



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

Unidad Académica Profesional Nezahualcóyotl

**“SISTEMA PARA LA EXTRACCIÓN AUTOMÁTICA DE INFORMACIÓN EN
ARCHIVOS DE TEXTO PLANO GENERADOS POR BASES DE DATOS”**

T E S I S

Para obtener el título de:

INGENIERA EN SISTEMAS INTELIGENTES

PRESENTA:

MARÍA ELENA MARTÍNEZ LÓPEZ

DIRECTOR DE TESIS:

DRA. DORA MARIA CALDERON NEPAMUCENO



Ciudad Nezahualcóyotl, Abril 2015.

Índice

Índice.....	2
Índice de figuras.....	3
Resumen	4
Antecedentes de la temática.	6
Planteamiento del problema.	9
Objetivo general.....	12
Objetivos específicos	12
Marco teórico.....	13
Bases de datos	13
Bases de Datos Distribuidas.....	20
Arquitectura de las bases de datos.....	31
Arquitectura de BDD.....	36
Programación en lenguajes HTML, Mysql y Java	38
Lenguaje HTML	39
Lenguaje SQL.....	47
Lenguaje Java	61
Sistemas Operativos.....	69
Minería de datos	78
Métodos y técnicas de investigación empleadas.	83
Presentación y discusión de resultados.....	85
Conclusiones y sugerencias.....	87
Terminología y Glosario	89
VII Bibliografía	92
Anexos.....	93
Anexo I Requerimientos para acceder a Control Escolar	93
Anexo 2 Manual de usuario e instalación.....	97
Agradecimientos.	99

Índice de figuras

Figura 1 Interfaz de acceso a Control Escolar	9
Figura 2 Ejemplo de Diagrama de modelo relacional.	15
Figura 3. Ampliación del catálogo de una arquitectura de BDD	34
Figura 4. Tres niveles de la arquitectura de bases de datos.	35
Figura 5. Esquema de arquitectura de BDD	37
Figura 6. Estructura general de una línea de código en el lenguaje de etiquetas HTML.	41
Figura 7. Ejemplo de código HTML con coloreado de sintaxis.	42
Figura 8. Se ejecuta el Sistema.....	85
Figura 9. Ingresar ubicación del archivo.	85
Figura 10 Resultado final.	86
Figura 11. Archivo de entrada.....	97
Figura 12. Archivo procesado el por el sistema	98

Resumen

El **“Sistema para la extracción automática de información en archivos de texto plano generados por bases de datos”**, es un sistema que consta básicamente de dos fases. En la primera fase depura los archivos de tags (separadores) y otros símbolos que son generados en la base de datos los cuáles tienen un formato bien estructurado para ser enviado a una impresora, pero que carecen de relevancia al momento de ser visualizados en un formato de texto plano es decir, no ordenado, sin saltos de línea y que por sus características, algunos signos ni si quiera se visualizan, tampoco son candidatos a ningún tipo de edición. En esta primera fase se prepara el archivo para posteriormente, en la segunda fase importarlo desde el mismo sistema a un archivo con diagramación que si permita la libre manipulación del documento.

Éste sistema fue diseñado para auxiliar en el área de Control Escolar de la Unidad Académica Profesional Nezahualcóyotl, departamento que al inicio de cada período escolar suministra las listas que utilizarán los docentes durante todo el curso para registrar sus evaluaciones así como las asistencias de los alumnos. Se estima que el tiempo promedio invertido para todo el proceso incluyendo la descarga, exportación y edición de cada lista es de 15 minutos, considerando que se ofertan 4 licenciaturas diferentes en dos turnos, y un promedio de 30 UA en cada una, podemos calcular que se trabaja sobre un total de 240 listas, lo que requiere un mínimo de 60 horas de trabajo.

Aunque este programa fue especialmente creado para auxiliar en una tarea determinada en el departamento de Control Escolar también puede usarse en otras áreas que requieran de un pre-procesamiento de datos para su posterior exportación a otra extensión. Por ejemplo tiene una gran utilidad en el área contable debido a que usualmente los manejadores de

bases de datos importan los archivos con extensión .txt, pero es necesario manipular los datos a través de un programa que contenga funciones matemáticas como lo es Excel.

Es cierto que Excel ya cuenta con una herramienta para la exportación e importación de archivos de otra extensión, sin embargo es necesario que el archivo que se va a exportar contenga la estructura del archivo final, es decir el número concreto e idéntico de columnas ya ordenadas, porque así se almacenará con ese mismo formato todo el documento.

Sin embargo, en el caso de las listas de asignaturas dicha herramienta no es eficaz porque éstas contienen un encabezado en un formato de alineación centrado que contiene un logotipo, así como el nombre de la Universidad, el nombre de la Unidad Académica, el nombre de la materia, el número de asignatura, el nombre del profesor y el período, por lo tanto se desconfigura al aplicarle el mismo formato de distribución de columnas que es necesario para el resto del documento.

Tomando en cuenta los requerimientos que se establecen en el programa de Control Escolar de la UAEM para poder ejecutarse, así como también la velocidad del proveedor del servicio de internet y las herramientas con las que cuenta Excel para dicho proceso, resulta un proceso muy costoso en tiempo. Por lo anteriormente expuesto se busca reducir significativamente, por medio de un sistema auxiliar, el tiempo y trabajo invertido hasta ahora en esta tarea, al menos en un 80% debido a que aún queda la limitante de la velocidad de descarga de los archivos. Al automatizar este proceso se espera como consecuencia inmediata agilizar la entrega de las listas a las coordinaciones correspondientes.

Antecedentes de la temática.

Los requerimientos oficiales que solicita el portal de la Universidad Autónoma del Estado de México para poder acceder a la información de Control Escolar son muy rígidos y no tienen mucha disponibilidad, incluso su uso ha sido discontinuado como la versión de internet Explorer 6 que condiciona a su vez el Sistema Operativo a Windows XP siendo su última versión estable la del 20 de abril de 2008 hace casi 7 años, sin embargo debido a la gran cantidad de usuarios que aún usaban esta versión, Microsoft ofreció soporte técnico para Windows XP hasta el 8 de abril de 2014. (Microsoft, support.office, 2014). Por otro lado Adobe Reader 4.x tampoco es vigente, esto complica sumamente la tarea de realizar el proceso de descarga de la información necesaria para generar las listas de asistencia y calificaciones de los docentes.¹

En las terminales del departamento de Control Escolar se hicieron las adecuaciones necesarias para poder instalar el Sistema Operativo Windows XP, que permitiera el uso de Explorer 6 así como del Adobe Reader 4.x. La consecuencia fue evidente: se “alentó” demasiado cada proceso que realizan estas terminales. Si además recordamos que el proveedor de servicios de internet es de baja calidad, cada acción en ese departamento es un verdadero caos.

Actualmente existen algunos programas como lo es Minitab17 que se desarrolló en Estados Unidos el cuál realiza además de exportación de archivos, análisis estadístico completo incluyendo gráficas e interpretación para un adecuado uso de los datos, es un tarea completa e incluso sobrada para obtener las 240 listas requeridas, y muy costoso pues se ha de pagar una renta mensual de \$29.99 dólares (aproximadamente \$450 pesos dependiendo del tipo de cambio vigente al corte del servicio).

¹Esta acción se describe con más detalle en el Anexo I.

Además, este sistema y otros similares que se venden y/o rentan requieren para garantizar su buen funcionamiento, en ocasiones extralimitado a los intereses de la UAP, de un importante espacio de almacenamiento disponible para alojar los archivos que lo conforman, así como un sistema operativo más reciente y evidentemente un servicio de internet de banda ancha. De otro modo sería absolutamente infructuoso pagar un servicio que no se explota.

Excel forma parte de la paquetería básica de Office que utiliza Microsoft y no tiene ningún costo adicional su uso. Excel tiene una herramienta para exportar e importar archivos de una extensión diferente, esta herramienta funciona cuando el documento que se exporta ya tiene un formato final deseado, es decir, aplicará a todo el documento el número exacto de columnas que previamente se estableció.

¿Por qué entonces los gestores de bases de datos no exportan nunca las consultas directamente a un archivo de Excel? Principalmente por dos grandes razones: incompatibilidad en la capacidad de almacenamiento e incompatibilidad en algunos tipos de datos permitidos en cada uno de los programas.

Las hojas de un libro de Excel tienen un número limitado de filas y no todas las consultas que se realizan en una base de datos podrán almacenarse dentro de este rango, porque el tamaño de las tablas tiene una capacidad de almacenamiento mayor, lo que causa un error "out file system" o fuera de rango, algo que evidentemente el usuario no sabría cómo resolver. No en todos los casos se da este fenómeno de desbordamiento pero es algo que previene sql.

Por otro lado, cuando se crean los campos que formarán las tablas de las bases de datos, se determinan los tipos de datos y extensión para cada campo. En SQL Server, cada columna,

variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de dato es un atributo que especifica el tipo de dato que el objeto puede contener: datos de enteros, datos de caracteres, datos de moneda, datos de fecha y hora, cadenas binarias, etc. SQL Server proporciona un conjunto de tipos de datos del sistema que define todos los tipos de datos que pueden utilizarse con SQL Server. También puede definir sus propios tipos de datos en Transact-SQL o Microsoft .NET Framework. Los tipos de datos de alias están basados en los tipos de datos proporcionados por el sistema. (Microsoft A. , 2012)

Sin embargo no todos los tipos de datos permitidos en sql tienen compatibilidad con los tipos de datos usados en Excel.

Planteamiento del problema.

El Departamento de Control Escolar de la Unidad Académica Profesional Nezahualcóyotl se encarga de generar las listas de asistencia de alumnos de cada período escolar mediante software desarrollado con tecnología actualmente obsoleta, el cual ha degradado la rapidez, eficiencia y productividad de los usuarios que interactúan con ésta herramienta.

El proceso se inicia con la inscripción al período en curso que es individual y cada alumno la realiza en un horario y fecha determinada que publica Control Escolar la cual genera priorizando los mejores promedios de cada uno de los planes de estudios que se ofertan en la Unidad Académica Profesional Nezahualcóyotl, todos los registros se almacenan en la base de datos de la UAEM.

La dirección electrónica para realizar la inscripción es:
<https://controlescolar.uaemex.mx/dce/sicde/Inscripciones/indexLogin.html>



The screenshot shows a web browser window with the URL <https://controlescolar.uaemex.mx/dce/sicde/Inscripciones/indexLogin.html>. The page header includes the UAEM logo and the text "Universidad Autónoma del Estado de México Sistema de Control Escolar". A message states: "Su sesión ha expirado o aún no se ha firmado Firmese nuevamente". Below this is the heading "Bienvenido al Sistema de Inscripciones en línea".

Campo	Valor
Tipo de usuario*	Alumno
Número de cuenta/Usuario*	
UNIP/Contraseña*	

Olvidé mi contraseña

Ver 2015.01.15B

Requerimientos de Visualización

- Guía
- Acrobat Reader Versión 4.x

Figura 1 Interfaz de acceso a Control Escolar

Al ingresar los alumnos con su número de cuenta y usuario, se despliegan las asignaturas que se ofertan en el presente ciclo escolar y que no han sido cursadas y/o aprobadas por el alumno, también puede ver los horarios y profesores que las impartirán. Los alumnos tienen un lapso de 15 minutos para concluir su inscripción antes de que su sesión caduque.

Durante el lapso aproximadamente de 5 días hábiles en un horario de 08:00 horas a 21:00 horas, se realiza reiteradamente este mismo trámite por todos los usuarios de la Universidad Autónoma del Estado de México.

Posteriormente, personal de Control Escolar de cada una de las Unidades Profesionales, Centros Universitarios, Facultades, etc. mediante el Portal designado para dicho fin (<https://controlescolar.uaemex.mx/dce/sicde/Inscripciones/indexLogin.html>) inicia un proceso de recuperación de registros correspondientes a cada uno de sus espacios académicos, en total 39:

Facultades:

- Facultad de Antropología
- Facultad de Arquitectura y Diseño
- Facultad de Artes
- Facultad de Ciencias
- Facultad de Ciencias Agrícolas
- Facultad de Ciencias de la Conducta
- Facultad de Ciencias Políticas y Sociales
- Facultad de Contaduría y Administración
- Facultad de Contaduría y Administración (los Uribe)
- Facultad de Derecho
- Facultad de Economía
- Facultad de Enfermería y Obstetricia
- Facultad de Geografía
- Facultad de Humanidades
- Facultad de Ingeniería
- Facultad de Lenguas
- Facultad de Medicina
- Facultad de Medicina Veterinaria y Zootecnia
- Facultad de Odontología

- Facultad de Planeación Urbana y Regional
- Facultad de Química
- Facultad de Química (unidad el Cerrillo)
- Facultad de Turismo y Gastronomía (el Rosedal)
- Facultad de Turismo y Gastronomía

Centros Universitarios:

- Centro Universitario Uaem Amecameca
- Centro Universitario Uaem Atlacomulco
- Centro Universitario Uaem Ecatepec
- Centro Universitario Uaem Temascaltepec
- Centro Universitario Uaem Tenancingo
- Centro Universitario UAEM Texcoco
- Centro Universitario UAEM Valle de Chalco
- Centro Universitario UAEM Valle de México
- Centro Universitario UAEM Valle de Teotihuacán
- Centro Universitario UAEM Zumpango

Unidades Académicas Profesionales:

- Unidad Académica Profesional Chimalhuacán
- Unidad Académica Profesional Cuautitlán Izcalli
- Unidad Académica Profesional Huehuetoca
- Unidad Académica Profesional Nezahualcóyotl
- Unidad Académica Profesional Tianguistenco

En el caso particular de la Unidad Académica Profesional Nezahualcóyotl, el Lic. Hugo Andrés Hernández Hernández descarga los archivos que contienen estos datos en un formato de “texto plano” (.txt) que se pueden abrir mediante un editor de texto tal como el bloc de notas que utiliza Windows por defecto. Luego para que ese archivo sea útil y pueda manipularse acorde a las necesidades de los docentes, se importa a un formato con diagramación que permite realizar los cambios adecuados como lo es Excel que tiene extensión .xls y también es parte de la paquetería de Office.

En la Unidad Académica Profesional Nezahualcóyotl se imparten cuatro licenciaturas diferentes, en dos turnos, con un promedio de 30 UA cada una. Esto da como resultado un total de 240 listas en cada período escolar, si el tiempo promedio para obtener cada una es de 15 minutos se necesitará al menos de 60 horas de trabajo de una persona. El Lic. Hugo Andrés Hernández Hernández se encarga de esta y otras actividades propias de su función como Jefe de Control Escolar, por lo tanto, no puede dedicar su jornada de trabajo “exclusivamente” a la extracción de dichas listas que le tomaría cerca de 9 días hábiles sin realizar ninguna otra actividad.

Esto retrasa considerablemente la entrega de las listas a las coordinaciones correspondientes hasta por un lapso de un mes, ya muy cerca a la fecha designada para la aplicación de la primera evaluación parcial.

Objetivo general

Diseñar un sistema portable que permita la generación de las listas de asistencia de los alumnos mediante un proceso completamente automatizado con la finalidad de proveer al usuario final de una herramienta confiable y eficaz para desarrollar sus funciones.

Objetivos específicos

- Diseñar un Sistema que sea compatible con los programas y aplicaciones instalados en las terminales del departamento de Control Escolar.
- Ofrecer una solución eficiente en base de un compilador de software libre y multiplataforma.
- Crear el sistema con una interfaz sencilla que sea intuitiva y directa.

Marco teórico.

Para realizar efectivamente el sistema es necesario contar con una formación que permita dar solución a todas las posibles dificultades que ostente este proyecto. En mi caso obtuve dichos conocimientos durante el curso de la Licenciatura en Ingeniería en Sistemas Inteligentes.

Principalmente se requiere dominar los siguientes temas: bases de datos, bases de datos distribuidas, gestores de las bases de datos, arquitectura de las bases de datos, programación en lenguajes HTML, Mysql y Visual C++, sistemas operativos, ingeniería del software, minería de datos (ETL extracción, transformación y carga de datos)

Bases de datos

Probablemente algunas bases de datos empiezan siendo una lista en un programa de procesamiento de texto o en una hoja de cálculo. Conforme crece la lista, empiezan a aparecer repeticiones e inconsistencias en los datos, rápidamente resulta más complicado comprender los datos presentados en la lista y existen pocos métodos para buscar o recuperar subconjuntos de datos para revisarlos. Cuando empiezan a observarse estos problemas, es aconsejable transferir la información a una base de datos creada mediante un sistema de administración de bases de datos (DBMS). Con el fin de evitar la redundancia en los datos y facilitar el rápido acceso a ellos, así como un mejor control de accesos no autorizados, la posibilidad de respaldar y recuperar datos cronológicamente.

