

/ADSI:

ANÁLISIS Y DESARROLLO
DE SISTEMAS DE INFORMACIÓN

FASE IDENTIFICACIÓN

Las características de ASP son:

- El código ASP se puede mezclar con el código HTML en la misma página (no es necesario compilarlo por separado).
- El código ASP se puede escribir con un simple editor de textos como el Bloc de notas de Windows o UltraEdit.
- Como el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
- Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tanto que proporcione un driver para el acceso.
- ASP permite usar componentes escritos en otros lenguajes (C#, C++, Visual Basic), que se pueden llamar desde el código ASP.

MOTORES BASES DE DATOS

HISTORIA - SISTEMA GESTOR BASE DATOS (SGBD)
COMPARACIÓN SGBD EN EL MERCADO

ACTIVIDAD DE PROYECTO

1. Determinar las especificaciones funcionales del Sistema de Información.

ACTIVIDAD DE APRENDIZAJE

1. Construir el modelo tecnológico de información empresarial de acuerdo con las tendencias nacionales e internacionales del sector desarrollo de software.

ramar:
si.dlsi.ua.es/

wikipedia.org/wiki/ASP.NET#Características

rw.clikear.com/manuales/csharp/c10.aspx

wikipedia.org/wiki/Microsoft_SQL_Server

BASE DE DATOS

Microsoft SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Características de Microsoft SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.

De clase mundial

- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.



CONTENIDO

COMPARACION SGBD EN EL MERCADO 8

Tabla comparativa

HISTORIA 4 SISTEMA GESTOR BASE DE DATOS (SGBD) 6

Concepto

Tipos

Integridad

Operaciones

Procesamiento

Definición

Características

Glosario

Referencias

10



Creative
Commos



Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

HISTORIA

Antes de utilizar las bases de datos conocidas hoy en día, existían los ficheros secuenciales como almacenes de datos. Los cuales permitían un acceso rápido a la información, pero solamente de forma secuencial, mas adelante aparecieron los ficheros indexados, donde el acceso ya podía ser aleatorio, el sistema de ficheros era el sistema más común de almacenamiento de datos. Para compartir los datos entre varias máquinas surgió el NFS (*Network File System*), y más tarde para evitar fallos en los sistemas de fichero aparecieron los sistemas RAID (*Redundant Array of Inexpensive Disks*).

A medida que pasaba el tiempo los programas y datos eran más complejos y grandes por tal motivo se requería de un almacenamiento que garantizara un cierto número de condiciones, volúmenes de datos y que permitiera operaciones complejas.

Adicionalmente cada usuario que necesitaba acceder a los datos debía tener su trabajo protegido de las operaciones que hicieran el resto de usuarios, respondiendo a estas necesidades, surgieron las bases de datos jerárquicas donde los datos se situaban siguiendo una jerarquía, estas bases de datos jerárquicas tenían el problema que los accesos a los datos eran unidireccionales, y era más complicado hacer el camino inverso, pero posible, aunque el tiempo de cálculo era mayor, (Por ejemplo, era fácil saber que cuentas tenía un cliente, pero no tan fácil saber de qué cliente era una cierta cuenta).

Para dar absoluta libertad a las relaciones entre tablas surgieron las bases de datos relacionales las cuales trajeron dos cosas muy importantes: las propiedades ACID (atomicidad, consistencia, aislamiento, durabilidad) y un lenguaje standard de acceso a los datos (SQL, lenguaje de consulta estructurado).

Atomicidad: Asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.

Consistencia: Asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.

Aislamiento: Asegura que una operación no puede afectar a otras.

Durabilidad: Asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

SQL: Agrupación de instrucciones SQL (*Selects, Inserts y Updates*). Para soportar la creación, manipulación y mantenimiento de una base de datos se debe utilizar un SGBD (Sistema Gestor de Bases de Datos).

(SISTEMA GESTOR DE BASES DE DATOS)

El sistema gestor de bases de datos (S.G.B.D.) ha sido y a introducido, y su importancia destacada en todas las características que debe poseer una base de datos. Existen muchas definiciones en la bibliografía pero podemos citar la más apropiada como: Programa que proporciona la interfaz entre los datos almacenados en la base de datos, y los programas de aplicación o interfaces y consultas hechos al sistema.

CARACTERÍSTICAS DE UN SGB

Interactuar con el Sistema Operativo: como se ha indicado, el S.G.B.D. no es más que un programa. El Sistema Operativo es el programa principal que se encarga de controlar que el computador funciona bien, entre otras cosas permitiendo el acceso a los dispositivos de entrada y de salida, como el teclado, el ratón, el monitor, y los dispositivos de almacenamiento: el disco duro, las disqueteras, el CD-ROM, las cintas magnéticas, etc. A sí, par a asegurar que no pasan cosas raras, el único que puede tocar estos dispositivos es el Sistema Operativo (S.O.).

