualquier momento los proyectos almacenados, mediante la eliminación, adición, actualización de los documentos depositados. Los usuarios con permisos superiores serán aquellos capaces de poder introducir los proyectos en forma directa, únicamente se requiere que se introduzcan los elementos pertinentes en la base de datos, los cuales, el mismo programa los indicará. Así mismo existirá un tercer tipo de usuario, el cual solo podrá acceder al sistema para ir viendo la información que en él se encuentre, este usuario no podrá realizar ningún tipo de modificación a ninguno de los proyectos almacenados.

Por tanto, con este sistema lo que se espera es incrementar la eficiencia en cuanto a las búsquedas realizadas, logrando un mejoramiento en los tiempos de búsqueda, facilidad para localizar a los autores, evitar repeticiones innecesarias de los trabajos, buscar el perfeccionamiento del nivel de software desarrollado en la Facultad de Ingeniería, generar más y mejores desarrollos dentro del colegio, tener una mayor participación de los estudiantes en el proceso de calidad del software, dar experiencia a los alumnos en cuanto al correcto desarrollo de un producto de software, crear un portal en el que se destaque el trabajo realizado en la Facultad de Ingeniería y que sirva de publicidad para la misma, tanto en México como en el mundo.

En esta justificación se realiza un desarrollo de un trabajo similar al que se espera poner en marcha en el siguiente trabajo que será sobre el desarrollo de la base de datos históricos para proyectos de software, en el trabajo final se tiene cometido que se implemente el sistema en un entorno web que vaya almacenando las contraseñas en una base de datos para que nos de la suficiente seguridad y confianza en el manejo de datos, lo cual traerá a la comunidad estudiantil la posibilidad de realizar las consultas desde el lugar de su preferencia.

Con este proyecto la sociedad contará además con la posibilidad de consultar proyectos que pueden ser muy antiguos, ya que al ser una base de datos históricos, se pretende que los proyectos más antiguos puedan estar dentro de esta, esto será posible con la ayuda de académicos que conserven proyectos de generaciones anteriores, no importando en que lenguaje hayan sido realizados, siendo así que nuestro sistema no está limitado a ún lenguaje de programación.

Conclusiones:

El presente documento se basa en la factibilidad y justificación de realizar una base de datos histórica para proyectos de software.

El trabajo no es limitado a una base de datos, el trabajo, como se menciona, requiere de una plataforma que permita la realización correcta de consultas, actualizaciones, eliminaciones y creaciones de proyectos de software. En el documento se realiza una simulación de lo que se pretende sea el trabajo ya finalizado, más no por eso, con la simulación es suficiente para liberar el trabajo, se pretende que en un posterior trabajo se implemente de manera total este trabajo. Por medio de la simulación realizada se ha podido definir que el entorno de ejecución más adecuado para la puesta en marcha del sistema debe ser un ambiente Web, pues este no está limitado a un Sistema Operativo, permite el acceso desde múltiples ubicaciones, es una de las formas más accesible para la comunidad en general, permitirá buscar la información más reciente, pues al tratarse de un ambiente web se mantendrá actualizado en cuanto ocurra algún cambio. Es muy importante que este trabajo pueda ser accesible desde varios lugares y que la cantidad de usuarios con acceso a él, tengan roles bien definidos en cuanto a permisos y accesos, pues de esta forma aseguraremos una base de datos.

El mejor entorno, es, por su facilidad en cuanto al acceso desde distintos equipos, Internet, debido a que de ejecutarse bien, permite actualizaciones, respaldos y la posibilidad de acceder desde muchas plataformas, desde equipos de escritorio hasta teléfonos celulares, y es muy difícil encontrar una aplicación tan práctica en forma independiente, si se continuara en ambiente local, en un periodo no muy largo, el proyecto seguramente terminará en el olvido, sin embargo, en ambiente Web y de facilitar los accesos, su ciclo de vida se vería incrementado incluso de forma exponencial.

Utilizar un ambiente como el de internet para desplegar la aplicación permitiría a la comunidad el poder realizar consultas y estudiar, a partir de los otros proyectos que se encuentren en la base de datos, con lo que se mejorará la calidad de las entregas de trabajos académicos y se evitará pérdida de tiempo en cuanto a la implementación de ejemplos que en cada semestre se tienen que repetir por alguna razón.