Por lo tanto podemos definir una base de datos como un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados sistemas gestores de bases de datos, abreviado DBMS, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; también son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental, sin embargo al ser una herramienta muy poderosa para recopilar y organizar información de cualquier tipo se puede utilizar en cualquier ámbito. (Microsoft, support.office, 2014)

Por su variabilidad en los tipos de datos que puede contener una base de datos se dividen en 2 tipos:

Bases de datos estáticas

Son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

Bases de datos dinámicas

Son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y edición de datos, además de las operaciones fundamentales de consulta. Un ejemplo, puede ser la base de datos utilizada en un sistema de información de un supermercado.

Basado en la percepción del mundo real, consta de una colección de objetos llamados *entidades* y de *relaciones* entre esos objetos.

Una *entidad* se describe mediante un conjunto de “atributos” Suele utilizarse el atributo id para identificar un objeto de cualquier otro. Mientras que la *relación* es una asociación entre diferentes entidades, las relaciones utilizan una correspondencia de cardinalidad entre estas relaciones y suele utilizarse la siguiente numeración:

- "0" si cada instancia de la entidad no está obligada a participar en la relación.
- "1" si toda instancia de la entidad está obligada a participar en la relación y, además, solamente participa una vez.
- "N" , "M", ó "*" si cada instancia de la entidad no está obligada a participar en la relación y puede hacerlo cualquier número de veces.

Para el garantizar el correcto funcionamiento de una base de datos es necesario establecer claramente cuales son cada uno de sus componentes (entidades) y delimitar sus relaciones. La estructura lógica de una base de datos se puede expresar gráficamente mediante un diagrama E-R, que consta de las siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan los atributos.
- Rombos, que representan las relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

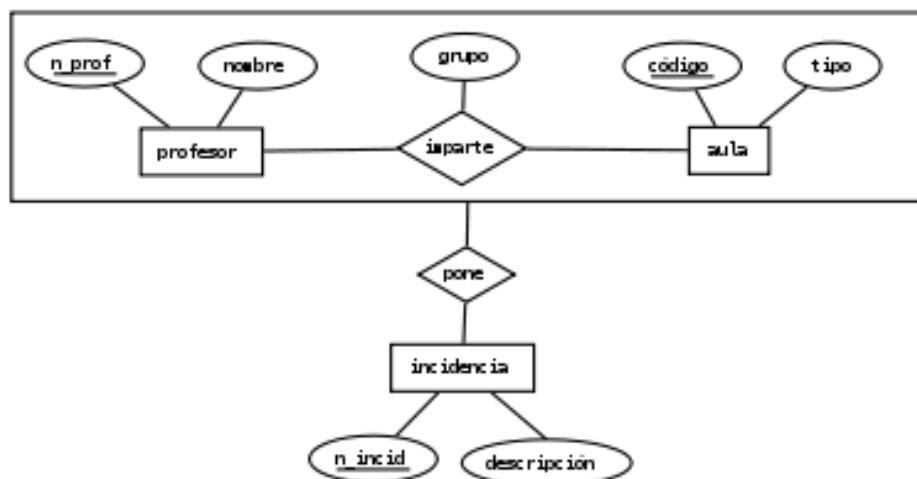


Figura 2 Ejemplo de Diagrama de modelo relacional.

Las relaciones también pueden tener atributos asociados que se representan igual que los atributos de las entidades. Un ejemplo típico son las relaciones de tipo "histórico" donde debe constar una fecha o una hora. Por ejemplo, supongamos que es necesario hacer constar la fecha de emisión de una factura a un cliente, y que es posible emitir duplicados de la factura (con distinta fecha). En tal caso, el atributo "Fecha de emisión" de la factura debería colocarse en la relación "se emite".

Las bases de datos nos permiten realizar las siguientes acciones:

- Agregar datos, por ejemplo, un elemento nuevo en un inventario.
- Modificar datos existentes en la base, por ejemplo, cambiar la ubicación de un elemento.
- Eliminar información, por ejemplo, si se ha vendido o retirado un artículo.
- Organizar y ver los datos de distintas formas.
- Compartir los datos con otros usuarios mediante informes, mensajes de correo electrónico, una intranet o Internet.

A continuación se hace una breve descripción de los componentes de una base de datos:

Tablas.

Para lograr la máxima flexibilidad en una base de datos, la información tiene que estar organizada en tablas, para que no haya redundancias. Por ejemplo, si se almacena información sobre alumnos de la UAEM, cada alumno se insertará una sola vez en una tabla que se configurará para contener únicamente datos de los alumnos. Los datos sobre unidades de aprendizaje se almacenarán en su propia tabla, y los datos sobre unidades académicas también tendrán su tabla aparte. Este proceso se conoce como *normalización*.

Cada fila de una tabla se denomina registro. En los registros es donde se almacena cada información individual. Cada registro consta de campos (o al menos uno). Los campos corresponden a las columnas de la tabla. Por ejemplo, se puede trabajar con una tabla denominada "Alumnos", en la que cada registro (fila) contiene información sobre un alumno distinto y cada campo (columna) contiene un tipo de información diferente, como el nombre, los apellidos, la dirección, o similares. Los campos se deben configurar con un determinado tipo de datos, ya sea texto, fecha, hora, numérico, o cualquier otro tipo.

Otra forma de describir registros y campos es imaginando un catálogo de fichas tradicional de una biblioteca. Cada ficha del armario corresponde a un registro de la base de datos. Cada información contenida en una ficha (autor, título, etc.) corresponde a un campo de la base de datos.

Formularios.

Los formularios se conocen a veces como "pantallas de entrada de datos". Son las interfaces que se utilizan para trabajar con los datos y, a menudo, contienen botones de comando que ejecutan diversas acciones. Se puede crear una base de datos sin usar formularios, editando los datos de las hojas de las tablas. No obstante, casi todos los usuarios de bases de datos prefieren usar formularios para ver, escribir y editar datos en las tablas.

Los formularios proporcionan un formato fácil de utilizar para trabajar con los datos. Además, se les puede agregar elementos funcionales, como botones de comando. Se pueden programar los botones para determinar qué datos aparecen en el formulario, abrir otros formularios o informes, o realizar otras tareas diversas. Por ejemplo, crear un formulario denominado "Formulario de alumno" para trabajar con datos de alumnos. El formulario de alumno podría tener un botón para abrir un formulario de inscripción en el que se pudiese escribir una nueva inscripción del alumno.

Los formularios también permiten controlar la manera en que otros usuarios interactúan con los datos de la base de datos. Por ejemplo, puede crear un formulario que muestre únicamente ciertos campos y que permita la ejecución de determinadas operaciones solamente. Así, se favorece la protección de los datos y se facilita la entrada correcta de datos.

Informes.

Los informes sirven para resumir y presentar los datos de las tablas. Normalmente, un informe responde a una pregunta específica, como "¿Cuántos alumnos se han incorporado por carrera este año?" o "¿Cuántos egresados hay en cada plan de estudio?" Y comparar ambos resultados. Cada informe se puede diseñar para presentar la información de la mejor manera posible.

Un informe se puede ejecutar en cualquier momento y siempre reflejará los datos actualizados de la base de datos. Los informes suelen tener un formato que permita imprimirlos, pero también se pueden consultar en la pantalla, exportar a un documento de texto o enviar por correo electrónico.

Consultas.

Las consultas son las que verdaderamente hacen el trabajo en una base de datos. Su función es recuperar datos específicos de las tablas. Los datos que desea ver suelen estar distribuidos por varias tablas y, gracias a las consultas, puede verlos en una sola hoja de datos. Además, puesto que normalmente no se desea ver todos los registros a la vez, las consultas permiten agregar criterios para "filtrar" los datos hasta obtener solo los registros que desee. Las consultas a menudo sirven de origen de registros para formularios e informes.

Hay dos tipos básicos de consultas: las de *selección* y las de *acción*. Una consulta de selección simplemente recupera los datos y hace que estén disponibles para su uso. Los resultados de la consulta pueden verse en la pantalla, imprimirse o copiarse al portapapeles. O se pueden utilizar como origen de registros para un formulario o un informe.

Una consulta de acción, como su nombre indica, realiza una tarea con los datos. Las consultas de acción pueden servir para crear tablas nuevas, agregar datos a tablas existentes, actualizar datos o eliminar datos.

Módulos

Los módulos, son objetos que sirven para aumentar la funcionalidad de la base de datos, se escriben en el lenguaje de programación de Visual Basic para Aplicaciones (VBA). Un módulo es una colección de declaraciones, instrucciones y procedimientos que se almacenan conjuntamente como una unidad. Un módulo puede ser de clase o estándar. Los módulos de clase se adjuntan a formularios o informes, y normalmente contienen procedimientos específicos del formulario o el informe al que se adjuntan. Los módulos estándar contienen procedimientos generales que no están asociados a ningún otro objeto. Los módulos estándar se enumeran en Módulos en el panel de exploración, pero los módulos de clase no. (Microsoft A. , 2012)

Bases de Datos Distribuidas

Originalmente se almacenaba la información de manera centralizada, pero con el paso del tiempo las necesidades aumentaron y esto produjo ciertos inconvenientes que no era posible solucionarlos o volverlos eficientes de la forma centralizada. Estos problemas impulsaron la creación de almacenamiento distribuido, los cuales hoy en día proveen características indispensables en el manejo de información; es decir, la combinación de las redes de comunicación y las bases de datos.

Hay varios factores que han hecho que las bases de datos evolucionen a bases de datos distribuidas. En el ámbito de los negocios se ha dado una creciente globalización y a la vez las operaciones de las empresas son cada vez más descentralizadas geográficamente. También el poder de las computadoras personales aumentó y el costo de los Mainframes ya no tenía sentido. Además la necesidad de compartir datos ha hecho que crezca el mercado de las bases de datos distribuidas.

Hardware

El hardware utilizado no difiere mucho del hardware utilizado en un servidor normal. Al principio se creía que si los componentes de una base de datos eran especializados serían más eficientes y rápidos, pero se comprobó que el descentralizar todo y adoptar un enfoque "nada compartido" (shared-nothing) resultaba más barato y eficaz. Por lo que el hardware que compone una base de datos distribuida se reduce a servidores y la red.

Software

El sistema manipulador de base de datos distribuida (*DDBMS*) está formado por las transacciones y los administradores de la base de datos distribuidos. Un DDBMS implica un conjunto de programas que operan en diversas computadoras, estos programas pueden ser subsistemas de un único DDBMS de un fabricante o podría consistir de una colección de programas de diferentes fuentes.

Administrador de transacciones distribuidas (DTM)

Este es un programa que recibe las solicitudes de procesamiento de los programas de consulta o transacciones y las traduce en acciones para los administradores de la base de datos. Los DTM se encargan de coordinar y controlar estas acciones. Este DTM puede ser propietario o libre.

Sistema manipulador de base de datos (DBMS)

Es un programa que procesa cierta porción de la base de datos distribuida. Se encarga de recuperar y actualizar datos del usuario y generales de acuerdo con los comandos recibidos de los DTM.

Nodo

Un nodo es una computadora que ejecuta un DTM o un DBM o ambos. Un nodo de transacción ejecuta un DTM y un nodo de base de datos ejecuta un DBM.

Planificador distribuido

El planificador está encargado de ordenar un conjunto de transacciones u operaciones que se deseen realizar sobre una base de datos. Cualquier orden en el que se decidan hacer este conjunto de operaciones se denomina planificación. Parte del trabajo del planificador es realizar estas operaciones de forma que sean serializables y recuperables.

Dos planificadores son serializables (o equivalentes) si

- Cada operación de lectura lee valores de los datos que son producidos por la misma operación de escritura en ambas planificaciones (es decir son iguales).
- La operación final de escritura en cada elemento de la data es la misma en ambas planificaciones.

Detección de bloqueos y concurrencia

Bloqueos

Un bloqueo en general es cuando una acción que debe ser realizada está esperando a un evento. Para manejar los bloqueos hay distintos acercamientos: prevención, detección, y recuperación. También es necesario considerar factores como que hay sistemas en los que permitir un bloqueo es inaceptable y catastrófico, y sistemas en los que la detección del bloqueo es demasiado costosa.

En el caso específico de las bases de datos distribuidas usar bloqueo de recursos, peticiones para probar, establecer o liberar bloqueos requiere mensajes entre los manejadores de transacciones y el calendarizador.

Para esto existen dos formas básicas:

- Autónoma: cada nodo es responsable por sus propios bloqueos de recursos.
 - Una transacción sobre un elemento con n replicas requiere $5n$ mensajes
 - Petición del recurso
 - Aprobación de la petición
 - Mensaje de la transacción
 - Reconocimientos de transacción exitosa
 - Peticiones de liberación de recursos
- Copia Primaria: un nodo primario es responsable para todos los bloqueos de recursos
 - Una transacción sobre un elemento con n copias requiere $2n+3$ mensajes
 - Una petición del recurso
 - Una aprobación de la petición
 - n mensajes de la transacción
 - n reconocimientos de transacción exitosa
 - Una petición de liberación de recurso

Podemos definir que dos operaciones entran en conflicto que debe ser resuelto si ambas acceden a los mismos datos, y una de ellas es de escritura y si fueron realizadas por transacciones distintas.

Concurrencia

El ejemplo más común de un bloqueo mutuo es cuando un recurso A está siendo utilizado por una transacción A que a su vez solicita un recurso B que está siendo utilizado por una transacción B que solicita el recurso A.

Control de concurrencia

- El problema de las actualizaciones perdidas: cuando dos transacciones concurrentes borran el efecto una de la otra
- Recuperaciones inconsistentes: acceder a información modificada parcialmente por una transacción de la.

Soluciones

El control de concurrencia y detección y manejo de bloqueos es un área de mucho estudio en las bases de datos distribuidas, a pesar de esto no hay ningún algoritmo aceptado para solucionar el problema. Esto se debe a varios factores de los cuales se consideran a los siguientes tres los más determinantes:

1. El dato puede estar duplicada en un BDD, por tanto, el manejador de la BDD es responsable de localizar y actualizar la data duplicada.
2. Si un nodo falla o la comunicación con un nodo falla mientras se realiza una actualización, el manejador debe asegurarse de que los efectos se reflejen una vez el nodo se recupere del fallo.
3. La sincronización de transacciones en sitios o nodos múltiples es difícil ya que los nodos no pueden obtener información inmediata de las acciones realizadas en otros nodos concurrentemente.

Para el control de bloqueos mutuos no se ha desarrollado ninguna solución viable y la forma más simple y que la mayoría de productos utilizan es la implementación de un tiempo máximo de espera en las peticiones de bloqueos.

Causa de estas dificultades más de 20 algoritmos de control de concurrencia se han propuesto en el pasado, y aun así siguen apareciendo nuevos.

Definición de transacciones

Una transacción es una secuencia de una o más operaciones agrupadas como una unidad. El inicio y el final de la transacción definen los puntos de consistencia de la base de datos. Si una acción de la transacción no se puede ejecutar, entonces ninguna acción dentro de la secuencia que conforma la transacción tendrá efecto.

Propiedades de las transacciones

- *Atomicidad:* Una transacción es una unidad atómica de procesamiento, esta se realiza o no se realiza.
- *Consistencia:* Si se ejecuta una transacción sobre un estado consistente, el resultado será un nuevo estado consistente.
- *Aislamiento:* Una transacción no hará visibles sus modificaciones a otras transacciones hasta que termine de ejecutarse completamente. Es decir, una transacción desconoce si otras transacciones se estén ejecutando en el sistema.
- *Durabilidad:* Una vez una transacción se ejecuta exitosamente y realiza cambios sobre el sistema, estos cambios nunca se deben perder a causa de fallas en el sistema.

Tipos de transacciones

Una transacción puede clasificarse de diferentes maneras dependiendo básicamente de tres criterios:

1. *Áreas de aplicación.* En primer lugar, las transacciones se pueden ejecutar en aplicaciones no distribuidas. Las transacciones que operan en datos distribuidos se les conoce como transacciones distribuidas. Por otro lado, dado que los resultados de una transacción que realiza un commit son durables, la única forma de deshacer los efectos de una transacción con commit es mediante otra transacción. A este

tipo de transacciones se les conoce como transacciones compensatorias. Finalmente, en ambientes heterogéneos se presentan transacciones heterogéneas sobre los datos.

