

Universidad Gerardo Barrios

Carrera: Ingeniería en Sistema y Redes Informáticas

Materia: Ingeniería en Software

Actividad: PHP y SGBD

Docente: Ing. Gisela Espinoza

Grupo: "A"

Ciclo: V

Integrantes:

- **José Daniel Lazo Arias**
- **Adonay Turcios Gálvez**
- **David Alexis Gallegos Gonzales**
- **Gerardo Steven Mejía Perdomo**

Año: 2018

2018

INTRODUCCION

Anteriormente en versiones iniciales de PHP, la única manera de acceder a un SGBD era a través de una biblioteca específica, que contenía las funciones necesarias. A esas bibliotecas se les suelen llamar también extensiones. En algunos casos esas extensiones se incluían en las distribuciones oficiales de PHP y para poder utilizarlas era suficiente con incluir la directiva correspondiente en el archivo de configuración php.ini. En otros casos esas extensiones no estaban incluidas en las distribuciones oficiales de PHP, pero estaban incluidas en PEAR o PECL y para poder utilizarlas era necesario instalarlas por separado.

Con el paso del tiempo, este enfoque fue mostrando sus limitaciones. El principal inconveniente de este enfoque es que cada extensión está estrechamente vinculada a cada SGBD y si se quiere cambiar de SGBD es necesario reescribir completamente la aplicación (los nombres de cada función son diferentes, el orden de los argumentos distinto, las funcionalidades disponibles son diferentes, etc.). Además, en algunos casos, las extensiones no están mantenidas adecuadamente (bugs de seguridad, etc.).

Actualmente (diciembre de 2017), PHP se sigue distribuyendo con muchas de estas extensiones, pero en su lugar se recomienda utilizar la biblioteca PDO que se comenta a continuación.

Definición de DBMS

(Data Base Management System). Son las siglas en inglés para los Sistemas de Gestión de Bases de Datos (SGBD). Bajo este nombre se conoce a productos de fabricantes como Oracle, Sybase, Informix, Ingres, Borland, Microsoft, IBM, etc.

Sistema de administración de bases de datos. Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta solicitudes de la aplicación y ordena al sistema operativo transferir los datos apropiados.

Los DBMS pueden trabajar con lenguajes de programación tradicionales (COBOL, C, etc.) o pueden incluir su propio lenguaje de programación. Por ejemplo, dBASE y Paradox son programas de base de datos con un DBMS, un lenguaje completo de programación y un lenguaje de cuarta generación, haciendo de ellos sistemas completos de desarrollo de aplicaciones. Los comandos de los lenguajes de cuarta generación permiten a los usuarios crear en forma interactiva archivos de bases de datos, editarlos, formular preguntas e imprimir informes sin necesidad de programación. Miles de aplicaciones han sido desarrolladas en ambientes como éstos.

Está compuesto por:

DDL: Lenguaje de Definición de Datos.

DML: Lenguaje de Manipulación de Datos.

SQL: Lenguaje de Consulta.

- Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.
- Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de almacenamiento.

• Manipular una base de datos: incluye funciones como consulta, actualización, etc. de bases de datos.

Existen distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información**: los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.
- **Independencia**: la independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia**: en aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.
- **Seguridad**: la información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de transacciones**: una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que tras una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- **Tiempo de respuesta**: es deseable minimizar el tiempo que el SGBD demora en proporcionar la información solicitada y en almacenar los cambios realizados.

Ventajas:

- Proveen facilidades para la manipulación de grandes volúmenes de datos. Entre éstas:
    - Simplifican la programación de equipos de consistencia.
    - Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores correctamente, etc.
    - Organizan los datos con un impacto mínimo en el código de los programas.
    - Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Desventajas:

- Típicamente, es necesario disponer de una o más personas que administren la base de datos, de la misma forma en que suele ser necesario en instalaciones de cierta embergadura disponer de una o más personas que administren los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.
- Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar algún otro método para guardarlos y consultarlos.

- Complejidad: algunos DBMS pueden ser complejos, por lo que los desarrolladores y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
- Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder ejecutarse.
- Coste del hardware adicional: los requisitos de hardware para ejecutar un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

SGBD (Sistema Gestor de Bases de Datos).

Una aplicación sencilla puede trabajar únicamente con ficheros, pero en cuanto el servicio deja de ser trivial aparece la necesidad de utilizar un SGBD que organice la información con la que se trabaja.

La historia de las bases de datos es larga, pero en cuanto a lo que una introducción a aplicaciones web se refiere se puede resumir en diferentes modelos:

SGBD relacional:

Las más extendidas y con una larga vida por delante. Guardan la información en registros de tablas y mediante SQL se realizan consultas u operaciones para manipular los datos. Dependiendo de la herramienta utilizada hay grandes diferencias en cuanto a sus capacidades y administración, pero en general, para las aplicaciones web prima la velocidad sobre las características.

Ejemplos de uso: MySQL y PostgreSQL

SGBD orientado a objetos:

En lugar de almacenar registros directamente guardan y recuperan objetos. De manera que cuando se utiliza un lenguaje de programación orientado a objetos nos ahorramos el trabajo de adaptar las entidades con las que trabajamos al modelo relacional.

Ejemplos de uso: DB4O y Zope Object Database

SGBD NoSQL:

Aquí se agrupan un conjunto de técnicas diferentes que tienen en común que no cumplen con los requisitos ACID y/o no estructuran la información en tablas como las bases de datos relacionales. La ventaja es que consiguen romper algunas de las limitaciones de los sistemas relacionales en cuanto a escalabilidad y rendimiento permitiendo crear ingentes bases de datos distribuidas.

