

Desarrollo de Software VII

Laboratorio # 20—Prof. Regis Rivera

Objetivo: Integración de base de datos no relacionales (NoSQL) con PHP

Parte I – Introducción a base de datos no relacionales

Definición

Las **bases de datos no relacionales** son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas. Esto no significa que no puedan usar el lenguaje SQL, pero no lo hacen como herramienta de consulta, sino como apoyo. Por ello también se les suele llamar **NoSQL** o «no solo SQL».

Tienen una gran escalabilidad y están pensadas para la **gestión de grandes volúmenes de datos**. Por otro lado, a diferencia de las bases de datos relacionales no cumple con el estándar ACID de atomicidad, consistencia, aislamiento y durabilidad.

Características

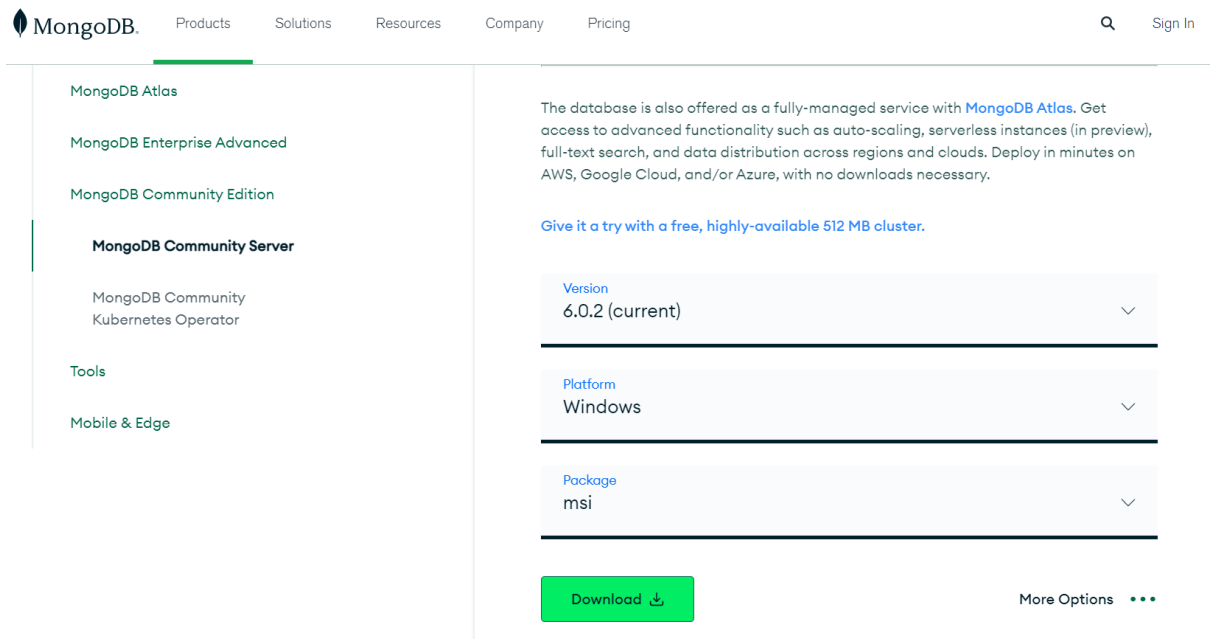
- La información no se almacena en tablas sino a través de documentos.
- Son bases de datos muy útiles para organizar y gestionar información no estructurada, o cuando no se tiene una noción clara de los datos a almacenar.
- Son bases de datos con alto grado de escalabilidad y están diseñadas para soportar grandes volúmenes de datos.
- No utilizan el lenguaje SQL para consultas, aunque sí lo pueden usar como herramienta de apoyo.
- Es un sistema de almacenamiento de datos relativamente nuevo, y como tal, todavía no posee un sistema estandarizado.
- A diferencia de las no relacionales, no garantizan el cumplimiento de las cualidades ACID, esto es, atomicidad, consistencia, integridad y durabilidad.