2. *Tiempo de duración.* Tomando en cuenta el tiempo que transcurre desde que se inicia una transacción hasta que se realiza un commit o se aborta, las transacciones pueden ser de tipo batch o en línea. Estas se pueden diferenciar también como transacciones de corta y larga vida. Las transacciones en línea se caracterizan por tiempos de respuesta muy cortos y por acceder una porción relativamente pequeña de la base de datos. Por otro lado, las transacciones de tipo batch toman tiempos relativamente largos y accedan grandes porciones de la base de datos.
3. *Estructura.* Considerando la estructura que puede tener una transacción se examinan dos aspectos: si una transacción puede contener a su vez sub-transacciones o el orden de las acciones de lectura y escritura dentro de una transacción.

Función del manejador

El manejador de transacciones es el encargado de definir la estructura de las transacciones, mantener la consistencia en la base de datos cuando se ejecuta una transacción o se cancela la ejecución de una, mantener protocolos de fiabilidad, implementar algoritmos para el control de la concurrencia y sincronizar las transacciones que se ejecutan simultáneamente. El manejador recibe solicitudes de procesamiento de transacciones y las traduce en acciones para el calendarizador.

La operación COMMIT señala el término exitoso de la transacción: le dice al manejador de transacciones que se ha finalizado con éxito una unidad lógica de trabajo, que la base de datos esta (o debería estar) de nuevo en un estado consistente, y que se pueden hacer permanentes todas las modificaciones efectuadas por esa unidad de trabajo.

La operación ROLLBACK, en cambio, señala el término no exitoso de la transacción: le dice al manejador de transacciones que algo salió mal, que la base de datos podría estar en un estado inconsistente y que todas las modificaciones efectuadas hasta el momento por la unidad lógica de trabajo deben retroceder o anularse.

Distribución de los datos

Una de las decisiones más importantes que el diseñador de bases de datos distribuidas debe tomar es el posicionamiento de la data en el sistema y el esquema bajo el cual lo desea hacer. Para esto existen cuatro alternativas principales: centralizada, replicada, fragmentada, e híbrida.

Centralizada

Es muy similar al modelo de Cliente/Servidor en el sentido que la BDD está centralizada en un lugar y los usuarios están distribuidos. Este modelo solo brinda la ventaja de tener el procesamiento distribuido ya que en sentido de disponibilidad y fiabilidad de los datos no se gana nada.

Replicadas

El esquema de BDD de replicación consiste en que cada nodo debe tener su copia completa de la base de datos. Es fácil ver que este esquema tiene un alto costo en el almacenamiento de la información. Debido a que la actualización de los datos debe ser realizada en todas las copias, también tiene un alto costo de escritura, pero todo esto vale la pena si tenemos un sistema en el que se va a escribir pocas veces y leer muchas, y dónde la disponibilidad y fiabilidad de los datos sea de máxima importancia.

Particionadas

Este modelo consiste en que solo hay una copia de cada elemento, pero la información está distribuida a través de los nodos. En cada nodo se aloja uno o más fragmentos disjuntos de la base de datos. Como los fragmentos no se replican esto disminuye el costo de almacenamiento, pero también sacrifica la disponibilidad y fiabilidad de los datos. Algo que se debe tomar en cuenta cuando se desea implementar este modelo es la granularidad de la fragmentación.

La fragmentación se puede realizar también de tres formas:

- Horizontal: Los fragmentos son subconjuntos de una tabla (análogo a un restringir)
- Vertical: Los fragmentos son subconjuntos de los atributos con sus valores (análogo a un proyectar)
- Mixto: Se almacenan fragmentos producto de restringir y proyectar una tabla.

Una ventaja significativa de este esquema es que las consultas (SQL) también se fragmentan por lo que su procesamiento es en paralelo y más eficiente, pero también se sacrifica con casos especiales como usar JUNTAR o PRODUCTO, en general casos que involucren varios fragmentos de la BDD.

Para que una fragmentación sea correcta esta debe cumplir con las siguientes reglas:

- Debe ser *Completa*: Si una relación R se fragmenta en R_1, R_2, \dots, R_n , cada elemento de la data de R debe estar en algún R_i .
- Debe ser *Reconstruible*: Debe ser posible definir una operación relacional que a partir de los fragmentos obtenga la relación.
- Los fragmentos deben ser *Disjuntos*: Si la fragmentación es horizontal entonces si un elemento e está en R_i este elemento no puede estar en ningún R_k (para k distinto a i).

En el caso de fragmentación vertical es necesario que se repitan las llaves primarias y esta condición solo se debe cumplir para el conjunto de atributos que no son llave primaria.

Híbrida

Este esquema simplemente representa la combinación del esquema de partición y replicación. Se particiona la relación y a la vez los fragmentos están selectivamente replicados a través del sistema de BDD.

Criterios para escoger la distribución

- *Localidad de la data*: la data debería ser colocada donde ésta se accede más seguido. El diseñador debe analizar las aplicaciones y determinar cómo colocar la data de tal forma que se optimicen los accesos a la data locales.
- *Fiabilidad de la data*: Almacenando varias copias de la data en lugares geográficamente apartados se logra maximizar la probabilidad de que la data va a ser recuperable en caso de que ocurra daño físico en cualquier sitio.

- *Disponibilidad de la data:* como en la fiabilidad, almacenar varias copias asegura que los usuarios tengan a su disponibilidad los elementos de la data, aún si el nodo al que usualmente acceden no está disponible o falla.
- *Capacidades y costos de almacenamiento:* a pesar de que los costos de almacenamiento no son tan grandes como los de transmisión, los nodos pueden tener diferentes capacidades de almacenamiento y procesamiento. Esto se debe analizar cuidadosamente para determinar dónde poner la data. El costo de almacenamiento se disminuye significativamente minimizando la cantidad de copias de la data.
- *Distribución de la carga de procesamiento:* una de las razones por la cual se escoge un sistema de BDD es porque se desea poder distribuir la carga de procesamiento para hacer este más eficiente.
- *Costo de comunicación:* el diseñador debe considerar también el costo de usar las comunicaciones de la red para obtener data. Los costos de comunicación se minimizan cuando cada sitio tiene su propia copia de la data, por otro lado cuando la data es actualizada se debe actualizar en todos los nodos.
- *Uso del sistema:* debe tomarse en consideración cual será el tipo principal de uso del sistema de BDD. Factores como la importancia en la disponibilidad de la data, la velocidad de escritura y la capacidad de recuperación de daños físicos deben tomarse en cuenta para escoger el esquema correcto.

Seguridad

Desde hace ya varios años las bases de datos son ampliamente utilizadas en departamentos de gobiernos, empresas comerciales, bancos, hospitales, etc. Actualmente se está cambiando el esquema bajo el cual se utilizan las bases de datos, ya no son utilizadas únicamente de forma interna, sino que se tienen muchos accesos externos de tipos distintos. Estos cambios que se han introducido en el uso de las bases de datos ha creado la necesidad mejorar las prácticas de seguridad ya que el ambiente ya no es tan controlado como el esquema antiguo.

Los problemas de mayor importancia en seguridad son autenticación, identificación, y refuerzo de los controles de acceso apropiados. El sistema de seguridad de niveles múltiples. Éste consiste en muchos usuarios con distintos niveles de permisos para una misma base de datos con información de distintos niveles. En las bases de datos distribuidas

se han investigado dos acercamientos a este modelo: data distribuida y control centralizado, y data y control distribuidos.

En el acercamiento de data distribuida y control centralizado se divide en dos soluciones: particionado y replicado. En el primero de estos lo que se tiene es un conjunto de nodos y cada uno de ellos opera a cierto nivel de seguridad, así el usuario con nivel de permisos X accede al servidor que maneja la data para X. El replicado surgió debido a que si alguien con altos derechos de seguridad deseaba consultar data de bajo nivel de seguridad debía enviar su petición a un servidor de bajo nivel de seguridad por lo cual se podría divulgar información sensible. En el esquema replicado entonces la data se repite en cascada de tal forma que el nivel más alto tiene una copia entera de la base de datos, y el más bajo solamente la información de más bajo nivel. El otro acercamiento de data y control distribuido cada nodo contiene información de distintos niveles y está diseñado para aceptar peticiones de cualquier nivel de usuario.

El problema de inferencia

El problema de inferencia consiste en usuarios tratando de ejecutar consultas sobre la BD y estos infiriendo información sobre la respuesta legítima que la base de datos debe responder. Las herramientas para minería de datos hacen este problema aún más peligroso ya que hacen que sea más fácil para cualquier novato poder deducir patrones e información importantes de simplemente probar consultas.

Ventajas

- Refleja una estructura organizacional ya que los fragmentos de la base de datos se ubican en los departamentos a los que tienen relación.
- Autonomía local debido a que un departamento puede controlar los datos que le pertenecen.
- Disponibilidad si hubiera un fallo en una parte del sistema solo afectará a un fragmento, en lugar de a toda la base de datos.
- Rendimiento porque los datos generalmente se ubican cerca del sitio con mayor demanda, también los sistemas trabajan en paralelo, lo cual permite balancear la carga en los servidores.

- Economía, es más barato crear una red de muchas computadoras pequeñas, que tener una sola computadora muy poderosa.
- Modularidad permitiendo modificar, agregar o quitar sistemas de la base de datos distribuida sin afectar a los demás sistemas (módulos).

Desventajas

- Complejidad. Se debe asegurar que la base de datos sea transparente, se debe lidiar con varios sistemas diferentes que pueden presentar dificultades únicas. El diseño de la base de datos se tiene que trabajar tomando en cuenta su naturaleza distribuida, por lo cual no podemos pensar en hacer joins que afecten varios sistemas.
- Economía. La complejidad y la infraestructura necesaria implica que se necesitará una mayor mano de obra.
- Seguridad. Se debe trabajar en la seguridad de la infraestructura así como cada uno de los sistemas.
- Integridad. Se vuelve difícil mantener la integridad, aplicar las reglas de integridad a través de la red puede ser muy caro en términos de transmisión de datos.
- Falta de experiencia. Las bases de datos distribuidas son un campo relativamente nuevo y poco común por lo cual no existe mucho personal con experiencia o conocimientos adecuados.
- Carencia de estándares. Aún no existen herramientas o metodologías que ayuden a los usuarios a convertir un DBMS centralizado en un DBMS distribuido.
- Diseño de la base de datos se vuelve más complejo.-Además de las dificultades que generalmente se encuentran al diseñar una base de datos, el diseño de una base de datos distribuida debe considerar la fragmentación, replicación y ubicación de los fragmentos en sitios específicos.

Arquitectura de las bases de datos

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

- *Nivel interno:* Tiene un esquema interno que describe la estructura física de almacenamiento de base de datos. Emplea un modelo físico de datos y los únicos datos que existen están realmente en este nivel.
- *Nivel conceptual:* tiene esquema conceptual. Describe la estructura de toda la base de datos para una comunidad de usuarios. Oculta los detalles físicos de almacenamiento y trabaja con elementos lógicos como entidades, atributos y relaciones.
- *Nivel externo o de vistas:* tiene varios esquemas externos o vistas de usuario. Cada esquema describe la visión que tiene de la base de datos a un grupo de usuarios, ocultando el resto.

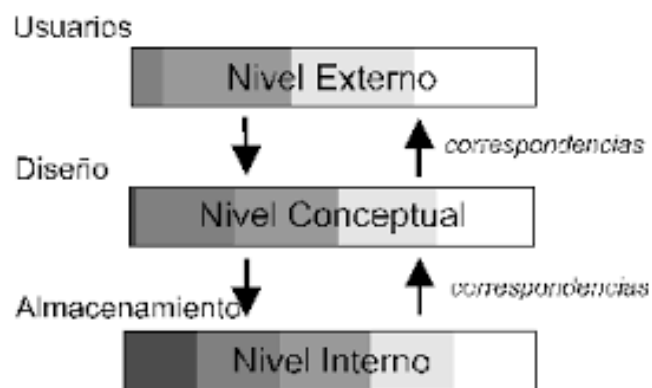


Figura 3. Niveles de la arquitectura de BDD.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física.

La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que se manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en los niveles: conceptual y externo.

Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco. En un SGBD basado en la arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el SGBD debe transformar cualquier petición expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario.

El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación. Estas correspondencias pueden requerir bastante tiempo, por lo que algunos SGBD no cuentan con vistas externas.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior.

Se pueden definir dos tipos de independencia de datos:

- La independencia *lógica* es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el

esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.

- La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

En los SGBD que tienen la arquitectura de varios niveles es necesario ampliar el catálogo o diccionario, de modo que incluya información sobre cómo establecer la correspondencia entre las peticiones de los usuarios y los datos, entre los diversos niveles (ver figura 5). El SGBD utiliza una serie de procedimientos adicionales para realizar estas correspondencias haciendo referencia a la información de correspondencia que se encuentra en el catálogo.

La independencia de datos se consigue porque al modificarse el esquema en algún nivel, el esquema del nivel inmediato superior permanece sin cambios, sólo se modifica la correspondencia entre los dos niveles. No es preciso modificar los programas de aplicación que hacen referencia al esquema del nivel superior.

Por lo tanto, la arquitectura de tres niveles puede facilitar la obtención de la verdadera independencia de datos, tanto física como lógica. Sin embargo, los dos niveles de correspondencia implican un gasto extra durante la ejecución de una consulta o de un programa, lo cual reduce la eficiencia del SGBD. Es por esto que muy pocos SGBD han implementado esta arquitectura completa.

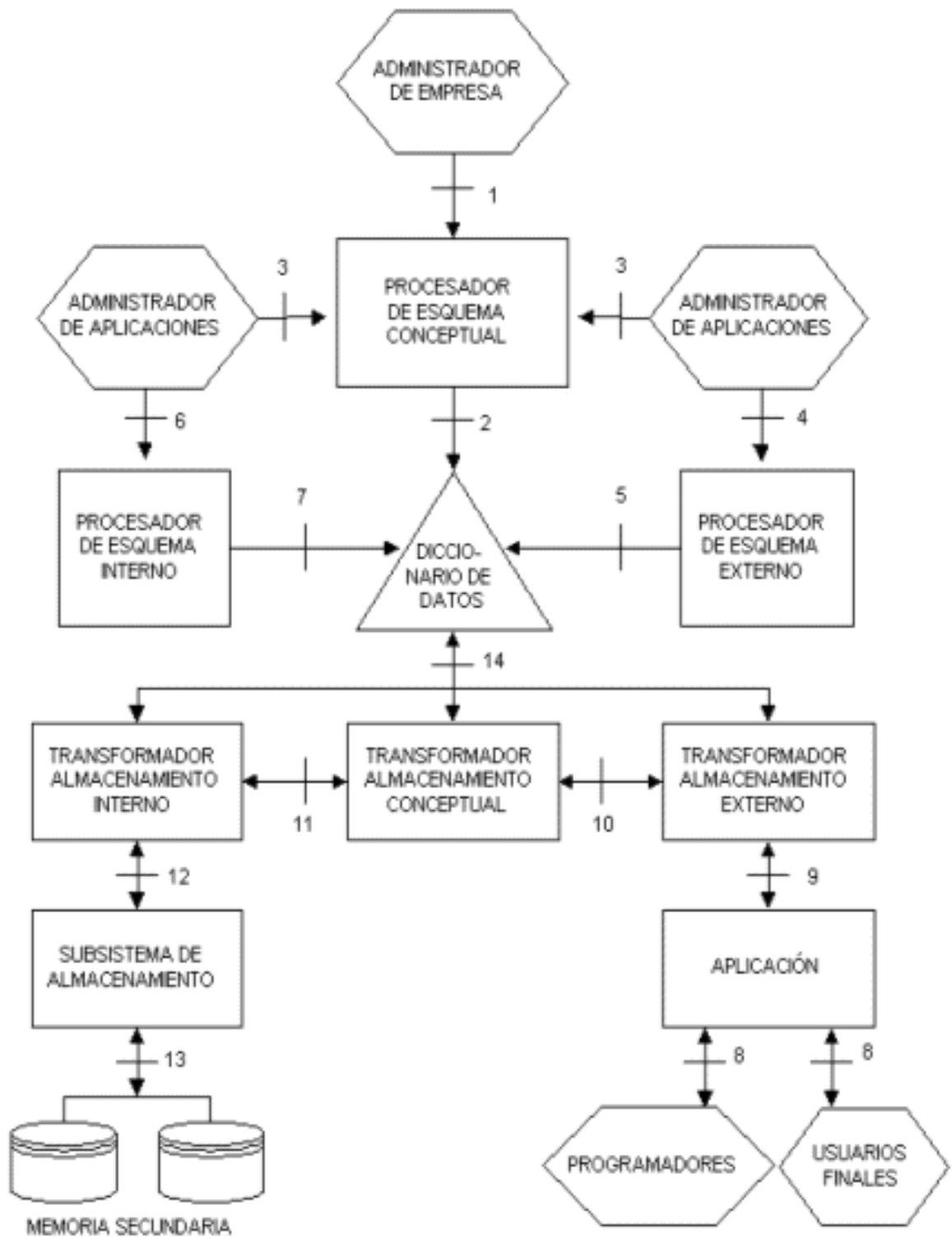


Figura 3. Ampliación del catálogo de una arquitectura de BDD

El nivel clave en esta arquitectura, es el conceptual. Éste contiene la descripción de las entidades, relaciones y propiedades de interés para la empresa (UoD), y constituye una plataforma estable desde la que proyectar los distintos esquemas externos, que describen los datos, sobre el esquema interno, que describe los datos según el sistema físico. Las posibles proyecciones de datos quedan resumidas en la gráfica.

