

Tarea 2: Trabajar con Bases de Datos en PHP

Francisco Javier Sueza Rodríguez

18 de diciembre de 2023

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Desarrollo Web en Entorno Servidor
Tema:	Tema 2 - Trabajar con Bases de Datos en PHP.

1. Caso Práctico

Ada está explicando el proyecto que van a empezar a realizar. **María** y **Juan**, están escuchando atentamente:

- En el proyecto de la asociación Respira una de las partes más peliagudas sería la gestión del seguimiento de los usuarios, por lo que quiero que empecéis por ahí. Es una de las partes más sensibles de la aplicación, dado que se va a gestionar información sensible de cada usuario -comenta **Ada**.

- En estos casos, ¿la información de seguimiento de los usuarios no suele borrarse verdad? Creo que **Clara**, la coordinadora de la asociación me comentó algo al respecto. - pregunta **Juan**.

- Si, efectivamente -comenta **Ada**-. La información de seguimiento no se puede borrar, lo que se hace es que se archiva. Se guarda en otra tabla por si hay que consultarla posteriormente una vez que deja de ser necesaria.

2. Ejercicios

2.1. Pregunta 1

Cita al menos tres sistemas que podrían utilizarse para organizar la información que maneja la asociación Respira sobre sus usuarios, sus empleados, y la información que manejan descrita en el apartado anterior (Descripción de la tarea). Estos sistemas no tienen porqué ser necesariamente una base de datos relacional (existen otras formas de almacenar la información en un ordenador), aunque una base de datos relacional es uno de los sistemas de almacenamiento de la información que deberías contemplar. Reflexiona sobre cuales son los sistemas más convenientes de cara a desarrollar una aplicación web desde PHP.

2.1.1. Solución

La asociación Respira podría usar diferentes sistema para el manejo de los datos, pudiendo tener en consideración alguno de los siguientes:

- **MySQL**: esta **base de datos relacional SQL** es una de las más empleadas en el mundo, junto con Oracle DB. Sería una muy buena opción ya que se integra perfectamente en el stack LAMP y dadas las características de la información que se quiere almacenar, una base de datos relacional sería la más adecuada.
- **MongoDB**: esta es una base de datos **No-SQL** basada en documentos. La ventajas de este tipo de bases de datos es que son menos costosas a nivel de computación y además más flexibles. En nuestro caso, los datos están muy estructurados, por lo que este tipo de bases de datos no representaría una ventaja, ya que su desempeño sería peor para este tipo de datos. Además, no proporcionan integridad de datos, algo que en el caso de la asociación Respira es importante.

Amazon Aurora: otra opción sería utilizad una **base de datos SQL en la Nube**, como es el caso de Amazon Aurora, o algunas de las decenas que podemos encontrar actualmente. Esta base de datos es relacional, por lo que cumpliría nuestras nuestros requerimientos, pero aportar ciertas características como flexibilidad, escalabilidad, adaptabilidad, etc., que dada la naturaleza del caso de uso que estamos tratando tampoco se van a aprovechar. Por lo que cabría valorar si la diferencia del costo entre el uso de una base de datos en la nube o una base de datos tradicional.

2.2. Pregunta 2

Si usamos una base de datos MySQL o MariaDB, ¿con qué tipo de sistema de almacenamiento de información de los citados por ti en el ejercicio anterior encajaría?

2.2.1. Solución

Tanto MySQL como MariaDB son bases SQL tradicionales. Son prácticamente lo mismo, MariaDB surgió como alternativa a MySQL tras la adquisición por parte de Oracle de ésta, pero a nivel funcional y de características son prácticamente idénticas.

2.3. Pregunta 3

Desde PHP se puede acceder a una base de datos MySQL o MariaDB a través de diferentes formas. Cita al menos dos formas diferentes y explica sus diferencias.

2.3.1. Solución

Las dos principales formas de acceder a una base de datos SQL en PHP son:

- **MySQLi**: esta es una extensión introducida desde PHP 5 y que trabaja directamente con bases de datos SQL. Permite una mayor rapidez en las consultas o modificaciones y permite usar todas las características del motor de MySQL o MariaDB. Proporciona dos tipos de acceso, usando objetos o funciones. El problema con esta forma de acceso es que si cambiamos la base de datos que empleamos tendremos que programar de nuevo las operaciones que se realicen en la base de datos.
- **PDO**: esta forma de acceso añade un nivel de abstracción, haciendo que las operaciones sobre ésta se realicen de forma más genérica y puedan ser utilizadas por diferentes bases de datos. A diferencia de MySQLi, solo ofrece la opción de trabajar mediante objetos con la base de datos. Las operaciones son más lentas que con MySQLi, pero por otro lado, no tendremos que volver a programar las interacciones con la base de datos si cambiamos ésta.

2.4. Pregunta 4

Investiga que diferencias hay entre MySQL y **Firestore**, y reflexiona sobre que limitaciones o ventajas tendría el uso de una u otra de cara a desarrollar la aplicación web para esta empresa.

2.4.1. Solución

Firestore es una base de datos **NO-SQL** basada en la nube. Como hemos comentado antes, este tipo de bases de datos suelen basarse en documentos, almacenando archivos de tipo JSON en la mayoría de los casos. Es rápida a la hora de manejar grandes cantidades de información. Tienen tanto versiones gratuitas como de pago, aunque la versión gratuita tiene limitaciones. Tiene soporte para los principales lenguajes de programación usados en el backend del desarrollo web, como Java, Node (Javascript), Go, ..etc.

MySQL es una base de datos **relacional SQL**. Almacena la información en forma de tablas y registros y permite un manejo rápido de información compleja y muy estructurada. Soporta una gran variedad de lenguajes de programación y utiliza SQL como lenguaje de consulta, modificación, etc. Es una base de datos de código abierto y gratuita. Tiene soporte para la mayoría de lenguajes de programación.

Si **analizamos las diferencias**, claramente **MySQL** sería la **opción adecuada** para nuestro caso.

Por un lado, dadas las **características de los datos** que vamos a manejar, necesitamos una base de datos que trabaje rápidamente con datos complejos y estructurados. En este aspecto, **MySQL** trabaja mucho más rápido con este tipo de datos de **Firestore**, que esta más orientada a los documentos y a procesar grandes cantidades de datos en este formato rápidamente.

Además, dada la naturaleza de la asociación, **no necesitamos** algunos de las **características** que ofrece **Firestore** y que están mas orientados a empresas de tamaño medio o grande, como una gran escalabilidad horizontal, adaptabilidad, etc. En cambio **si podemos sacar partido** de algunas de las características que aporta **MySQL**, como la integridad de datos, algo que dado el tipo de datos sensible que va a manejar la asociación es de suma importancia.

Por otro lado, **MySQL** es de código abierto y gratuita. **Firestore** también ofrece una versión gratuita, pero habría que analizar con mayor profundidad si esta versión sería suficiente para nuestra aplicación para la asociación respira.

En definitiva, **MySQL** es la opción más adecuada para la asociación Respira, dado las necesidades de la asociación y el tipo de datos que se van a manejar.