Tipos


Clave-valor	Documentos	Gráficos	En memoria
Se trata de bases de datos no relacionales que almacenan la información en base a pares de clave valor. Es decir, cada clave sirve como un identificador único, y a cada una de ellas se le aplica un valor. Son especialmente usadas a la hora de almacenar datos de juegos, aplicaciones o aparatos que funcionan mediante el internet de las cosas (IoT).	En una base de datos relacional basada en documentos la información se representa como objetos o documentos JSON. Su principal ventaja es que los documentos son de naturaleza flexible, semiestructurada y jerárquica, lo que facilita a los desarrolladores las tareas de almacenamiento, gestión y consulta de datos. Es un modelo usado habitualmente en sistemas de administración de contenidos o para gestionar perfiles de usuarios.	Las bases de datos no relacionales basadas en gráficos están pensadas para crear relaciones y navegar por ellas. Las entidades de datos se almacenan mediante nodos y los bordes son los que crear las relaciones entre entidades. Con frecuencia las bases de datos gráficas se emplean en redes sociales, sistemas de detección o prevención de fraudes o sistemas de recomendaciones.	Son bases de datos no relacionales diseñadas para ofrecer respuestas en milisegundos y soportar grandes picos de tráfico. Un ejemplo de bases de datos en memoria son las empleadas en tablas de clasificaciones de juegos o en herramientas para hacer análisis en tiempo real.

Parte II – Instalación de motor de base de datos (MongoDB)

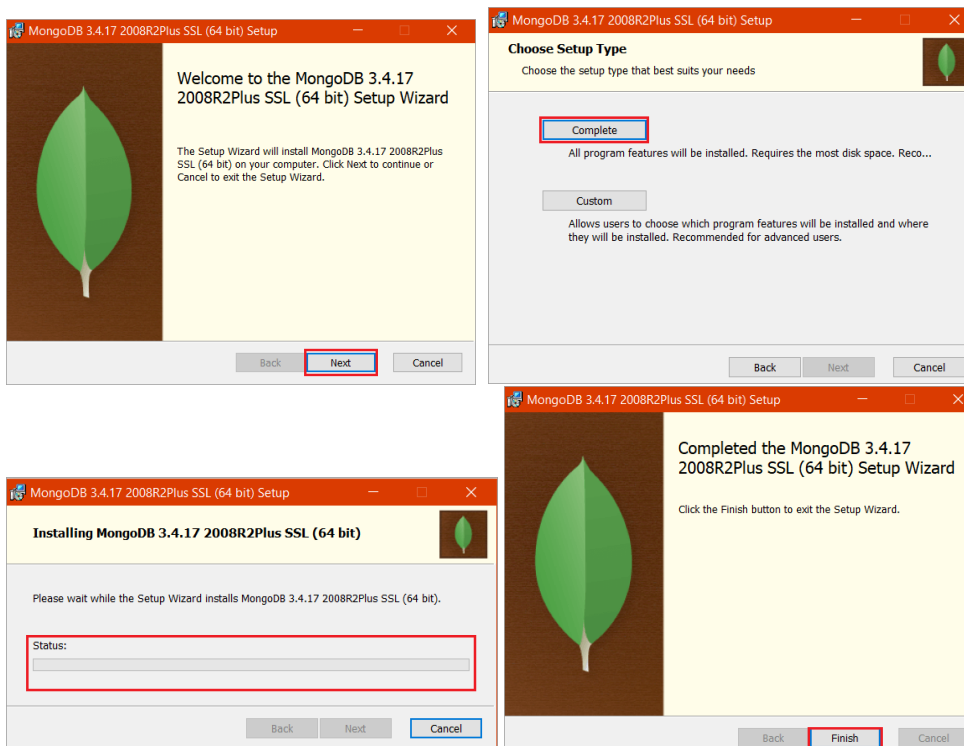
1. Ir al siguiente URL <https://www.mongodb.com/try/download/community>
2. Ir a la sección MongoDB Community Server
3. Descargar la versión MSI para Windows (si es Linux, cambiar la opción en Platform)