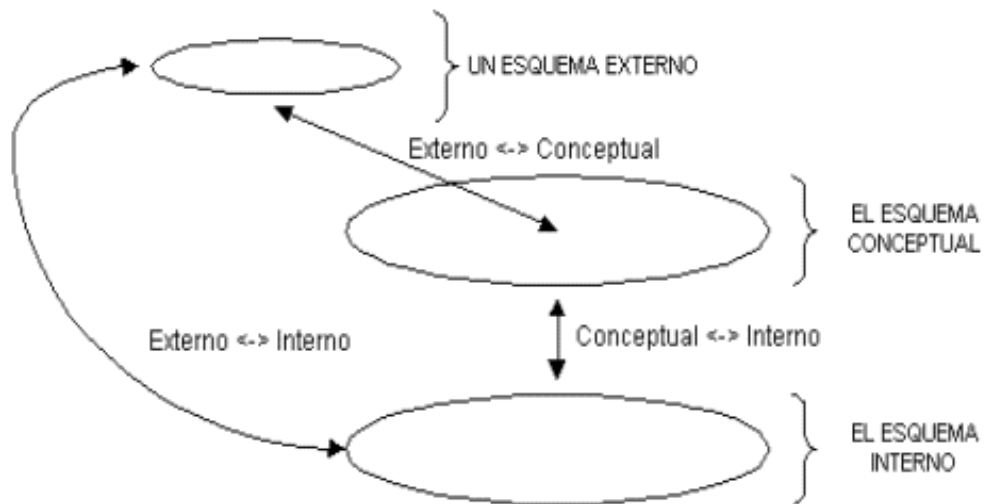


Figura 4. Tres niveles de la arquitectura de bases de datos.

Arquitectura de BDD

En un sistema de bases de datos distribuidas, existen varios factores que se deben tomar en consideración y que definen la arquitectura del sistema:

- *Distribución*: Los componentes del sistema están localizados en la misma computadora o no.
- *Heterogeneidad*: Un sistema es heterogéneo cuando existen en él componentes que se ejecutan en diversos sistemas operativos, de diferentes fuentes, etc.
- *Autonomía*: Se puede presentar en diferentes niveles, los cuales se describen a continuación:
 1. *Autonomía de diseño*: Habilidad de un componente del sistema para decidir cuestiones relacionadas a su propio diseño.
 2. *Autonomía de comunicación*: Habilidad de un componente del sistema para decidir cómo y cuándo comunicarse con otros SGBD (Sistema Gestor de Bases de Datos).
 3. *Autonomía de ejecución*: Habilidad de un componente del sistema para ejecutar operaciones locales como quiera.

La Arquitectura debe reflejar los objetivos de las BDD

Transparencia de ubicación. Permite a los usuarios tener acceso a los datos sin que tenga conocimiento de la ubicación de éstos.

Transparencia de duplicación. Para que la transparencia de duplicación sea posible, los administradores de transacciones deben traducir las solicitudes de procesamiento de transacción en acciones para el administrador de datos.

Transparencia de concurrencia. Cuando varias transacciones se ejecuten al mismo tiempo, los resultados de las transacciones no deberán afectarse.

Transparencia de fallas. Significa que a pesar de fallas las transacciones sean procesadas de un modo correcto. Frente a una falla, las transacciones deben ser atómicas, significa que se procesen todas o ninguna de ellas.

Localidad del procesamiento. Los datos se deben distribuir lo más cerca posible de las aplicaciones que los usan para maximizar la localidad del procesamiento, este principio responde a minimizar el acceso remoto a los datos.

Particionado de la Base de Datos. La base de datos se distribuye de modo que no haya duplicación de los datos mantenidos en las diferentes localidades, como no hay duplicaciones de los datos, se evitan los costos asociados con el almacenamiento y mantenimiento de datos redundantes.

Fragmentación de datos. Consiste en subdividir las relaciones y distribuirlas entre los sitios de la red, tiene como objetivo buscar formas alternativas de dividir una las instancias de relaciones en otras más pequeñas.

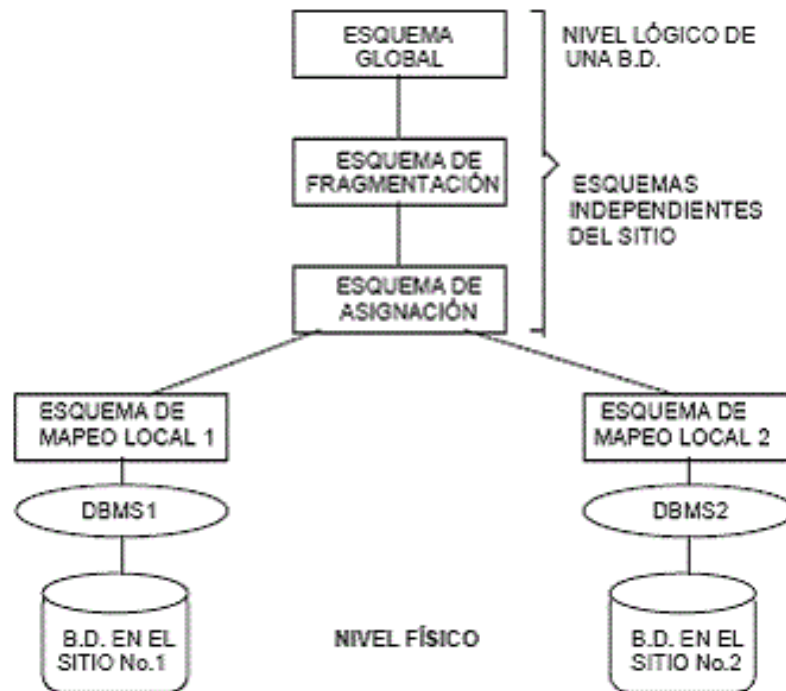


Figura 5. Esquema de arquitectura de BDD

Programación en lenguajes HTML, Mysql y Java

Los programas de ordenador se pueden clasificar según el paradigma del lenguaje de programación utilizado para producirlos. Dos de los principales paradigmas son los imperativos y los declarativos.

Los programas escritos con un *lenguaje imperativo* especifican un algoritmo utilizando declaraciones, expresiones e informes. Una declaración asocia un nombre de variable a un tipo de datos. Por ejemplo, `var x: integer;`

Una expresión produce un valor. Por ejemplo: `2 + 2` produce 4.

Por último, una declaración puede asignar una expresión a una variable o usar el valor de una variable para alterar las estructuras de control del programa.

Por ejemplo: `x := 2 + 2; if x = 4 then hacer_algo();`

Una desventaja de los lenguajes imperativos es el efecto secundario de una sentencia de asignación en una clase de variables llamadas variables no locales.

Los programas escritos en un *lenguaje declarativo* especifican las propiedades que tienen o que deben cumplirse para la salida. No especifican detalles expresados en términos de flujo de control de la máquina de ejecución pero sí de las relaciones matemáticas entre los objetos declarados y sus propiedades. Los lenguajes funcionales y lógicos son dos amplias categorías de lenguajes declarativos. El principio detrás de los lenguajes funcionales (como Haskell) es el de no permitir efectos secundarios, lo que hace que sea más fácil para razonar sobre los programas como si se tratasen de funciones matemáticas. El principio detrás de los lenguajes lógicos (como Prolog) es definir el problema a ser resuelto *la meta* y dejar la solución detallada al propio sistema Prolog. El objetivo se define proporcionando la lista de

sub-objetivos. Luego, cada sub-objetivo se define más arriba, proporcionando la lista de sus sub-objetivos, etc. Si la ruta de sub-objetivos no encuentra una solución, entonces ese sub-objetivo se retrocede y otra vía se intenta sistemáticamente.

Lenguaje HTML

HTML, siglas de *HyperText Markup Language* («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, *script*, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (intérprete del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización

automática u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

Marcado HTML

El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). El HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado *script*, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML consta de varios componentes vitales, entre ellos los *elementos* y sus *atributos*, *tipos de data* y la *declaración de tipo de documento*.

Elementos

Son la estructura básica de HTML, tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-

de-elemento>). Algunos elementos, tales como `
`, no tienen contenido ni llevan una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en HTML.



Figura 6. Estructura general de una línea de código en el lenguaje de etiquetas HTML.

El marcado *estructural* describe el propósito del texto. Por ejemplo, `<h2>Golf</h2>` establece «Golf» como un encabezamiento de segundo nivel, el cual se mostraría en un navegador de una manera similar al título «Marcado HTML» al principio de esta sección. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Puede aplicarse un formato específico al texto por medio de hojas de estilo en cascada.

El marcado *presentacional* describe la apariencia del texto, sin importar su función. Por ejemplo, `negrita` indica que los navegadores web visuales deben mostrar el texto en **negrita**, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta). En el caso de `negrita` e `<i>itálica</i>`, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: `énfasis fuerte` y `énfasis`. Es fácil ver cómo un lector de pantalla debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos presentacionales: un lector de pantalla no debería decir más fuerte el nombre de un libro,

aunque este esté en *itálicas* en una pantalla. La mayoría del marcado presentacional ha sido desechada con HTML 4.0, en favor de hojas de estilo en cascada.

Atributos

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual «=» y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas es considerado poco seguro. En contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia (tal como el atributo `ismap` para el elemento `img`).

```
1  <!DOCTYPE html PUBLIC "-//W3C/DTD HTML
2  <html>
3    <head>
4      <title>Example</title>
5      <link href="screen.css" rel="sty
6    </head>
7    <body>
8      <h1>
9        <a href="/">Header</a>
10     </h1>
11     <ul id="nav">
12       <li>
13         <a href="one/">One</a>
14       </li>
15       <li>
16         <a href="two/">Two</a>
17       </li>
```

Figura 7. Ejemplo de código HTML con coloreado de sintaxis.

`<html>`: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML. Esto es así *de facto*, ya que en teoría lo que define el tipo de documento es el DOCTYPE, que significa la palabra justo tras DOCTYPE el tag de raíz.

`<script>`: incrusta un script en una web, o llama a uno mediante `src="url del script"`. Se recomienda incluir el tipo MIME en el atributo `type`, en el caso de JavaScript `text/javascript`.

`<head>`: define la cabecera del documento HTML; esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario como, por ejemplo, el título de la ventana del navegador. Dentro de la cabecera `<head>` es posible encontrar:

`<title>`: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.

`<link>`: para vincular el sitio a hojas de estilo o iconos. Por ejemplo: `<link rel="stylesheet" href="/style.css" type="text/css">`.

`<style>`: para colocar el estilo interno de la página; ya sea usando CSS u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta `<link>`.

`<meta>`: para metadatos como la autoría o la licencia, incluso para indicar parámetros http (mediante `http-equiv=""`) cuando no se pueden modificar por no estar disponible la configuración o por dificultades con server-side scripting.

`<body>`: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo

`<body>` es posible encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:

`<h1>` a `<h6>`: encabezados o títulos del documento con diferente relevancia.

`<table>`: define una tabla.

`<tr>`: fila de una tabla.

`<td>`: celda de una tabla (debe estar dentro de una fila).

`<a>`: hipervínculo o enlace, dentro o fuera del sitio web. Debe definirse el parámetro de pasada por medio del atributo *href*. Por ejemplo: `Ejemplo` se representa como Ejemplo).

`<div>`: división de la página. Se recomienda, junto con *css*, en vez de `<table>` cuando se desea alinear contenido.

``: imagen. Requiere del atributo *src*, que indica la ruta en la que se encuentra la imagen. Por ejemplo: ``. Es conveniente, por accesibilidad, poner un atributo `alt="texto alternativo"`.

- ``: etiquetas para listas.
- ``: texto en negrita (*etiqueta desaprobadada. Se recomienda usar la etiqueta ``*).
- `<i>`: texto en cursiva (*etiqueta desaprobadada. Se recomienda usar la etiqueta ``*).
- `<s>`: texto tachado (*etiqueta desaprobadada. Se recomienda usar la etiqueta ``*).

- `<u>`: Antes texto subrayado. A partir de HTML 5 define porciones de texto diferenciadas o destacadas del resto, para indicar correcciones.

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra («/») tal como se muestra en los siguientes ejemplos:

- `<table><tr><td>Contenido de una celda</td></tr></table>`.
- `<script>Código de un script integrado en la página</script>`

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim, Notepad++, entre otros.

Existen, además, otros editores para la realización de sitios web con características WYSIWYG (*What You See Is What You Get*, o en español: «lo que ves es lo que obtienes»). Estos editores permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple, ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML, la cual va generando todo el código a medida que se va trabajando. Algunos ejemplos de editores WYSIWYG son KompoZer, Microsoft FrontPage o Adobe Dreamweaver.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo, si se edita todo en HTML y el desarrollador olvida algún código o etiqueta, basta con dirigirse al editor visual o WYSIWYG y se continúa ahí la edición o viceversa, ya que hay casos en que resulta más rápido y fácil escribir directamente el código de alguna característica que el usuario desea adherir al sitio que buscar la opción en el programa mismo.

Existe otro tipo de editores HTML llamados WYSIWYM que dan más importancia al contenido y al significado que a la apariencia visual. Entre los objetivos que tienen estos editores es la separación del contenido y la presentación, fundamental en el diseño web.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que deben aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. En general las etiquetas se aplicarán de dos formas especiales:

- Se abren y se cierran, como por ejemplo: negrita, que se vería en su navegador web como **negrita**.
- No pueden abrirse y cerrarse, como <hr />, que se vería en su navegador web como una línea horizontal.
- Otras que pueden abrirse y cerrarse, como por ejemplo <p>.