Dado que el S.G.B.D. necesita almacenar datos en el disco duro, por ejemplo, debe interactuar con el S.O. para poder acceder al disco duro y que allí se almacenen los datos que se quieren. Igualmente debe contactar con el S.O. siempre que desee recuperar algún dato de estos dispositivos.

Mantener la integridad: Como ya se ha dicho, debe mantener las restricciones de integridad propias de la aplicación concreta que sea. Por ejemplo, evitar que la edad de un cliente supere los 90 años.

Mantener la seguridad: Evitar accesos fraudulentos a los datos, así como la extracción de información codificada.

Permitir las copias de seguridad. Dado que un computador no es un sistema infalible, y puede romperse por causas propias (fallo de un circuito), o ajenas (aumento de la tensión en la red eléctrica), es posible que los datos almacenados por él lleguen a corromperse con la consiguiente pérdida de información y los problemas que ello puede acarrear a la empresa. Para evitar estos desagradables resultados, es buena idea el efectuar una copia de los datos a un dispositivo auxiliar de almacenamiento, pensado precisamente para guardar fiel copia del contenido de la base de datos en un momento determinado. Si los datos originales se destruyen, bastará volcar la copia sobre el disco duro del computador central, con lo que los datos volverán a tomar la misma forma que cuando se efectuó la copia. De esta manera, para que la base de datos recupere la forma que tenía en el momento en que quedó destruida, bastará con efectuar los cambios que se hicieron en el tiempo transcurrido desde la copia de seguridad que se acaba de volcar.

Controlar la concurrencia: Como ya se explicó anteriormente, debe permitirse el acceso simultáneo a los datos por parte de varios usuarios, lo que conlleva numerosos problemas de coherencia y coordinación. el S.G.B.D. debe controlar que la información representada por los datos al final de cada acceso de usuario siga siendo consistente.

Suministrar mecanismos que faciliten la interacción con la base de datos: Estos mecanismos suelen venir dados en forma de lenguajes de manipulación y definición de datos. Además, suministran independencia de los datos, en el sentido de que, a pesar de la evolución del es quema de los datos, las aplicaciones deben sufrir las mínimas modificaciones imprescindibles. Por ejemplo, si las aplicaciones antiguas están pensadas para trabajar sobre números de teléfono expresados mediante dígitos, ¿qué ocurre si se decide cambiar todos los números a formato textual? En este caso, debe haber un mecanismo que oculte a las aplicaciones antiguas el nuevo formato de los números de teléfono, y le haga vez el formato antiguo; en definitiva, debe haber algo que suministre a las aplicaciones antiguas una visión ligeramente distinta de lo que hay realmente almacenado en la base de datos. Actualmente en el mercado se encuentran muchos motores de bases de datos, al momento de decidirse por cual escoger para implementar un sistema de información es importante realizar una investigación de las ventajas, desventajas, costos que implica determinada tecnología.

COMPARACIÓN DE LOS SGBD EXISTENTES EN EL MERCADO

En la industria del software, existen sistemas manejadores de bases de datos (SGBD) libres y propietarios, para lo cual tenemos que tener en cuenta al momento de implantar un sistema con acceso a base de datos.

Software Libre: software que no requiere valor para poder ser utilizado.
Software propietario: software que para poder utilizarlo requiere costo.

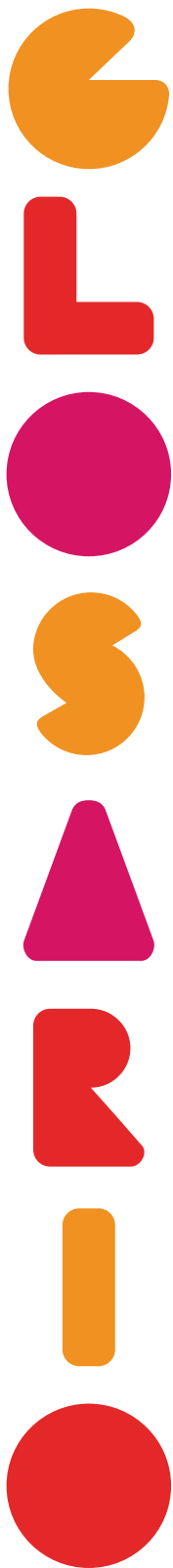
Motor Bases de Datos	Libre	Propietario	Característica	Inconveniente
MySql			<div>Más utilizado al momento de realizar aplicaciones de tipo web.</div> <div>El motor es una versión web y ofrece soporte a las bases de datos creadas en su servidor web.</div> <div>Lo pueden utilizar sistemas operativos Windows y Linux</div> <div>Permite Almacenar la estructura de la base de datos y utiliza tablas. Capacitación en la herramienta.</div> <div>Descargue fácil desde internet.</div>	
Postgre Sql			<div>Orientado a objetos.</div> <div>Crea objetos, no tablas.</div> <div>Las tuplas son instancias de un objeto.</div> <div>Tiene integridad referencial, vistas, multitud de funcionalidades.</div> <div>Opera con el modo cliente-servidor.</div>	Es lento, pesado