4. Esperar termine de descargar el instalador correspondiente

 `mongodb-windows-x86_64-6.0.2-signed.msi`

5. Abrir el instalador y seguir el paso a paso de la misma




Parte III – Configuración de PHP

- Ir al siguiente enlace: <https://pecl.php.net/package/mongodb> para descargar el driver de conexión de MongoDB con PHP, específicamente un DLL (observar aparecen varias versiones, nosotros instalaremos específicamente la versión 1.13.0 la cual se puede acceder desde <https://pecl.php.net/package/mongodb/1.13.0/windows>)
- Descargar la DLL (Thread Safe) según su versión de PHP y su arquitectura de SO


DLL List	
PHP 8.1	8.1 Non Thread Safe (NTS) x64
	8.1 Thread Safe (TS) x64
	8.1 Non Thread Safe (NTS) x86
	8.1 Thread Safe (TS) x86
PHP 8.0	8.0 Non Thread Safe (NTS) x64
	8.0 Thread Safe (TS) x64
	8.0 Non Thread Safe (NTS) x86
	8.0 Thread Safe (TS) x86
PHP 7.4	7.4 Non Thread Safe (NTS) x64
	7.4 Thread Safe (TS) x64
	7.4 Non Thread Safe (NTS) x86
	7.4 Thread Safe (TS) x86

- Descomprimir el archivo descargado

 php_mongodb-1.13.0-7.4-ts-vc15-x64.zip

- Extraer la DLL llamada php_mongodb.dll y colocarla en la carpeta \xampp\php\ext (D:) > xampp > php > ext

Nombre

 php_mongodb.dll

- Editar el archivo php.ini



- Buscar esta sección del archivo php.ini

php.ini: Bloc de notes

Archivo Edición Formato Ver Ayuda

```
;extension=sqllite3
;extension=tidy
;extension=xmlrpc
;extension=xsl

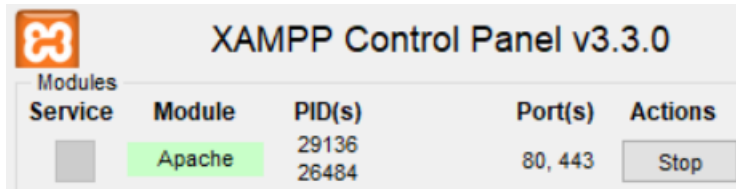
;;;;;;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;;;;;;
asp_tags=Off
display_startup_errors=On
track_errors=Off
y2k_compliance=On
allow_call_time_pass_reference=Off
safe_mode=Off
safe_mode_gid=Off
safe_mode_allowed_env_vars=PHP_
safe_mode_protected_env_vars=LD_LIBRARY_PATH
error_log="\\xampp\\php\\logs\\php_error_log"
register_globals=Off
register_long_arrays=Off
magic_quotes_gpc=Off
magic_quotes_runtime=Off
magic_quotes_sybase=Off
extension=php_openssl.dll
extension=php_ftp.dll
```

12. Agregar la siguiente línea debajo de php_ftp.dll

```
extension=php_mongodb.dll
```

13. Guardar cambios en php.ini

14. Reiniciar apache (stop/start)



15. Validar que mongoDB esté integrado correctamente. Crear un código en un archivo temporal en php con la función phpinfo(); y de encontrar una sección dedicada a mongoDB, significa está integrado correctamente