Las etiquetas básicas o mínimas son:

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Ejemplo1</title>  
  </head>  
  <body>  
    <p>ejemplo1</p>  
  </body>  
</html>
```

Lenguaje SQL

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros y no a registros individuales permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

Lenguaje de definición de datos: El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.

Lenguaje interactivo de manipulación de datos: El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.

Integridad: El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.

Definición de vistas: El LDD incluye comandos para definir las vistas.

Control de transacciones: SQL tiene comandos para especificar el comienzo y el final de una transacción.

SQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.

Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Tipos de Datos

Algunos de los tipos de datos básicos de SQL son:

Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.

Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).

Timestamp: la combinación de Date y Time.

Optimización

Como ya se dijo antes, y suele ser común en los lenguajes de acceso a bases de datos de alto nivel, el SQL es un lenguaje declarativo. O sea, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución.

El orden de ejecución interno de una sentencia puede afectar seriamente a la eficiencia del SGBD, por lo que se hace necesario que éste lleve a cabo una optimización antes de su ejecución. Muchas veces, el uso de índices acelera una instrucción de consulta, pero ralentiza la actualización de los datos. Dependiendo del uso de la aplicación, se priorizará el acceso indexado o una rápida actualización de la información. La optimización difiere sensiblemente en cada motor de base de datos y depende de muchos factores.

Lenguaje de definición de datos (DDL)

El lenguaje de definición de datos (en inglés *Data Definition Language*, o *DDL*), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE.

CREATE | CREAR

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas, vistas y procedimientos almacenados.

Ejemplo (crear una tabla)

```
CREATE TABLE 'CUSTOMERS';
```

ALTER | MODIFICAR

Este comando permite modificar la estructura de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla)

```
ALTER TABLE 'ALUMNOS' ADD EDAD INT UNSIGNED;
```

DROP | ELIMINAR

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

Ejemplo

```
DROP TABLE 'ALUMNOS';
```

TRUNCATE | BORRAR TABLA

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

Ejemplo

```
TRUNCATE TABLE 'NOMBRE_TABLA';
```

Lenguaje de manipulación de datos DML(Data Manipulation Language)

Definición

Un lenguaje de manipulación de datos (*Data Manipulation Language*, o *DML* en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

SELECT | SELECCIONAR

La sentencia **SELECT** nos permite consultar los datos almacenados en una tabla de la base de datos.

Forma básica

```
SELECT [ ALL | DISTINCT ] <nombre_campo> [ { , <nombre_campo> } ]
```

```
FROM <nombre_tabla> | <nombre_vista>
```

```
[ { , <nombre_tabla> | <nombre_vista> } ]
```

```
[ WHERE <condición> [ { AND | OR <condición> } ] ]
```

```
[ GROUP BY <nombre_campo> [ { , <nombre_campo > } ] ]
```

```
[ HAVING <condición> [ { AND | OR <condición> } ] ]
```

```
[ORDER BY <nombre_campo>|<indice_campo> [ASC | DESC]
```

```
[{,<nombre_campo>|<indice_campo> [ASC | DESC ]}]
```

Palabras reservadas, debido a que realizan una acción específica.

SELECT Palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.

ALL Indica que queremos seleccionar todos los valores. Es el valor por defecto y no suele especificarse casi nunca.

DISTINCT Indica que queremos seleccionar sólo los valores distintos.

FROM Indica la tabla (o tablas) desde la que queremos recuperar los datos. En el caso de que exista más de una tabla se denomina a la consulta "consulta combinada" o "join". En las consultas combinadas es necesario aplicar una condición de combinación a través de una cláusula **WHERE**.

WHERE Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Admite los operadores lógicos **AND** y **OR**.

GROUP BY Especifica la agrupación que se da a los datos. Se usa siempre en combinación con funciones agregadas.

HAVING Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Su funcionamiento es similar al de **WHERE** pero aplicado al conjunto de resultados devueltos por la consulta. Debe aplicarse siempre junto a **GROUP BY** y la condición debe estar referida a los campos contenidos en ella.

ORDER BY Presenta el resultado ordenado por las columnas indicadas. El orden puede expresarse con **ASC** (orden ascendente) y **DESC** (orden descendente). El valor

predeterminado es **ASC**.

Ejemplo:

Para formular una consulta a la tabla Coches y recuperar los campos matricula, marca, modelo, color, numero_kilometros, num_plazas debemos ejecutar la siguiente consulta. Los datos serán devueltos ordenados por marca y por modelo en orden ascendente, de menor a mayor. La palabra clave **FROM** indica que los datos serán recuperados de la tabla Coches.

```
SELECT matricula, marca, modelo, color, numero_kilometros,  
num_plazas  
  
FROM Coches  
  
ORDER BY marca, modelo;
```

Ejemplo de Consulta simplificada a través de un comodín de Campos (*):

El uso del asterisco indica que queremos que la consulta devuelva todos los campos que existen en la tabla y los datos serán devueltos ordenados por marca y por modelo.

```
SELECT * FROM Coches ORDER BY marca, modelo;
```

Cláusula WHERE

La cláusula *WHERE* es la instrucción que nos permite filtrar el resultado de una sentencia **SELECT**. Habitualmente no deseamos obtener toda la información existente en la tabla, sino que queremos obtener sólo la información que nos resulte útil en ese momento. La cláusula **WHERE** filtra los datos antes de ser devueltos por la consulta. Cuando en la Cláusula WHERE queremos incluir un tipo texto, debemos incluir el valor entre comillas simples.

Ejemplos:

En nuestro ejemplo, se desea consultar un coche en concreto, para esto se agregó una cláusula **WHERE**. Esta cláusula especifica una o varias condiciones que deben cumplirse para

que la sentencia **SELECT** devuelva los datos. En este caso la consulta devolverá sólo los datos del coche con matrícula para que la consulta devuelva sólo los datos del coche con matrícula MF-234-ZD o bien la matrícula FK-938-ZL. Se puede utilizar la cláusula **WHERE** solamente, o en combinación con tantas condiciones como queramos.

```
SELECT matricula, marca, modelo, color, numero_kilometros,  
num_plazas
```

```
FROM Coches
```

```
WHERE matricula = 'MF-234-ZD'
```

```
OR matricula = 'FK-938-ZL';
```

Una Condición **WHERE** puede ser negada a través del Operador Lógico **NOT**. La Siguiete consulta devolverá todos los datos de la tabla *Coches*, menos el que tenga la Matrícula MF-234-ZD.

```
SELECT matricula,marca, modelo, color, numero_kilometros,  
num_plazas
```

```
FROM coches
```

```
WHERE NOT matricula = 'MF-234-ZD';
```

La Siguiete consulta utiliza la condicional **DISTINCT**, la cual nos devolverá todos los valores distintos formados por los Campos *Marca* y *Modelo*. de la tabla *coches*.

```
SELECT DISTINCT marca, modelo FROM coches;
```

Cláusula **ORDER BY**

La cláusula *ORDER BY* es la instrucción que nos permite especificar el orden en el que serán devueltos los datos. Podemos especificar la ordenación ascendente o descendente a través de las palabras clave **ASC** y **DESC**. La ordenación depende del tipo de datos que esté definido en la columna, de forma que un campo numérico será ordenado como tal, y un alfanumérico se ordenará de la A a la Z, aunque su contenido sea numérico. El valor predeterminado es ASC si no se especifica al hacer la consulta.

Ejemplos:

```
SELECT matricula,
```

```
marca,
```

```
modelo,
```

```
color,
```

```
numero_kilometros,
```

```
num_plazas
```

```
FROM coches
```

ORDER BY marca **ASC**, modelo **DESC**; Este ejemplo, selecciona todos los campos matricula, marca, modelo, color, numero_kilometros y num_plazas de la tabla coches, ordenándolos por los campos marca y modelo, marca en forma ascendente y modelo en forma descendente.

```
SELECT matricula,
```

```
marca,
```

```
modelo,
```

```
color,
```

```
numero_kilometros, num_plazas
```

FROM coches

ORDER BY 2;

Este ejemplo, selecciona todos los campos matrícula, marca, modelo, color, numero_kilometros y num_plazas de la tabla coches, ordenándolos por el campo *marca*, ya que aparece en segundo lugar dentro de la lista de campos que componen la **SELECT**.

INSERT | INSERTAR

Una sentencia *INSERT* de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Forma básica

INSERT INTO 'tablatura' ('columna1', ['columna2,... '])

VALUES ('valor1', ['valor2,...'])

Las cantidades de columnas y valores deben ser iguales. Si una columna no se especifica, le será asignado el valor por omisión. Los valores especificados (o implícitos) por la sentencia `INSERT` deberán satisfacer todas las restricciones aplicables. Si ocurre un error de sintaxis o si alguna de las restricciones es violada, no se agrega la fila y se devuelve un error.

Ejemplo

INSERT INTO agenda_telefonica (nombre, numero)

VALUES ('Roberto Jeldrez', 4886850);

Cuando se especifican todos los valores de una tabla, se puede utilizar la sentencia acortada:

INSERT INTO nombreTabla **VALUES** ('valor1', ['valor2,...'])

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas de la tabla 'agenda_telefonica'):

INSERT INTO agenda_telefonica

VALUES ('Jhonny Aguiar', 080473968);

Formas avanzadas

Una característica de SQL (desde SQL-92) es el uso de *constructores de filas* para insertar múltiples filas a la vez, con una sola sentencia SQL:

INSERT INTO "tabla" ("columna1", ["columna2,... "])

VALUES ("valor1a", ["valor1b,..."]),

("value2a", ["value2b,..."]),...;

Esta característica es soportada por DB2, PostgreSQL (desde la versión 8.2), MySQL, y H2.

Ejemplo (asumiendo que 'nombre' y 'número' son las únicas columnas en la tabla 'agenda_telefonica'):

INSERT INTO agenda_telefonica **VALUES** ('Roberto Fernández', '4886850'), ('Alejandro Sosa', '4556550');

Que podía haber sido realizado por las sentencias

INSERT INTO agenda_telefonica **VALUES** ('Roberto Fernández', '4886850');

INSERT INTO agenda_telefonica **VALUES** ('Alejandro Sosa', '4556550');

Notar que las sentencias separadas pueden tener semántica diferente (especialmente con respecto a los triggers), y puede tener diferente rendimiento que la sentencia de inserción múltiple.

Para insertar varias filas en MS SQL puede utilizar esa construcción:


```
INSERT INTO phone_book
```

```
SELECT 'John Doe', '555-1212'
```

```
UNION ALL
```

```
SELECT 'Peter Doe', '555-2323';
```

Tenga en cuenta que no se trata de una sentencia SQL válida de acuerdo con el estándar SQL.

Para hacer lo mismo en Oracle se usa DUAL TABLE, siempre que se trate de solo una simple fila:

```
INSERT INTO phone_book
```

```
SELECT 'John Doe', '555-1212' FROM DUAL
```

```
UNION ALL
```

```
SELECT 'Peter Doe', '555-2323' FROM DUAL
```

Una implementación conforme al estándar de esta lógica se muestra el siguiente ejemplo, o como se muestra arriba (no aplica en Oracle):

```
INSERT INTO phone_book
```

```
SELECT 'John Doe', '555-1212' FROM LATERAL ( VALUES (1) ) AS t(c)
```

```
UNION ALL
```

```
SELECT 'Peter Doe', '555-2323' FROM LATERAL ( VALUES (1) ) AS t(c)
```

Copia de filas de otras tablas

Un INSERT también puede utilizarse para recuperar datos de otros, modificarla si es necesario e insertarla directamente en la tabla. Todo esto se hace en una sola sentencia SQL que no implica ningún procesamiento intermedio en la aplicación cliente. Un SUBSELECT se utiliza en lugar de la cláusula VALUES. El SUBSELECT puede contener join, llamadas a funciones, y puede incluso consultar en la misma TABLA los datos que se inserta. Lógicamente, el SELECT se evalúa antes que la operación INSERT esté iniciada. Un ejemplo se da a continuación.

```
INSERT INTO phone_book2
```

```
SELECT *
```

```
FROM phone_book
```

```
WHERE name IN ('John Doe', 'Peter Doe')
```

Una variación es necesaria cuando algunos de los datos de la tabla fuente se están insertando en la nueva tabla, pero no todo el registro. (O cuando los esquemas de las tablas no son iguales.)

```
INSERT INTO phone_book2 ( [name], [phoneNumber] )
```

```
SELECT [name], [phoneNumber]
```

```
FROM phone_book
```

```
WHERE name IN ('John Doe', 'Peter Doe')
```

El SELECT produce una tabla (temporal), y el esquema de la tabla temporal debe coincidir con el esquema de la tabla donde los datos son insertados.

```
UPDATE
```

Una sentencia *UPDATE* de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo

```
UPDATE My_table SET field1 = 'updated value asd' WHERE field2 = 'N';
```

DELETE

Una sentencia *DELETE* de SQL borra uno o más registros existentes en una tabla.

Forma básica

```
DELETE FROM tabla WHERE columna1 = 'valor1'
```

Ejemplo

```
DELETE FROM My_table WHERE field2 = 'N';
```

Recuperación de clave

```
SELECT *
```

```
FROM NEW TABLE ( INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' ) ) AS
```

Utilizando una sentencia *SELECT* después de la sentencia *INSERT* con función específica de base de datos, que devuelve la clave primaria generada por el registro insertado más recientemente.

Utilizando una combinación única de elementos del original SQL *INSERT* en una posterior sentencia *SELECT*.

Utilizando un *GUID* en la sentencia SQL *INSERT* y la recupera en una sentencia *SELECT*.

Utilizando la función de PHP `mysql_insert_id()` de MySQL después de la sentencia *INSERT*.

```
INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' )
```

```
RETURNING phone_book_id INTO v_pb_id
```

En el caso de MS SQL se puede utilizar la siguiente instrucción:

```
SET NoCount ON;
```

```
INSERT INTO phone_book VALUES ( 'Cristobal Jeldrez','0426.817.10.30' );
```

```
SELECT @@IDENTITY AS ID
```

Disparadores

Los disparadores, también conocidos como desencadenantes (*triggers* en inglés) son definidos sobre la tabla en la que opera la sentencia INSERT, y son evaluados en el contexto de la operación. Los desencadenantes BEFORE INSERT permiten la modificación de los valores que se insertarán en la tabla. Los desencadenantes AFTER INSERT no pueden modificar los datos de ahora en adelante, pero se puede utilizar para iniciar acciones en otras tablas, por ejemplo para aplicar mecanismos de auditoría.

Sistemas de gestión de base de datos

Los sistemas de gestión de base de datos con soporte SQL más utilizados son, por orden alfabético:

- DB2
- Firebird
- HSQL
- Informix
- Interbase
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Progress
- PervasiveSQL
- SQLite

- Sybase ASE

Lenguaje Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionan si no tienen Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

Otro beneficio es que se descarga Java de forma gratuita y se puede obtener la última versión en java.com. La versión más reciente de Java contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones Java que se ejecutan en cualquier equipo. La instalación de la actualización gratuita garantiza que sus aplicaciones Java sigan ejecutándose de forma segura y eficaz.

Java Runtime Environment (JRE) es lo que se obtiene al descargar el software de Java. JRE está formado por Java Virtual Machine (JVM), clases del núcleo de la plataforma Java y bibliotecas de la plataforma Java de soporte. JRE es la parte de tiempo de ejecución del software de Java, que es todo lo que necesita para ejecutarse en el explorador web.

El software del plugin de Java es un componente de Java Runtime Environment. JRE permite applets escritos en el lenguaje de programación de Java para ejecutar en varios exploradores. El software del plugin de Java no es un programa autónomo y no se puede instalar de forma independiente. Java Virtual Machine es sólo un aspecto del software de

Java que interviene en interacción web. Java Virtual Machine está incorporado en la descarga del software de Java y ayuda a ejecutar las aplicaciones Java.

Los applets de Java proporcionan funciones interactivas en un explorador web mediante Java Virtual Machine (JVM).

Comprobar/Seleccionar

Método por el que los usuarios introducen información en un cuadro de diálogo o página web, ya sea activando una casilla de control con el cursor para crear una marca de verificación o colocando el cursor sobre un botón de radio y haciendo clic para "pulsar" el botón.

Activar/Desactivar

Configuración del software accesible para el usuario que activa o desactiva ciertas funciones o capacidades. Resulta útil para resolver problemas de la configuración de Java, ya que se le puede solicitar al usuario que utilice un cuadro de diálogo para activar o desactivar una configuración o función específicas.

La tecnología Java Plugin forma parte de Java Runtime Environment y establece la conexión entre los exploradores más conocidos y la plataforma Java. Esta conexión permite que los applets de sitios Web se ejecuten en el explorador en el escritorio.

Caché de Java Runtime Environment (JRE) es el área de almacenamiento de la consola de Java que, en ocasiones, se debe borrar de forma manual para permitir la carga e instalación de la última versión de Java.

Java Virtual Machine (JVM). En Java, un conjunto de programas de software que permiten la ejecución de instrucciones y que normalmente están escritos en código byte de Java. Las máquinas virtuales de Java están disponibles para las plataformas de hardware y software de uso más frecuente.

javaw.exe

El proceso javaw.exe es un programa propiedad de Oracle, Inc., que funciona junto con el explorador Internet Explorer como un plugin de Java. Este programa es similar al programa java.exe. La única diferencia radica en que el proceso javaw.exe no tiene ninguna ventana de consola al ejecutarse. Si no desea ver una ventana de símbolo del sistema, podría utilizar mejor el proceso javaw.exe. El archivo javaw.exe es un programa de ejecución que mostrará un cuadro de diálogo durante los momentos en los que se produzca un fallo al iniciar un programa.

jucheck.exe

El proceso jucheck.exe forma parte de la instalación de Java en Windows y comprueba si hay nuevas versiones de Java. El proceso no instala Java y le notifica de que una versión más reciente esté lista para su descarga.

Los usuarios de Java pueden escoger entre tres procedimientos de instalación: instalación en línea, fuera de línea o manual.

- La instalación en línea se lleva a cabo de forma automática mientras permanece conectado a Internet haciendo clic en el botón "Descarga gratuita de Java" desde java.com
- Para la instalación fuera de línea es necesario descargar un archivo ejecutable que aparece en la lista de descarga manual de Java y que incluye todos los archivos necesarios para realizar la instalación completa a discreción del usuario. No es

necesario permanecer conectado a Internet durante la instalación. El archivo puede copiarse también e instalarse en otro equipo que no tenga conexión a Internet.

- La instalación manual descarga un archivo de programa ejecutable para instalar desde la red (IFTW) y apenas requiere intervención del usuario. Al ejecutarlo, el programa obtiene de la red todos los archivos necesarios, por lo que es imprescindible permanecer conectado a Internet durante la instalación.

Manifiesto es un archivo especial que puede contener información sobre los archivos empaquetados en un archivo JAR.

En programación, parámetro es un valor transmitido a una función. La función utiliza el parámetro en sus tareas o realiza una operación con el mismo.

La memoria física muy a menudo hace referencia a una forma de almacenamiento semiconductor en equipos conocida como memoria de acceso aleatorio (RAM).

Servidor proxy

Equipo intermediario entre el equipo del usuario e Internet. Puede utilizarse para registrar el uso de Internet y también para bloquear el acceso a un sitio web. El firewall del servidor proxy bloquea algunos sitios o páginas web por diversas razones. Como resultado, es posible que los usuarios no puedan descargar Java o ejecutar alguno de los applets de Java sin configurar el proxy en el explorador web del equipo.

Configuración de proxy

Una configuración de proxy correcta permite que los usuarios se conecten a Internet cuando se utiliza un servidor proxy. Como norma, los usuarios deben contactar con el administrador de red para obtener la información necesaria para configurar el proxy.

JRE privada

Al instalar JDK, se instala una JRE privada y, opcionalmente, una copia pública. La JRE privada se necesita para ejecutar las herramientas que incluye el JDK. No cuenta con configuración de registro y está totalmente incluida en un directorio Java, normalmente C:\Program Files\jdk1.7.0\jre, cuya ubicación sólo conoce el JDK.

JRE pública

La JRE pública la pueden utilizar otras aplicaciones Java y no va incluida en el JDK (normalmente está en C:\Program Files\Java\jre1.7.0). Se registra en Windows (enHKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft). Se puede eliminar con la opción Agregar o quitar programas y también se registra en los exploradores.

Caché del explorador web

La caché es una zona de almacenamiento temporal en la que se guardan los datos a los que se accede con frecuencia para acelerar la carga de los mismos. La caché del explorador web guarda una copia de las páginas que pasan a través del mismo. Para solucionar ciertos problemas de instalación o configuración de Java, a veces es necesario limpiar la caché de forma manual accediendo a un cuadro de diálogo.

La función Lista de excepciones de sitios permite a los usuarios finales ejecutar applets Java y aplicaciones de Java Web Start (también denominadas aplicaciones de Internet enriquecidas) que no cumplen con los requisitos de seguridad más recientes. Las aplicaciones de Internet enriquecidas incluidas en a lista de excepciones de sitios se pueden ejecutar una vez realizadas las peticiones de datos de seguridad aplicables.

Fecha de caducidad de Java

JRE caduca cada vez que hay disponible una nueva versión con correcciones a las vulnerabilidades de seguridad. Para los sistemas que no se pueden ejecutar en servidores Oracle, un mecanismo secundario se encargará de caducar esta versión de JRE. Una vez se haya cumplido cualquiera de las condiciones (la nueva versión esté disponible o se haya alcanzado la fecha de caducidad) Java enviará mensajes de advertencia y recordatorios sobre la nueva versión. Se recomienda encarecidamente a los usuarios que reciban los mensajes de la fecha de caducidad que actualicen Java a su última versión.

Línea base de seguridad

La línea base de seguridad es la mínima actualización recomendada para Java. No se recomienda a los usuarios que ejecuten versiones de Java que están por debajo de la última línea base de seguridad. Cada versión principal de Java (como Java 6 o Java 7) debe tener su propia actualización de línea base de seguridad.

Parche

Se refiere a los cambios incrementales en una instalación de software. Puede incluir correcciones para solucionar problemas generales de rendimiento y seguridad.

Alerta de seguridad

Oracle emitirá una alerta de seguridad (por ejemplo, la versión de una corrección de seguridad fuera del programa CPU normal) en casos en los que la urgencia de una solución requiere que se publique antes de la siguiente actualización de parche crítico.

Vulnerabilidades y exposiciones comunes (CVE)

Los números CVE son identificadores comunes y únicos para la información pública sobre vulnerabilidades de seguridad. El programa CVE está copatrocinado por la Oficina de ciberseguridad y comunicaciones del departamento de seguridad nacional de EE. UU. y está gestionado por MITRE Corporation.

Autoridad de certificación (CA)

Una autoridad de certificación es un tercero de confianza, por lo general, una empresa comercial que emite certificados digitales. Los certificados se emiten a las organizaciones o a las personas después de verificar su identidad. El certificado digital se agrega a las aplicaciones de la computadora con el fin de validar que la aplicación viene del propietario del certificado. Para obtener más información, consulte http://wikipedia.org/wiki/Certificate_authority.

Certificados:

- De confianza. Si la aplicación tiene un certificado de CA de confianza, se le mostrará un icono de escudo azul y deberá prestar atención a las alertas de advertencia amarillas (iconos de escudo o triángulo). Compruebe que una aplicación es de un publicador verificable comprobando que éste ha sido certificado por una CA de confianza.
- Autofirmados. No son comprobados por una autoridad de certificación de confianza. Las aplicaciones de este tipo presentan el mayor nivel de riesgo porque el publicador no está identificado.

Aplicación Sandbox

Hace referencia al nivel de acceso a los archivos del sistema, como al disco duro, y a la red. Un sandbox impide el acceso a los recursos del sistema por parte de aplicaciones que no son de confianza o que pueden no estar autorizadas.

Aplicaciones firmadas y sin firmar

La firma de código es una tecnología de seguridad por medio de la cual las aplicaciones se firman digitalmente para confirmar el autor o publicador del software. El término Aplicaciones firmadas hace referencia a las aplicaciones firmadas por el proveedor de la aplicación (autofirmadas) o por una autoridad de certificación. Esto no asegura que el propio código sea de confianza, sólo que proviene del origen indicado. Con aplicaciones sin firmar nos referimos al código que no ha sido firmado, en cuyo caso el origen de la aplicación no se puede verificar.

Sistemas Operativos

Un sistema operativo (OS Operating System) es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes (aunque puede que parte del mismo se ejecute en espacio de usuario).

Suele ser un error común muy extendido denominar al conjunto completo de herramientas sistema operativo, es decir, la inclusión en el mismo término de programas como el explorador de ficheros, el navegador web y todo tipo de herramientas que permiten la interacción con el sistema operativo. Otro ejemplo para comprender esta diferencia se encuentra en la plataforma Amiga, donde el entorno gráfico de usuario se distribuía por separado, de modo que, también podía reemplazarse por otro, como era el caso de directory Opus o incluso manejarlo arrancando con una línea de comandos y el sistema gráfico. De este modo, comenzaba a funcionar con el propio sistema operativo que llevaba incluido en una ROM, por lo que era cuestión del usuario decidir si necesitaba un entorno gráfico para manejar el sistema operativo o simplemente otra aplicación. Uno de los más prominentes ejemplos de esta diferencia, es el núcleo Linux, usado en las llamadas distribuciones Linux, ya que al estar también basadas en Unix, proporcionan un sistema de funcionamiento similar. Este error de precisión, se debe a la modernización de la informática llevada a cabo a finales de los 80, cuando la filosofía de estructura básica de funcionamiento de los grandes computadores se rediseñó a fin de llevarla a los hogares y facilitar su uso, cambiando el concepto de computador multiusuario, (muchos usuarios al mismo tiempo) por un sistema monousuario (únicamente un usuario al mismo tiempo) más sencillo de gestionar.

En ciertos textos, el sistema operativo es llamado indistintamente como núcleo o kernel, pero debe tenerse en cuenta que esta identidad entre kernel y sistema operativo es solo cierta si el núcleo es monolítico, un diseño común entre los primeros sistemas. En caso contrario, es incorrecto referirse al sistema operativo como núcleo.

Uno de los propósitos del sistema operativo que gestiona el núcleo intermediario consiste en gestionar los recursos de localización y protección de acceso del hardware, hecho que alivia a los programadores de aplicaciones de tener que tratar con estos detalles. La mayoría de aparatos electrónicos que utilizan microprocesadores para funcionar, llevan incorporado un sistema operativo (teléfonos móviles, reproductores de DVD, computadoras, radios, enrutadores, etc.). En cuyo caso, son manejados mediante una interfaz gráfica de usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador.

La forma en que se crea el programa puede ser textual o visual. En un programa de lenguaje visual, los elementos en vez de ser textualmente especificados son manipulados gráficamente.

Windows XP fue una versión de Microsoft Windows, línea de sistemas operativos desarrollado por Microsoft. Lanzado al mercado el 25 de octubre de 2001, en diciembre de 2013, tenía una cuota de mercado de 500 millones de ordenadores. Las letras "XP" provienen de la palabra eXPeriencia (eXPerience en inglés). (Windows, 2014)

Durante la década de 1990, Microsoft producía dos líneas separadas de sistemas operativos. Una línea estaba dirigida a las computadoras domésticas basada en un núcleo MS-DOS y representada por Windows 95, Windows 98 y Windows Me, mientras que la otra, basada en un núcleo "NT" y representada por Windows NT y Windows 2000, estaba pensada para el mercado corporativo y empresarial e incluía versiones especiales para servidores. Windows

XP implicó la fusión de ambas líneas en un sistema operativo único basado enteramente en la arquitectura NT y contando con la funcionalidad y compatibilidad de la línea doméstica; con él, se eliminó definitivamente el lastre de seguridad y estabilidad que involucraba llevar el código del ya obsoleto MS-DOS junto con el sistema operativo.

Características

Windows XP introdujo nuevas características:

Ambiente gráfico más agradable que el de sus predecesores.

Secuencias más rápidas de inicio y de hibernación.

Capacidad del sistema operativo de desconectar un dispositivo externo, de instalar nuevas aplicaciones y controladores sin necesidad de reiniciar el sistema.

Una nueva interfaz de uso más fácil, incluyendo herramientas para el desarrollo de temas de escritorio.

Uso de varias cuentas, lo que permite que un usuario guarde el estado actual y aplicaciones abiertos en su escritorio y permita que otro usuario abra una sesión sin perder esa información.

ClearType, diseñado para mejorar legibilidad del texto encendido en pantallas de cristal líquido (LCD) y monitores similares CRT de Pantalla Plana.

Escritorio Remoto, que permite a los usuarios abrir una sesión con una computadora que funciona con Windows XP a través de una red o Internet, teniendo acceso a sus usos, archivos, impresoras, y dispositivos.

Interfaz

Windows XP ofrece una nueva interfaz gráfica, llamada Luna. El menú Inicio y la capacidad de indexación de los directorios de Windows fueron reajustados, y otros efectos visuales fueron agregados, incluyendo:

- Colores brillantes.

- Botón "Cerrar" en forma de "Cruz" de color Rojo.
- Botones estándar de colores en las barras de herramientas de Windows e Internet Explorer.