El desarrollo de un proyecto en ambiente local, en el presente trabajo, permite ver cuáles son los aspectos a mejorar, la factibilidad de realizar el proyecto propuesto, pero sobre todo, da un muy buen panorama de cómo sería la programación que se implementará; sobre todo si se elige realizar el desarrollo en un lenguaje como Java, más específicamente en un ambiente que utilice J2EE, que facilitará el uso de las bases de datos, los accesos a esta y los ambientes Web, así como el diseño y las actualizaciones de la aplicación, o bien su migración a lenguajes más modernos y avanzados que los actuales, de tal forma que el ciclo de vida de la aplicación puede ser tan duradero como cualquier otra página que sea implementada por la misma Facultad de Ingeniería y cuya implementación ayudará mucho a las futuras generaciones de alumnos, lo mismo que facilitará la labor del personal académico.

Mesografía:

<http://www.seoprofesional.com/infonetics-indica-un-crecimiento-del-software/>

<http://www.idg.es/computerworld/El-mercado-de-software-empresarial-registrara-un-c/seccion-mercado/noticia-78728>

http://economictimes.indiatimes.com/Infotech/Software_BPO_industry_growth_will_slow_down_in_2009_Nasscom/articleshow/3215690.cms

<http://www.globaltimes.cn/www/english/sci-edu/china/2009-08/460037.html>

<http://www.forrester.com/ER/Press/Release/0,1769,1241,00.html>

<http://www.idg.es/dealerworld/%28S%28jolewceuft4sutvyx1i5xf45%29%29/Noticia.aspx?idn=83661&seccion=mercado>

http://www.gartner.com/it/section.jsp?type=press_releases&format=xhtml&year=2009&show_archived=true

<http://www.fernandoflores.cl/node/467>

http://www.thaindian.com/newsportal/business/nasscom-pegs-revenues-of-indian-software-industry-at-60-bn-lead_100150974.html

<http://spanish.mofcom.gov.cn/aarticle/reportajeexterior/200907/20090706422079.html>

<http://www.computerdictionaries.org/software>

http://what-is-what.com/what_is/software.html

http://www.ochocolumnas.com.mx/2009/08/NFI090819_06.html

<http://www.infochannel.com.mx/6-24573/retrocede-mexico-en-indicador-de-la-sociedad-de-la-informacion>

Índice de imágenes:

Fig.1.1 Tabla CU-01	23
Fig.1.2 Tabla CU-02	26
Fig.1.3 Tabla CU-03	30
Fig.4.1 Selección de proyecto en Netbeans	54
Fig.4.1.1 Nombre y localización del proyecto	54
Fig.4.1.2 Presentación básica del proyecto	55
Fig.4.1.3 Propiedades de los controles Swing	56
Fig.4.1.4 JLabel “Usuario” junto con un jPasswordField “Contraseña”	57
Fig.4.1.5 jButton “Crear”	57
Fig.4.1.6 Ventana principal	58
Fig.4.1.7 Pestaña Source	59
Fig.4.1.8 Localización del código	59
Fig.4.2 Ventana automatizada para bases de datos	69
Fig. 4.3 Ventana modificada	70
Fig. 4.4 Cambios de jTextField por jTextArea	71
Fig. 4.5 Administración de componentes	72
Fig.4.6 Fuente del programa	73
Fig. 4.7 Selección de un JFrame	76
Fig. 4.8 Programa de pruebas	77
Fig. 5.1 Proyectos y soluciones	79
Fig. 5.2 Proyectos y soluciones, administración BD	80
Fig. 5.3 Secretaria Parlamentaria	81
Fig. 5.4 Proyectos encontrados	82
Fig. 5.5 Búsqueda general	83
Fig. 5.6 Documentos encontrados	83
Fig. 5.7 Página principal	84