mongodb

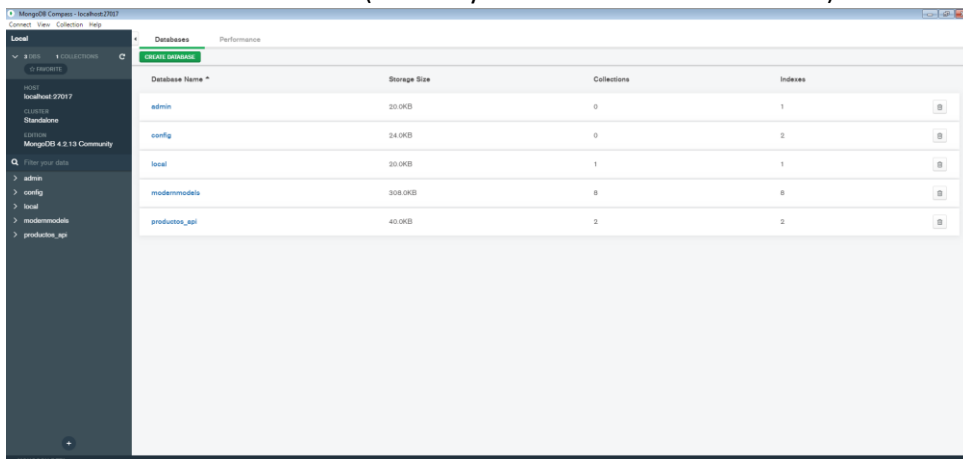
MongoDB support	enabled	
MongoDB extension version	1.13.0	
MongoDB extension stability	stable	
libbson bundled version	1.21.1	
libmongoc bundled version	1.21.1	
libmongoc SSL	enabled	
libmongoc SSL library	OpenSSL	
libmongoc crypto	enabled	
libmongoc crypto library	libcrypto	
libmongoc crypto system profile	disabled	
libmongoc SASL	enabled	
libmongoc ICU	disabled	
libmongoc compression	disabled	
libmongocrypt bundled version	1.3.2	
libmongocrypt crypto	enabled	
libmongocrypt crypto library	libcrypto	

Directive	Local Value	Master Value
mongodb.debug	no value	no value
mongodb.mock_service_id	Off	Off

Parte IV – Creación de la base de datos y colección

16. Abrir Mongo DB (Mongo DB Compass)

17. Conectarse a localhost:27017 (usuario y contraseña vacíos o defaults)



18. Crear nueva base de datos



19. Configurar nueva base de datos (usuarios_api) y nueva colección (usuarios)

Create Database

Database Name

Collection Name

☐ Capped Collection ⓘ

☐ Use Custom Collation ⓘ

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

CANCEL CREATE DATABASE

20. Observar aparece en el panel izquierdo de mongo DB Compass y en el listado de colecciones



21. Click en la base de datos usuarios_api (en el listado de la derecha), observar la colección usuarios no posee documentos aun

Collections

CREATE COLLECTION

Collection Name ^	Documents	Avg. Document Size	Total Document Size
usuarios	0	-	0.0 B

22. Click en la colección usuarios (en el listado), observar el detalle de la colección indicado aun no existen datos (documentos)

usuarios_api.usuarios

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

0 FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW 0 0

Displaying documents 0 - 0 of N/A C REFRESH

This collection has no data


It only takes a few seconds to import data from a JSON or CSV file

Import Data

Parte V – Codificación de la aplicación Web

23. Formulario y procesamiento del mismo, para insertar registro y leer los existentes


```
1 <?html>
2 <?head>
3 <title>Laboratorio 20</title>
4 </head>
5 <?body>
6 <form action = "lab201.php" method = "POST">
7 Nombre: <input type = "text" name="nombre" required><br>
8 Apellido: <input type = "text" name="apellido" required><br>
9 Email: <input type = "email" name="email"><br>
10 Edad: <input type = "number" name="edad" min="0" max="120"><br><br>
11 <input type = "submit" name="guardar" value="Guardar datos">
12 </form>
13 <?php
14 include("UsuariosMDB.php");
15 $usrs = new UsuariosMDB();
16
17 if(array_key_exists('guardar', $_POST)){
18     $usrs->insertarRegistro($_REQUEST['nombre'],$_REQUEST['apellido'],$_REQUEST['email'],$_REQUEST['edad']);
19     echo "Registro insertado exitosamente <br><br>";
20 }
21
22 echo "Registros en la coleccion usuarios: <br>";
23 $usrs->obtenerRegistros();
24 <?>
```



Lab201.php

24. Clase con código para conectar a mongo db a la base de datos usuarios, insertar registro en colección usuarios y leer los mismos