- Un rectángulo azul translúcido en la selección de los archivos.
- Un gráfico en los iconos de la carpeta, indicando el tipo de información que se almacena.
- Sombras para las etiquetas del icono en el tablero del escritorio.
- Capacidad de agrupar aplicaciones similares en la barra de tareas.
- Capacidad para prevenir cambios accidentales.
- Resalta de color Naranja claro programas recién instalados en el menú de inicio.
- Sombras bajo los menús en (Windows 2000 solo lo tenía bajo el puntero del ratón, pero no en las ventanas o menús).

Al igual que en los anteriores Windows (Windows 98, Windows ME, Windows 2000), el Explorador de Windows incluye la vista preliminar (en miniatura) de archivos Web (*.htm, *.html) en los detalles en la barra de tareas comunes en las carpetas y en la vista en miniatura, ya sean páginas Web guardadas localmente o accesos directos a Internet.

Windows XP analiza el impacto del funcionamiento de efectos visuales y mediante esto determina si debe o no permitirlos, para evitar que la nueva funcionalidad consuma recursos en forma excesiva. Los usuarios pueden modificar más estos ajustes para requisitos particulares. Algunos efectos, tales como mezcla alfa o (transparencia), son dirigidos enteramente a muchas tarjetas de vídeo más nuevas.

El papel tapiz por defecto, es una fotografía.png de un paisaje en valle de Napa (California), con colinas verdes y un cielo azul con estratocumulos y nubes cirros. Existen varias utilidades de terceros que proporcionan centenares de diversos estilos visuales. Además, Microsoft creó el tema llamado "Energy Blue", que fue incluido con la edición Media Center 2005 de Windows XP y también fue lanzado para otras versiones de Windows XP. El tema clásico de las ventanas es extensamente popular (debido a la familiaridad con las versiones anteriores de Windows como Windows 98 y Windows ME.), no obstante las ventanas "clásicas" utilizan la misma interfaz que el otro tema estándar de Windows XP y no afectan el funcionamiento.

Programa de Ventajas de Windows Original (WGA)

El sistema Windows Genuine Advantage se instala con la actualización KB892130 y verifica si la copia de Windows es original; lo cual permite acceder a las actualizaciones de productos y seguridad de Microsoft.⁶

Windows no Original

Si la clave de producto no es Original o Genuina Windows despliega ventanas de advertencia que solicitan al usuario que adquiriera una licencia válida de Microsoft e instala un icono a lado del reloj en la barra de tareas. Además el escritorio se vuelve negro y si no se valida el sistema no se podrá descargar software o programas de Microsoft tales como Windows Media Player 11, Windows Defender, entre otros . Sin embargo, esta protección de Windows no es difícil de burlar y multitud de usuarios en todo el planeta utilizan versiones no genuinas de Windows XP sin ningún problema.

Service Packs

Cada cierto tiempo, Microsoft distribuye paquetes denominados Service Packs (Paquetes de servicio), en él se incluyen mejoras y actualizaciones a la fecha, además de algunas nuevas aplicaciones con los que aseguran un Sistema operativo (u OS) seguro. A continuación se detallan, desde su lanzamiento Microsoft que ha desarrollado:

Service Pack 1

El SP1 para Windows XP fue lanzado el 9 de noviembre de 2002. Las características que tiene son las siguientes:

La novedad más visible fue la incorporación de la utilidad Configurar acceso y programas predeterminados, para poder elegir de forma más sencilla qué programas se desea utilizar para las acciones o tareas más comunes.

Otra novedad que introdujo fue el soporte para USB 2.0 y de LBA de 48 bits, por lo que Windows XP podría soportar discos duros de más de 120 GB.

Como consecuencia de un conflicto con Sun Microsystems, Microsoft se vio forzada a sacar una revisión a este SP, llamada Service Pack 1ª (SP1a), en la que se removía la Máquina virtual Java de Microsoft.

No hay vista preliminar (en miniatura) de archivos Web (*.htm, *.html) en los detalles en la barra de tareas comunes en las carpetas ni en la vista en miniatura. Ya sean páginas Web guardadas localmente o accesos directos a Internet (url).

Al igual que el anterior sistema operativo Windows XP (sin Service Pack) y a diferencia de los posteriores (Service Pack 2 y Service Pack 3), se mantiene la barra Multimedia en Internet Explorer (versión 6.0.2600.0000), que lo integra con el Reproductor de Windows Media.

El soporte de Windows XP Service Pack 1 finalizó el 10 de octubre de 2006.

Ciclo de vida

El soporte de Windows XP RTM (sin Service Pack) finalizó el 30 de septiembre de 2004 y el Soporte de Windows XP Service Pack 1 finalizó el 10 de octubre de 2006. El soporte de Windows XP Service Pack 2 se retiró el 13 de julio de 2010, 6 años después de su disponibilidad general. La compañía terminó la distribución de licencias a los OEM y al comercio minorista de este sistema operativo el 30 de junio de 2008, 22 meses después del lanzamiento de Windows Vista. Sin embargo siguió distribuyéndose la versión "Home Edition", dirigida especialmente para portátiles ultra-baratos de forma preinstalada y como un downgrade pagado para equipos con Windows Vista preinstalado hasta el año 2010. Además el Service Pack 3 tuvo soporte hasta el 8 de abril de 2014, al igual que el soporte extendido para el Service Pack 2.

Proceso de salida del mercado.

En junio de 2008 Microsoft anunció oficialmente que ya no circularían Windows XP. Después, pasado el 30 de junio del mismo año, también afirmó que no está "terminando" con su segundo sistema operativo más popular. Aunque Windows XP no se venderá más en tiendas, Microsoft y sus socios continuaron ofreciendo soporte técnico para Windows XP durante meses y años. De hecho, Microsoft ofreció soporte técnico para Windows XP hasta el 8 de abril de 2014, como se tenía planeado. El hecho de que Windows Vista requiera semejante cantidad de memoria RAM, junto con la aparición de los ordenadores microportátiles (con 1 GB de memoria RAM) han sido decisivos en los sucesivos retrasos en la despedida de Windows XP.

El 8 de abril de 2013 el sistema operativo ocupaba el segundo lugar en uso mundial a un año de su fin de soporte, significando que de cada diez ordenadores, cuatro utilizaban el sistema. La cuota se ha reducido en su recta final de existencia pero no lo suficiente todavía; Microsoft convino a actualizar a una versión más reciente, ya que en los últimos meses el sistema se volverá más vulnerable ante ataques de piratas informáticos que están esperando que Microsoft deje sin soporte técnico el sistema, se estima que millones de sistemas ejecutándose en abril de 2014 sean atacados, a este fenómeno se le denomina "Xpocalypse". Actualmente, en febrero de 2014 el sistema todavía presenta una alta demanda del 30% mientras Windows 8 que es el sistema que Microsoft invita a cambiar no supera el 7,8%.¹⁵.

Problemas legales y críticas

Sobre XP han llovido fuertes críticas e investigaciones debido a la integración de múltiples aplicaciones para las cuales existía tradicionalmente un mercado de terceros, como cortafuegos, reproductores multimedia (Windows Media Player), programas de mensajería instantánea (Windows Messenger), así como bastante desconfianza respecto al servicio Passport de Microsoft.

Seguridad

Windows XP ha sido criticado por su susceptibilidad a malware, como virus, troyanos o gusanos. Las opciones de seguridad por defecto crean una cuenta del administrador que proporciona el acceso sin restricción a todo el sistema, incluyendo los puntos vulnerables.

Minería de datos

La depuración está estrechamente relacionada con el concepto de calidad. En un sentido amplio, depurar un programa significa librarlo de errores e inconvenientes más o menos graves, que con frecuencia es un proceso mucho más costoso y arduo de lo que pudiera parecer a primera vista, en especial en programas grandes y complejos. Sin embargo, es imprescindible si queremos ofrecer al usuario un producto con un mínimo de calidad.

Es interesante comprobar como la posición comercial de las grandes empresas de software ha evolucionado hasta una posición que podría considerarse envidiable y casi única, en relación con los productos producidos por el resto de la industria. Los que manejamos habitualmente software de cualquier tipo, desde la más humilde utilidad shareware hasta los propios sistemas operativos, estamos acostumbrados a ver durante el proceso de instalación cláusulas en las que se anuncia que el producto se vende "Tal cual" ("As is"); sin ningún tipo de garantía o de responsabilidad que pudiera derivarse de su posible mal funcionamiento. Es cierto que el software tiene unas características muy especiales y que se podría decir mucho al respecto, pero no es menos cierto que habitualmente aceptamos condiciones que no le permitiríamos al fabricante que nos vende un automóvil, un par de zapatos o una lata de conserva por ejemplo.

Podemos observar cómo incluso productos software muy conocidos y de gran consumo, producidos por compañías muy poderosas, son lanzados a veces sin un proceso de depuración adecuado; motivado la mayoría de las veces por la excesiva presión de su propia maquinaria de marketing, que las fuerza a lanzar nuevos productos y versiones a un ritmo desenfrenado.

En Junio de 2002, el NIST ("National Institute of Standards and Technology"), organismo dependiente del Gobierno USA publicaba un informe titulado "The Economic Impacts of Inadequate Infrastructure for Software Testing", en el que se analiza la repercusión económica debida a los defectos del software en dos sectores económicos del país: Servicios financieros, e industria de fabricación de medios de transporte. El resultado es que solo en estos sectores, el coste de los errores (bugs) derivados de la incorrecta depuración del software asciende a unos 60.000 millones de dólares anuales.

Depuración

En general la depuración de un programa se completa en tres fases; es un proceso en el que el producto se va acercando a la perfección (estar libre de errores e inconvenientes graves) mediante una serie de transformaciones sucesivas que responden al esquema de la figura.

En una primera fase, cuando ya el programa está prácticamente terminado, se somete a pruebas que podríamos llamar "de laboratorio" para comprobar que todo marcha según lo esperado y sin errores. Son pruebas que realiza el propio programador antes de dar a conocer el producto. Estas versiones se suelen denominar alfa y corresponden a un punto en que el programa todavía está en fase de gestación. En una versión alfa son de esperar todo tipo de errores y "cuelgues".

En una segunda fase, cuando el programador cree que su producto ya está suficientemente presentable y él no consigue encontrar más errores aparentes (o los ya conocidos están en proceso de depuración), se procede a la distribución del producto a una serie de probadores seleccionados ("beta testers"). Son las denominadas versiones beta, que aunque con errores esporádicos, pueden tener un comportamiento más o menos aceptable.

Finalmente, en una tercera fase, con la información, opiniones y sugerencias de los "beta testers", se procede a lanzar la primera versión pública v 1.0 del producto ("Release"), también denominadas versiones gamma. A partir de aquí, lo normal es que se vayan recogiendo las sugerencias, cuestiones y posibles errores que hayan pasado inadvertidos en las pruebas beta y sean reportados por los usuarios. Las soluciones a estos problemas, junto con las mejoras que se vayan incorporando, son implementadas en versiones sucesivas. Generalmente a partir de la versión 2.0 el producto se considera estable.

Puede tratarse de un simple fichero de texto plano con cualquier nombre alusivo: Cambios.txt; Historia.txt; Versiones.txt, Etc. En él se recoge la información que se estima pertinente. Por ejemplo, las fechas de publicación de las diversas versiones; las mejoras introducidas respecto a las anteriores, y sobre todo los errores ("bugs") corregidos. Este fichero suele acompañar a cada nueva versión, pero lo mejor es darles la máxima publicidad entre los usuarios potenciales y actuales. Los primeros, porque se pueden hacer una idea de la vitalidad del producto (una historia larga de actualizaciones regulares y frecuentes es buena señal). A los usuarios actuales porque les permite evaluar si les merece la pena o no el proceso de actualización a las nuevas versiones.

El depurador

Como se ha señalado anteriormente, los compiladores tienen una opción que permite incluir o no información adicional de depuración en el ejecutable. Esta información adicional consiste básicamente en la inclusión del número de línea (del código fuente) de cada sentencia, aunque también se pueden tomar otras medidas. Por ejemplo, tratar todas las funciones como si fuesen normales, no realizándose en estos casos sustituciones inline.

Nota: en el caso del compilador Borland C++ 5.5, la opción correspondiente es el comando `v`, que está conectado (ON) por defecto. El compilador GNU gcc utiliza la opción `-g` para este propósito. No olvide que la presencia de la información de depuración produce un gran

aumento de tamaño del ejecutable, por lo que solo debe utilizarse cuando verdaderamente sea necesaria.

Los entornos de desarrollo C++ actuales incluyen potentes depuradores con los que es posible controlar prácticamente todos los aspectos de ejecución de versiones alfa. Su manejo depende naturalmente de cada caso concreto, pero en general permiten inspeccionar el estado de llamadas de la pila, lo que significa controlar la "traza" de la ejecución. De esta forma es posible conocer el camino que ha seguido la ejecución hasta llegar a un punto concreto (que funciones han sido invocadas y cuál es el valor de las variables). Por ejemplo, cuantas veces se ha invocado a sí misma una función recursiva y que valores tienen sus variables en cada una de las instancias; el valor de las variables globales, locales, automáticas y estáticas.

Las opciones anteriores son las que podríamos llamar mínimas. Por supuesto las versiones más avanzadas de los productos punteros permiten hurgar más cómoda y profundamente en las entrañas del ejecutable. Por ejemplo, depurar funciones miembro y aplicaciones multihebra; controlar la ejecución paso a paso; a nivel de instrucciones máquina (ensamblador) o a nivel de sentencia de nuestro código fuente, así como instalar puntos de control ("break points") en el ejecutable. Se trata de instrucciones específicas instaladas a la entrada de las zonas de código donde sospechamos que se presentan los problemas. Esto nos permite correr el programa a velocidad normal, pero al llegar a dichos puntos, es invocado automáticamente el depurador que nos muestra todas sus herramientas; entonces se pueden realizar las comprobaciones oportunas, y si todo está correcto, volver a modo ejecución normal hasta que se alcanza el próximo punto de control (en los ejemplos que siguen aprenderemos a instalar en nuestro código puntos de control rudimentarios sin necesidad de utilizar ningún depurador).

Para cuando estas posibilidades no bastan, o sencillamente no se dispone de ellas, existen una serie de trucos y técnicas generales que facilitan la depuración. El proceso más general y socorrido consiste en incluir en el código una serie de puntos testigo ("flags") con salidas provisionales que nos informan que el programa ha pasado por el punto sin novedad y los valores de las variables sospechosas de mal funcionamiento.

Métodos y técnicas de investigación empleadas.

Para la elaboración de este sistema, se realizó un exhaustivo análisis en busca de ofrecer la mejor solución al planteamiento del problema y cumplir con los objetivos particulares y generales. A continuación se menciona la sucesión de pasos necesarios para desarrollar el sistema y garantizar un resultado satisfactorio.

1. Discriminar inicialmente de manera visual, a través del archivo generado de la consulta de las inscripciones correspondientes al período de interés, los datos que se desean extraer, así como los que no serán extraídos.
2. Analizar el orden y la distribución de estos datos.
3. Por ejemplo: se inicia con el encabezado de las listas el cual estará centrado respecto de la página, la primera línea se coloca a partir de la tercera columna y contiene el nombre de la universidad: UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO, en la segunda línea los datos de la UAP Nezahualcóyotl irán colocados a partir de la segunda columna, luego se copian los datos de la materia como son el nombre, clave, el grupo, etc. En la siguiente fila se deja un espacio y posteriormente se agrega una fila para nombrar a cada una de las columnas con el tipo de dato que va a contener esta, por ejemplo: número consecutivo, número de cuenta, nombre, etc. Finalmente la lista a partir de la primera columna.
4. Identificar los patrones que rodean la información de interés.
Por ejemplo: el tipo de información, si es numérica o no, y su respectiva extensión lo cual ayuda a limitar la información que será almacenada en cada columna.

Una vez que se ha concluido el diseño, se inicia la etapa de programación. Para este sistema se hizo uso del lenguaje multiplataforma Java, esto debido a que, como ya se estableció en