Motor Bases de Datos	Libre	Propietario	Característica	Inconveniente
Oracle			<div>Soporta grandes volúmenes de datos.</div> <div>Rápido y efectivo al momento de procesar datos.</div> <div>Permite hacer reportes plsql propio de oracle.</div> <div>Manejado por versiones.</div> <div>Permite ir mas allá de las bases de datos (bodegas)</div>	
Sql Server			<div>Soporta grandes volúmenes de datos.</div> <div>Rápido y efectivo al momento de procesar datos.</div> <div>Permite ir más allá de las bases de datos (bodegas).</div> <div>Manejado por versiones (2000 hasta 2010).</div> <div>Propietario: Microsoft.</div>	
SqlServerExpress			<div>Limita volúmenes de datos.</div> <div>Rápido y efectivo al momento de procesar datos.</div> <div>Manejado por versiones (2000 hasta 2010)</div> <div>Propietario: Microsoft</div>	

Tabla 1: Tomada (<http://www.monografias.com/trabajos29/comparacion-sistemas/comparacion-sistemas.shtml>)

CONCLUSIONES

Al momento de iniciar un desarrollo de software que tenga inmerso el almacenamiento de información es necesario indagar el tipo de Gestor de Bases de Datos que se utilizará, realizando una exhaustiva comparación de los existentes en el mercado y de acuerdo las necesidades del cliente determinar el SGBD a utilizar para el proyecto.



GLOSARIO

EACID: conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

En concreto ACID es un acrónimo de *Atomicity, Consistency, Isolation and Durability*: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

Administración de bases de datos: La gestión de la implementación física de un sistema de bases de datos, lo que incluye el diseño físico de la base de datos y su implementación, la configuración de los controles de seguridad e integridad, la monitorización de las prestaciones del sistema y la reorganización de la base de datos según sea necesario.

Administración de datos: La gestión de los recursos de datos, lo que incluye la planificación de la base de datos, el desarrollo y el mantenimiento de estándares, políticas y procedimientos, así como el diseño conceptual y lógico de la base de datos.

NFS: *El Network File System* (Sistema de archivos de red), o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

RAID: (*Redundant Array of Inexpensive Disks*, «conjunto redundante de discos independientes») hace referencia a un sistema de almacenamiento que usan múltiples discos duros o SSD entre los que se distribuyen o replican los datos. Dependiendo de su configuración (*a la que suele llamarse «nivel»*), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos, mayor throughput (rendimiento) y mayor capacidad.



1. Silberschatz, A., Korth. H., Sudarshan. S., (2007). Principios de bases de dato. McGraw-Hill

REFERENCIAS

2. <http://es.wikipedia.org/wiki/>

3. <http://gva1.dec.usc.es>



El lenguaje de programación a utilizar es ASP.Net C#

C# es un lenguaje de programación orientado a objetos desarrollado estandarizado por Microsoft como parte de su plataforma .NET

C# elimina muchos elementos que otros lenguajes incluyen y que innecesarios en .NET. Por ejemplo:

- El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres.

Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años han ido demostrándose que son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ han que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción `foreach` que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico `string` para representar cadenas o la distinción de un tipo `bool` específico para representar valores lógicos.



Software Propietario

El Software propietario es aquel que algún individuo o compañía retiene el derecho de autor exclusivo sobre una pieza de programación, al mismo tiempo que niega a otras personas el acceso al código fuente del programa y el derecho a copiarlo, modificarlo o estudiarlo. El término "propietario" alude a que está "poseído y controlado privadamente". No obstante, el programa propietario aunque su código fuente se haya hecho

ENVIA SOLICITUDES

TIPO DE LICENCIA

ENVIA RESPUESTAS

CLIENTE

- Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos,

ENVIA SOLICITUDES

ENVIA RESPUESTAS

LÍDER DEL PROGRAMA ADSI

Vanessa Cristina Miranda Cano
vanessa24@misena.edu.co

COMPILACIÓN Y PREPARACIÓN

Leydy Carolina Muñoz Pachajoa

DISEÑO EDITORIAL Y PORTADA

Ricardo Burbano Martínez
ribuma@gmail.com

ASESORÍA PEDAGÓGICA

Claudia Herrera Cifuentes
pipelore@yahoo.com

ILUSTRACIÓN PORTADA

Saúl Suaza
ssuaza@gmail.com

LÍDER LÍNEA DE PRODUCCIÓN

Ilina Eneth Molina Cuarta
ilmocu@sena.edu.co

DIAGRAMACIÓN

Sergio A. Areiza Serna
sergioareiza@hotmail.com

Ricardo Burbano Martínez
ribuma@gmail.com

- ✓ Propiedad y decisión de uso de la empresa
- ✓ Soporte para todo tipo de hardware
- ✓ Menor necesidad de técnicos especializados
- ✓ Mejor protección de las obras con copyright