Fig. 5.8 Ejemplo de documentos encontrados	86
Fig. 5.9 Búsqueda general Faolex	87
Fig. 5.10 Resultados Faolex	89
Fig. 5.11 Búsqueda Isis ql	90
Fig. 5.12 Portal Aragón	91
Fig. 5.13 Resultados Aragón	92
Fig. 5.14 Índices de propagación	94
Fig.5.15 Resultados Índices de propagación	95
Fig. 5.16 Acceso a la información de agricultura orgánica	96
Fig. 5.17 Búsqueda básica	97
Fig. 5.18 Búsqueda por Proximidad	98
Fig. 5.19 Búsqueda booleana	99
Fig. 5.20 Búsqueda bibliográfica	100
Fig.5.21 Historial de búsquedas	101
Fig. 5.22 Resultados agricultura orgánica	102

Índice de tablas

Tabla cu2id	27
Tabla idCu3	30
Comparación entre los distintos proyectos	103

Bibliografía:

Fundamentos De Bases De Datos, Abraham Silberschatz, Henry F. Korth, Bell Laboratories, S. Sudarshan, Instituto Indio de Tecnología, Bombay.

Sistemas de bases de datos: Diseño, implementación y administración - Página 486

Peter Rob, Carlos Coronel 2004 838p

Beginning with MySQL, Robert Sheldon and Geoff Moes

Wiley publishing Inc

Estados Unidos 2005, 865p

Diseño conceptual de bases de datos, de Carlo Batini, Stefano Ceri, Shamkant B. Navathe

Addison-Wesley Iberoamericana 1994

Ingeniería de software, Sommerville Ian

Pearson Educación S.A. 2005 712p

Software Engineering: A Practitioner's Approach, Pressman Roger

McGraw-Hill Higher Education, 2009 - 928 páginas

Software Creativity 2.0, Robert L. Glass, Tom DeMarco

developer.* Books, 2006 - 484 páginas

Diseño de bases de datos objeto-relacionales con UML, Esperanza Marcos, Esperanza Marcos, Esperanza Marcos Belén Vela Juan M. Vara, Belén Vela, Juan M. Vara

Librería-Editorial Dykinson, 2005 - 70 páginas

Java in a nutshell, David Flanagan

O'Reilly Media, Inc., 2005 - 1224 páginas

Los sistemas de información en la sociedad del conocimiento, Fernando Giner de la Fuente

ESIC Editorial, 2004 - 215 páginas

Informática y comunicaciones en la empresa, Carmen de Pablos Heredero

ESIC Editorial, 2004 - 316 páginas

Apple Inc., Jason D. O'Grady

ABC-CLIO, 2008 - 185 páginas

Steve Jobs: The Brilliant Mind Behind Apple, Anthony Imitola

Gareth Stevens, 2009 - 112 páginas

Enterprise Cloud Computing: Technology, Architecture, Applications, Gautam Shroff

Cambridge University Press, 2010 - 308 páginas

Mobile Methods, Monika Büscher, John Urry, Katian Witchger

Taylor & Francis, 2010 - 224 páginas

Advances in Multimedia and Network Information System Technologies, Ngoc-Thanh Nguyen, Aleksander Zgrzywa, Andrzej Czyzewski

Springer, 2010 - 340 páginas

The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad, Michael A. Cusumano

Simon and Schuster, 2004 - 352 páginas

Secrets of software success: management insights from 100 software firms around the world, Detlev J. Hoch

Harvard Business Press, 2000 - 312 páginas

SQL: The Complete Reference, Second Edition, James R Groff, Paul N. Weinberg

McGraw-Hill, 2002 - 1080 páginas

SQL server 2005: SQL, Transact SQL, Jérôme Gabillaud

Ediciones ENI, 2006 - 11 páginas

SQL: The Programming Language, Kirk Scott

Jones & Bartlett Learning, 2009 - 76 páginas

SQL: el lenguaje de consulta estructurado SQL, Carolyn J. Hursch, Gerardo Martín González, Jack L. Hursch, Ramón M. Chordá

Ra-Ma, 1989 - 209 páginas

Database--principles, programming, and performance, Patrick O'Neil, Elizabeth O'Neil

Morgan Kaufmann, 2001 - 870 páginas

Diseño web: guía de referencia, Jennifer Niederst Robbins

Anaya Multimedia, 2007 - 928 páginas