el planteamiento del problema, los requerimientos solicitados para poder acceder al portal electrónico de Control Escolar son obsoletos, sin embargo el sistema debe funcionar en cualquier terminal muy independiente del software instalado. Otro beneficio del uso del lenguaje Java es que muestra una interfaz amigable e intuitiva para el usuario, muy similar a las ventanas por defecto de Microsoft con las que el usuario ya está muy familiarizado.

El sistema está modularizado en 3 clases llamadas: depuración, exportación y sistema. Como su nombre lo indica en la clase depuración se recibe el archivo de entrada que fue descargado de la base de datos de la UAEM y se encuentra en un formato de texto plano con extensión .txt. Aquí se especifica con órdenes precisas cuáles son los tags o signos que no se copiarán al archivo de salida.

Evidentemente en la clase exportación, se envían las cadenas que si se utilizarán en el archivo de salida a un archivo con extensión .xls. Esto se hace posible gracias a los flujos de entrada y salida que guardan temporalmente la información que utilizaremos más tarde.

La clase Sistema es la que organiza y llama a las otras dos, aquí se encuentra el nombre y la forma que tiene el programa

Presentación y discusión de resultados.

Al concluir el “Sistema para la extracción automática de información en archivos de texto plano generados por bases de datos” se observan los siguientes resultados:

La figura 8, muestra la ventana que se abre al ejecutarse el Sistema. En esta primera ventana se solicita la ruta y nombre del archivo que se desea procesar.

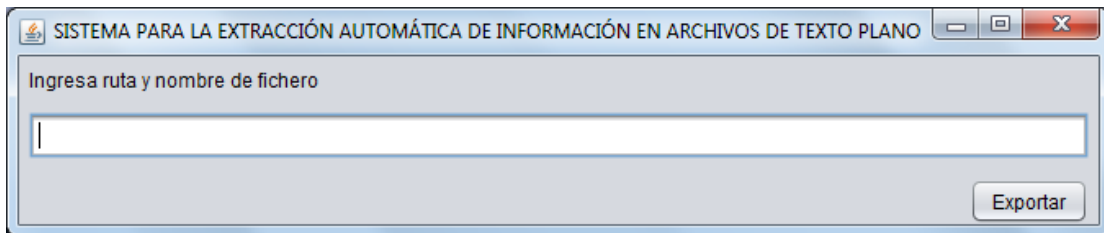


Figura 8. Se ejecuta el Sistema.

La figura 9, muestra como se debe ingresar la ubicación del archivo que se desea procesar en el sistema, este debe tener un formato .txt.

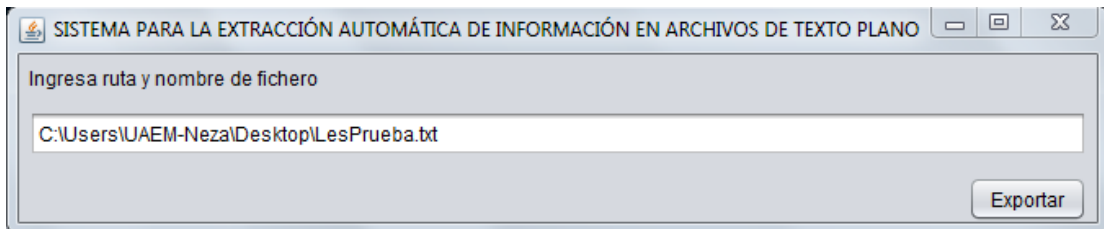


Figura 9. Ingresar ubicación del archivo.

Finalmente el archivo generado en extensión .xls, se guardará en la misma ubicación del archivo origen y con el mismo nombre, sólo que con la nueva extensión

Num	Cve Alumno	Nombre del alumno	% Asistencia	1er Parciales	2do Parciales	Promedio	Ordinario Número	Firma
1	1328621	ACOSTA AGUILAR PAULA GUADALUPE						
2	1328620	ANGEL VEGA GUSTAVO ALEJANDRO						
3	1328619	ARIAS LOPEZ MARICARMEN						
4	1328617	ARTIAGA LOPEZ BRENDA ELIZABETH						
5	1328633	CABRERA GONZALEZ MARIA VICTORIA						
6	1328630	CAMPOS FABILA SARA KAMICHI						
7	815608	CAPISTRAN CERVANTES CECILIA						
8	815604	CAPISTRAN CERVANTES MIRNA						
9	1370709	CARBAJAL ISLAS VASTI						
10	1328631	CARRANZA HERNANDEZ MIGUEL ANGEL						
11	1328626	CARRASCO GONZALEZ ESTHEFANNY LIZBETH						
12	1370708	CARRILLO DOMINGUEZ LUCIA ESTEFANI						
13	1041038	CARRILLO PEREZ HILDA ABRIL						
14	1328634	CRUZ MARTINEZ ABEL HAZEL						
15	1328638	ESCALONA MUCI O MONSERRAT ABIGAIL						
16	1227953	FALFAN VALENCIA SERGIO IVAN						
17	1328637	FLORES GARCIA LESLIE FABIOLA						
18	1229963	FLORES PEREZ VIRGINIA						
19	1328641	GARCIA LUCAS GUADALUPE ALEJANDRA						
20	1328644	GARCIA REYES JESSICA VIANEY						
21	1328643	GARCIA VELAZQUEZ NADIA CARINA						
22	1328639	GUZMAN MARTINEZ JORDAN						
23	1328653	HERNANDEZ LOPEZ GILBERTO						
24	1370710	HERNANDEZ PAEZ ANA LAURA						
25	1370705	ISLAS GOMEZ KARLA GIOVANNA						
26	1328656	JIMENEZ AYALA BRENDA VANIA						
27	1328635	JUAREZ CARRASCO ALEJANDRO						

Figura 10 Resultado final.

El Sistema cumple el objetivo para el cuál fue creado, de una forma muy sencilla para el usuario ya que al ejecutarlo únicamente se le solicita la ubicación del archivo y con un solo clic el sistema arroja el resultado esperado.

Conclusiones y sugerencias.

Al término de la propuesta: Creación, desarrollo e implementación del **“Sistema para la extracción automática de información en archivos de texto plano generados por bases de datos”**, se comprobó la eficiencia del mismo ya que el proceso para el cuál fue desarrollado dicho sistema fue optimizado con éxito. Esto se refleja en la reducción significativa de tiempo que se invirtió en la tarea de la recuperación, limpieza, conversión y edición de las listas de todos los alumnos de la Unidad Académica Profesional Nezahualcóyotl, que los docentes utilizarán durante el período escolar 2015A. Se estima que la reducción del tiempo fue del 80%, lo cual podría reducirse aún más pero todavía existe el inconveniente de la velocidad del servicio de internet. Por otro lado, recordando que es un problema para los 51 espacios académicos que conforman a la UAEM sería de suma utilidad darle continuidad a la aplicación por tal motivo se sugiere compartir o en su defecto probar el sistema con los demás espacios académicos, así como ampliar su uso para la edición de reportes de alumnos inscritos por semestre, reportes de bajas, renunciaciones, materias con mayor índice de reprobación etc., cuya información es solicitada, de igual forma al departamento de Control Escolar, por áreas como Tutoría Académica, coordinaciones de los programas educativos, subdirección académica etc., esta información es de suma importancia para llevar a cabo distintos programas como el rescate académico.

O la otra sugerencia sería actualizar los requerimientos solicitados por la interfaz oficial del sitio de Control Escolar de la UAEMex, debido a que los que se muestran actualmente resultan obsoletos y hasta discontinuados y que el código fuente de la base de datos que contiene las inscripciones de los alumnos fuera modificado de tal modo que contuviera una sentencia especial para que el usuario (empleado de departamento de Control Escolar) pueda exportar las listas directo a un archivo con extensión .xls, si bien esto no resuelve el problema en su totalidad, también ahorraría algo de tiempo. No hay que olvidar que se requiere limpiar algunos campos y por ende el programa seguiría utilizándose.

Finalmente se concluye que el sistema desarrollado genera un impacto para la reducción de tiempo de trabajo y de fácil utilización debido a que la plataforma utilizada son en un ambiente amigable para el usuario sin ser experto en programación o en bases datos.

Terminología y Glosario

Extracción de la información. Por sus siglas en inglés IE, (Information Extraction) es un tipo de recuperación de la información cuyo objetivo es extraer automáticamente información estructurada o semi estructurada desde documentos legibles por una computadora.

Extracción de terminología. Identifica y extrae candidatos a términos de los textos explorados.

Tags. Líneas verticales “|” que se utilizan en documentos de texto para delimitar palabras, oraciones, ideas, etc.

Archivo o texto plano. Es un archivo con extensión .txt (separado por tabuladores o tags) o .csv (separado por comas) los editores de este tipo no permiten darle ningún formato, sólo se puede copiar, cortar, pegar y guardar.

Archivo con diagramación. Es un archivo que puede tener diferentes extensiones y que el programa con que se abre si permite darle un formato de resaltado, color, margen, tipografía, etc. Así como manipular los datos con operaciones, tablas, imágenes etc.

Tipo de dato. Es un atributo que especifica el tipo de dato que el objeto puede contener: datos de enteros, datos de caracteres, datos de moneda, datos de fecha y hora, cadenas binarias, etc.

DBMS. Data Base Management System. Son las siglas en inglés para los Sistemas de Gestión de Bases de Datos(SGBD)

Entidad. Cosa u objeto del mundo real (abstracción).

Relación. Es una asociación entre diferentes entidades, las relaciones utilizan una correspondencia de cardinalidad.

Cardinalidad. Es la correspondencia de las relaciones entre entidades.

DDBMS. Sistema manipulador de base de datos distribuida

Applet. Es un componente de software (código de programa) que el explorador descarga para proporcionar funciones en una página web.

Cuadro de diálogo. Ventana especial de una interfaz de usuario gráfica que informa al usuario o le solicita información.

Firewall. Software o hardware que protegen una terminal de ataques externos de Internet que puedan suponer una amenaza para la seguridad para los archivos contenidos en esta ya que permite establecer reglas para permitir o denegar el pasaje del tráfico

Archivo Java (.jar) Es una formato de archivo que se utiliza para agrupar varios archivos dentro de un único archivo de almacenamiento. Contiene los archivos de clase y recursos auxiliares asociados a applets y aplicaciones.

Proceso. Término general para describir un programa que se está ejecutando para realizar una tarea específica, con bastante frecuencia junto con otros programas.

Solución alternativa. Una solución alternativa es una solución temporal para evitar un problema de sistema identificado a la espera de una solución más permanente.

VII Bibliografía

http://es.wikipedia.org/wiki/Extracci%C3%B3n_de_la_informaci%C3%B3n. (s.f.).

Microsoft. (2014). *support.office*. Recuperado el 4 de Enero de 2015, de <https://support.office.com/es-mx/article/Conceptos-b%25C3%25A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>

Microsoft. (s.f.). *Microsoft Download Explorer*. Recuperado el 2 de Enero de 2015, de <http://www.microsoft.com/es-mx/download/details.aspx?id=1>

Microsoft, A. (Junio de 2012). *Support Office*. Recuperado el Enero de 2015, de <https://support.office.com/es-mx/article/Conceptos-b%25C3%25A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>

Tinoco Gómez, O. R. (2010). Criterios de selección de metodologías de desarrollo de software Industrial Data. *Redalyc*, 70-74.

Windows. (28 de Noviembre de 2014). *Windows Microsoft downloads XP*. Obtenido de <http://windows.microsoft.com/es-xl/windows/help/learn-how-to-install-windows-xp-service-pack-3-sp3>

Anexos

Anexo I Requerimientos para acceder a Control Escolar

El portal de la UAEM para acceder al departamento de Control Escolar como usuario de tipo administrativo, es la URL: <http://controlescolar.uaemex.mx/dce/sicde/site/login.jsp>

Al acceder a ella se muestra la siguiente interfaz



En ésta página se observan los siguientes requerimientos de visualización:



A continuación realizo un análisis de los requerimientos que a su vez se solicitan para poder instalar estos otros requerimientos de visualización:

1.- Internet Explorer Versión 6.x, 7 y 8

Sistema operativo compatible:

Windows XP:

32 MB of RAM minimum

Full install size: 12.0 MB

CD-ROM drive (if installation is done from a CD-ROM)

Algunos componentes pueden requerir recursos adicionales no especificados en línea.

(Microsoft, Microsoft Download Explorer)

2.- Sistema Operativo Windows XP

Requisitos del sistema

	Mínimos	Recomendados	Óptimos
Procesador	233 MHz	300 MHz o superior	500 MHz o superior
Memoria RAM	64 MB	128 MB o +	512 MB o +
Vídeo	Super VGA	(800×600) o resolución superior	
Espacio en disco duro		1,5 GB o superior	

(se necesitan 1.8 GB más para el Service Pack 2 y otros 900 MB adicionales para el Service Pack 3)

Dispositivos ópticos	Unidad de CD-ROM o DVD-ROM
Periféricos	Teclado y mouse u otro dispositivo señalizador
Multimedia	Tarjeta de sonido, altavoces o auriculares

Es posible instalar y ejecutar el sistema operativo en procesadores IA-32 antiguos como los P5 Pentium sin instrucciones MMX.17 Windows XP no es compatible con procesadores anteriores a los Pentium (como el 486) debido a que requiere de las instrucciones CMPXCHG8B.

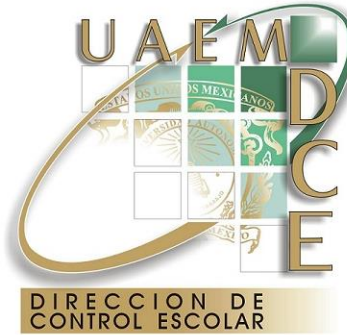
Para muchas tareas, incluyendo la navegación web, el correo electrónico y otras actividades sencillas, 64 MB de memoria RAM proporcionan una experiencia de usuario equivalente o superior a la de Windows Me en el mismo tipo de hardware. (Windows, 2014)

3.- Los Manuales y Utilería se descargan en la dirección electrónica <https://controlescolar.uaemex.mx/dce/Manuales/Default.html>



4.- Archivo Digital

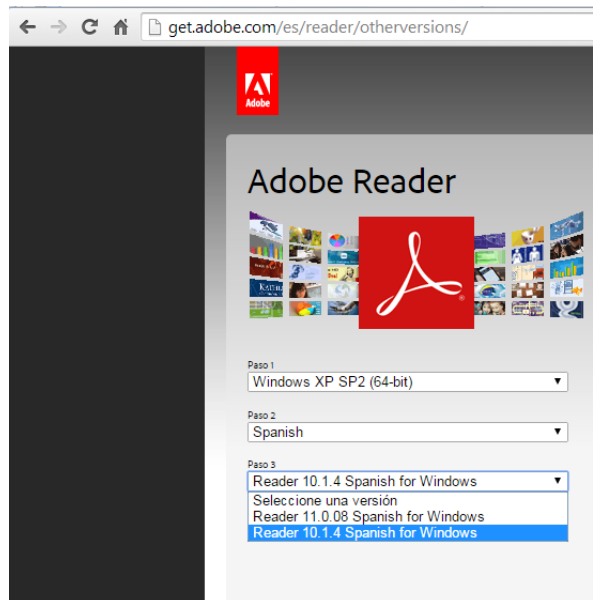
<http://148.215.124.40:8090/fortimax/>



D.R. ©2006 Sistemas y Computadores de Gestión S.A. de C.V.
Av. Revolución 2042, Col. La Otra Banda, Del. Álvaro Obregón, C.P 01090, México D.F.

5.- Adobe Reader Versión 4.x

En la página oficial <http://get.adobe.com/es/reader/otherversions/> ya no se encuentra disponible la Versión 4.x que se requiere en el portal de Control Escolar



Anexo 2 Manual de usuario e instalación

El programa recibe un archivo como el que se muestra a en la figura 11. Encerrado en elipses se hace notar la información que tiene que eliminarse para generar las listas.

Figura 11. Archivo de entrada.

UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO
DIRECCION DE CONTROL ESCOLAR
UNIDAD ACADÉMICA PROFESIONAL NEZAHUALCOYOTL
PERIODO DE CURSOS OTOÑO 2014
LISTA DE ASISTENCIA DE ALUMNOS INSCRITOS EN EL CURSO

Unidad de Aprendizaje: L40251 PRACTICA INTEGRAL COMUNITARIA II Grupo: M7
Profesor: AOV8910418 I F S APOI TNAR VELASCO BRENDA VOITZTN

Num	CveAlu	Nombre del alumno	
1	1127338	ALLENDE BALDERRAMA / JOSELINNE
2	1127339	ALONSO MINOR / FATIMA
3	1127345	BAILON CHAVEZ / NANCY ALEJANDRA
4	1127342	BECERRA VENEGAS / GABRIELA
5	1127390	CASTAYEDA GONZALEZ / ARACELI
6	1127347	CASTELAN CERON / SERGIO MICHEL
7	1127354	CEDILLO CORTES / ALEJANDRA
8	1129325	CORONA CANO / AUREA MONTSERRAT
9	1127415	DE LA VEGA PELAEZ / ANAHI
10	1129326	DEL CUETO APARICIO / LAURA FERNANDA
11	1127387	FARIAS FRANCO / JESSICA BRENDA
12	1127356	FIGUEROA ALCANTARA / DAVID
13	1127355	FIGUEROA RAMIREZ / PRISCILA ALIN
14	0841308	GALICIA MANCILLA / EDUARDO
15	1127361	GOMEZ PELCASTRE / SOFIA NAYELI
16	1127398	LARA CHAVERO / MARIELA
17	1127371	LOPEZ AGUILAR / ANA KAREN
18	0647370	LUCERO CHAVEZ / MAGALY
19	1129327	MARQUEZ MARTINEZ / MARI JOSE
20	1127376	NERI RODRIGUEZ / LIZHETT
21	1127386	PALMERIN BENITEZ / YESICA
22	1127381	POLICARPO SEGUNDO / SERGIO
23	1127397	RAMOS TRUJILLO / LESLY
24	1127399	RODRIGUEZ BELLO / NAYELI ITALDIBE
25	1127411	RUIZ HERNANDEZ / FLOR ELISA
26	0748095	RUVALCABA DE ALBA / ROCIO
27	1129331	SANCHEZ ESCANDON / VIRIDIANA RUBI
28	1127393	SANTIAGO GARCIA / FERNANDO
29	1127394	TEJEDA GARCIA / NANCY JESUS
30	1127412	TORRES MARTINEZ / JOSE ANTONIO
31	1127407	URRUTIA SILVA / VERONICA MIRIAM

Se hace notar que el archivo no tiene ningún formato diferente al de una impresión. No cumple con las columnas necesarias para los registros de los docentes que les sirven para evaluar a los alumnos en cuanto al primer y segundo parcial, el ordinario y la firma, así como las asistencias.

El sistema genera un archivo en Excel como el que se muestra a continuación:

	A	B	C	D	E	F	G	H	I
1			UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO						
2			DIRECCION DE CONTROL ESCOLAR						
3			UNIDAD ACADEMICA PROFESIONAL NEZAHUALCOYOTL						
4									
5									
6			Unidad de Aprendizaje: L40251 PRACTICA INTEGRAL COMUNITARIA II Grupo:M7						
7			Profesor: AOV910418 L.E.S. APOLINAR VELASCO BRENDA YOLTZIN						
8									
9									
10				Asistencia	Parciales			Ordinario	
11	No.	No.CTA	NOMBRE	%	1er.	2do.	Prom	Num	Firma
12	1	1127338	ALLENDE BALDERRAMA JOSELINNE						
13	2	1127339	ALONSO MINOR FATIMA						
14	3	1127345	BAILON CHAVEZ NANCY ALEJANDRA						
15	4	1127342	BECERRA VENEGAS GABRIELA						
16	5	1127390	CASTAÑEDA GONZALEZ ARACELI						
17	6	1127347	CASTELAN CERON SERGIO MICHEL						
18	7	1127354	CEDILLO CORTES ALEJANDRA						
19	8	1129325	CORONA CANO AUREA MONTSERRAT						
20	9	1127415	DE LA VEGA PELAEZ ANAHI						
21	10	1129326	DEL CUETO APARICIO LAURA FERNANDA						
22	11	1127387	FARIAS FRANCO JESSICA BRENDA						

Figura 12. Archivo procesado el por el sistema

Pasos para el correcto funcionamiento del sistema:

1. Abrir el sistema.
2. Proporcionar la ubicación del archivo que se desea procesar incluyendo el nombre y la extensión.
3. Presionar el botón para la “Exportación”, se generará un archivo con el mismo nombre y ubicación que el archivo procesado pero con extensión .xls.
4. En caso de que la ubicación no sea correcta, el sistema emitirá un mensaje con la advertencia correspondiente.

Agradecimientos.

A mi hijo Rodrigo por su ternura e inocencia que tanto amo, por ser mi primordial estímulo de superación personal y profesional.

A mis padres por su carácter firme, que son el principal cimiento en la formación de la mujer responsable que hoy soy.

A mi esposo por su apoyo económico y abandono moral en el momento preciso que comenzaba la carrera, ello me dio la determinación para perseguir mi sueño.

A mi asesora de tesis, maestra y gran amiga, Dra. Dora María Calderón por brindarme su apoyo, enseñanza y amistad, que me dieron la confianza de proseguir en cada período escolar.

Al Mtro. Efrén González por creer en mí aún en los momentos más críticos durante toda la carrera, por sus enseñanzas y apoyo incondicional que me devolvieron la fe.

A cada uno de los docentes que me compartieron sus conocimientos y experiencias coadyuvando a mi formación académica.

A mis hermanas y hermanos que me alentaron siempre a continuar.

A la UAEM que me proporcionó una oportunidad a través de la UAP Nezahualcóyotl de superarme y por apoyarme con una beca institucional a lo largo de casi toda la carrera.

A mis compañeros y compañeras que me compartieron su tiempo.

A Dios por darme la vida y los medios para trascender en ella.