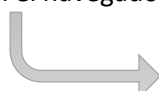
```
1 <?php
2
3 class UsuariosMDB{
4
5     private $mongo_client;
6     private $mongo_host;
7     private $mongo_user;
8     private $mongo_pswd;
9     private $mongo_database;
10    private $mongo_collection;
11    private $mongo_document;
12
13    function __construct(){
14        $this->mongo_host = "localhost:27017";
15        $this->mongo_user = "";
16        $this->mongo_pswd = "";
17        $this->mongo_database = "usuarios_api";
18        $this->mongo_collection = "usuarios";
19        $this->conectarMongodb();
20    }
21
22    public function conectarMongodb(){
23        try {
24            $this->mongo_client = new MongoClient("mongodb://" . $this->mongo_host);
25        }
26        catch (Exception $e) {
27            die("Ocurrio la siguiente excepcion al tratar de conectarse a la base de datos: " . $e->getMessage());
28        }
29    }
30
31    public function insertarRegistro($nombre,$apellido,$email,$edad){
32        try {
33            $this->mongo_document = new MongoDB\Driver\BulkWrite;
34
35            $doc = ['_id' => new MongoDB\BSON\ObjectID, 'nombre'=>"$nombre", 'apellido'=>"$apellido", 'email'=>"$email", 'edad'=>intval($edad)];
36            $this->mongo_document->insert($doc);
37
38            $this->mongo_client->executeBulkWrite($this->mongo_database.'.'.$this->mongo_collection, $this->mongo_document);
39        }
40        catch (MongoDB\Driver\Exception\Exception $e){
41            echo "Ocurrio la siguiente excepcion al tratar de guardar el registro: ", $e->getMessage(), "\n";
42        }
43    }
44
45    public function obtenerRegistros(){
46        try {
47            $query = new MongoDB\Driver\Query([]);
48            $rows = $this->mongo_client->executeQuery($this->mongo_database.'.'.$this->mongo_collection, $query);
49
50            foreach ($rows as $row) {
51                echo "ID: $row->_id - Nombre Completo: $row->nombre $row->apellido - E-mail: $row->email - Edad: $row->edad <br>";
52            }
53        }
54        catch (MongoDB\Driver\Exception\Exception $e) {
55            echo "Ocurrio la siguiente excepcion al tratar de leer los registros: ", $e->getMessage(), "\n";
56        }
57    }
58 <?>
```



UsuariosMDB.php

Parte VI – Pruebas de la Aplicación Web y la base de datos

25. Probar en el navegador



Nombre:

Apellido:

Email:

Edad:

26. Insertar datos

Nombre:
Apellido:
Email:
Edad:

27. Observar resultados en el formulario

Nombre:
Apellido:
Email:
Edad:

Registro insertado exitosamente

Registros en la coleccion usuarios:

ID: 636301b174b5751670000ea9 - Nombre Completo: Maria Perez - E-mail: maria.perez@elcorreo.com - Edad: 25

28. Actualizar datos en mongoDB

DOCUMENTS	0	TOTAL SIZE	0B	AVG. SIZE	0B
INDEXES	1	TOTAL SIZE	36.0KB	AVG. SIZE	36.0KB

Displaying documents 0 - 0 of N/A

29. Observar documento aparece en la colección usuarios de la base de datos usuarios_api

[usuarios_api.usuarios](#)

Documents Aggregations Schema Explain Plan Indexes Validation

{ field: 'value' }

```
_id: ObjectId("636301b174b5751670000ea9")
nombre: "Maria"
apellido: "Perez"
email: "maria.perez@elcorreo.com"
edad: 25
```

30. Insertar otros dos datos adicionales, observar resultados en Mongo DB

[usuarios_api.usuarios](#)

Documents Aggregations Schema Explain Plan Indexes Validation

{ field: 'value' }

```
_id: ObjectId("636301b174b5751670000ea9")
nombre: "Maria"
apellido: "Perez"
email: "maria.perez@elcorreo.com"
edad: 25
```

```
_id: ObjectId("6363025a74b5751670000eaa")
nombre: "Juan"
apellido: "Perez"
email: "juan.perez@elcorreo.com"
edad: 30
```

```
_id: ObjectId("6363028674b5751670000eab")
nombre: "Sandra"
apellido: "Sanchez"
email: "sandra@otrocorreo.com"
edad: 31
```