Ejemplos de uso: Apache Cassandra y Apache CouchDB

Servidor web

El servidor web se encarga la comunicación a través de la red con el navegador del usuario. Normalmente escucha en el puerto TCP 80 cuando se trata de una conexión HTTP, y en el TCP 443 cuando se trata de una HTTPS, aunque este parámetro es configurable.

Cuando recibe una petición del usuario puede atenderla de manera estática cuando contesta con un fichero que está en el sistema de archivos (la descarga de una imagen, un fichero css o html) o de manera dinámica cuando se ejecuta algún programa y se envía al cliente el resultado de dicha ejecución.

El servidor web evidentemente es capaz de atender muchas peticiones de forma concurrente pudiendo realizar diferentes funciones habituales entre las que se encuentran:

- Registro de actividad y errores
- Control de acceso basado en la dirección del cliente, contenido o usuario/contraseña
- Virtual Hosts, para mantener diferentes webs (por ejemplo http://elpuig.xeill.net y http://blog.elpuig.xeill.net)
- Proxy, para reenviar las peticiones a otro servidor
- Reescritura de URLs
- Alias o mapeados de rutas

Cuando una aplicación web necesita conservar información de forma permanente para recuperarla posteriormente, suele ser conveniente recurrir a un sistema gestor de bases de datos (SGBD).

Normalmente los SGBD son aplicaciones externas que se instalan y administran de forma separada. PHP permite utilizar la mayoría de los SGBD más conocidos, libres (MySQL, PostgreSQL, MariaDB, Firebird, etc) o comerciales (Oracle, MS SQL Server, Ingres / Actian, etc) como se puede consultar en el apartado sobre acceso a bases de datos del manual de PHP.

Un caso particular es SQLite, que no es una aplicación externa sino una biblioteca en C que implementa un motor de bases de datos SQL. Eso quiere decir que PHP puede gestionar bases de datos directamente, sin necesidad de recurrir a SGBD externos.

Aunque PHP siempre ha permitido utilizar numerosos SGBD, la forma de hacerlo ha ido variando con el tiempo.

Acceso mediante PDO

Para poder escribir programas independientes del SGBD elegido, es necesario utilizar una capa de abstracción que permita acceder de una forma común a cualquier SGBD. Esta necesidad no es exclusiva de PHP, sino que afecta a cualquier aplicación que necesita trabajar con un SGBD y no tiene solución completa pues las diferencias entre los SGBD no permiten olvidarse de cuál estamos utilizando realmente.

Una de estas capas de abstracción es ODBC (Open DataBase Connectivity), una API independiente del sistema operativo, lenguaje de programación o SGBD utilizada que se empezó a desarrollar en 1992. En PHP existe una extensión llamada también ODBC que permite conexiones ODBC (creo que esta extensión se incorporó en PHP 3.0, publicado en junio de 1998, en cualquier caso estaba activada en PHP 4.0, publicado en mayo de 2000). El inconveniente de ODBC es que introduce un elemento más en la cadena, el controlador ODBC del sistema operativo, lo que puede ralentizar el rendimiento.

La capa de abstracción propia de PHP se llama PDO. PDO, biblioteca orientada a objetos, se podía utilizar como extensión PECL de PHP 5.0, publicado en agosto de 2004 y está incluida en PHP desde PHP 5.1, publicado en noviembre de 2005. PDO incluye controladores para acceder a las bases de datos más populares y la lista se va ampliando con el tiempo (aunque no incluye todos los SGBD que disponen de extensiones específicas). Utilizando PDO no podemos olvidarnos completamente del SGBD utilizado, pero la mayor parte del código es independiente del SGBD y sólo en algunas partes del programa (en la conexión con el SGBD o en la creación de tablas, por ejemplo) el código es específico del SGBD.

PHP y MySQL

PHP ha tenido tres extensiones para acceder a la base de datos MySQL:

- Extensión mysql: Esta extensión fue la primera extensión para MySQL incluida en PHP (creo que esta extensión se incorporó en PHP 3.0, publicado en junio de 1998, en cualquier caso estaba activada en PHP 4.0, publicado en mayo de 2000). En diciembre de 2012 se decidió eliminar la extensión. Desde PHP 5.5 (publicado en junio de 2013) la extensión estuvo desaconsejada y su uso generaba un aviso E_DEPRECATED. Desde PHP 7.0 (publicada en diciembre de 2015), ya no está incluida en PHP.
- Extensión mysqli (improved mysql): Esta extensión se incluyó por primera vez en PHP 5.0 (publicado en julio de 2004). Las principales ventajas de esta extensión son la posibilidad de utilizar una sintaxis orientada a objetos, sentencias preparadas, transacciones y otras.
- Extensión PDO: esta extensión, que permite trabajar con diferentes bases de datos, incluye un controlador para MySQL desde que fue incluida en PHP 5.1 (publicado en noviembre de 2005).

Aunque la extensión mysqli se sigue incluyendo en PHP, en su lugar se recomienda utilizar acceder a MySQL mediante la extensión PDO.

Internamente, estas bibliotecas utilizan un controlador para acceder a MySQL. Antiguamente el controlador se llamaba libmysqlclient y había sido desarrollado por la empresa MySQL AB, pero a partir de PHP 5.3 (publicado en junio de 2009), el controlador es mysqlnd (MySQL Native Driver), desarrollado por el equipo de PHP. Nuestros programas no pueden utilizar directamente estos controladores.

Bibliografías:

- ✓ http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=1048
- ✓ https://plataformasistemas.wordpress.com/bases-de-datos/ques-es-un-dbms/
- ✓ https://elpuig.xeill.net/Members/vcarceler/asix-m09/uf1/nf1/a1
- ✓ http://www.mclibre.org/consultar/php/lecciones/php-basesdedatos.html