**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Desarrollo Backend con Node.JS y MongoDB. |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501123 - Construir sitios web según técnicas de interoperabilidad y protocolos técnicos. | RESULTADOS DE APRENDIZAJE | 220501123-01 - Elaborar el sistema de almacenamiento NoSQL de acuerdo con los requerimientos técnicos. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 01 |
| NOMBRE DEL COMPONENTE FORMATIVO | Gestión de bases de datos NoSQL con MongoDB |
| BREVE DESCRIPCIÓN | En este componente formativo, se aborda el tema de las bases de datos no relacionales, en el que se presentan sus diferentes tipos y la forma en la que sus características se acoplan más fácilmente a ciertos contextos. Adicionalmente, se introduce la plataforma MongoDB como ejemplo para la gestión de bases de datos NoSQL de tipo documental. |
| PALABRAS CLAVE | MongoDB, NoSQL, Bases, Datos, No relacionales |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - Ciencias naturales, aplicadas y relacionadas |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS**

**Introducción**

**1. Bases de datos** NoSQL

**2. MongoDB**

2.1 Documentos

2.2 Comandos básicos

2.3 Operadores

1. **INTRODUCCIÓN**

Cordial bienvenida a este componente formativo en el que trataremosla gestión de bases de datos NoSQL con MongoDB; para conocer la importancia del tema y los diferentes temas que se incluirán, le invitamos a ver el siguiente video:



1. **DESARROLLO DE CONTENIDOS**
2. **Bases de datos** NoSQL

El concepto de bases de datos NoSQL está ampliamente divulgado en el entorno del desarrollo de *software* para identificar aquellos sistemas de bases de datos que no son gestionados bajo el estándar de SQL y/o no se pueden clasificar dentro de las bases de datos relacionales.

Estas bases de datos NoSQL son una forma de solucionar los problemas de diseño de bases de datos que no podían ser resueltos mediante los sistemas de bases de datos relacionales o que, en caso de ser implementados (los sistemas de bases de datos relacionales), representaban un manejo de estructuras demasiado complejas que sacrificaban el rendimiento de las aplicaciones que hacían uso de estas. En este sentido, las bases de datos NoSQL permiten resolver problemas en contextos específicos.

Entonces, analice las características de los sistemas de datos relacionales (SQL) frente a los no relacionales (NoSQL):



Es importante tener en cuenta que, con la globalización de las aplicaciones, el uso masivo del Internet y las telecomunicaciones, la forma de procesar los datos se encuentra en constante evolución, lo que requiere de esquemas menos rígidos que permitan cambios de sus estructuras de una manera natural y que sean más fáciles de manejar y escalar a grandes volúmenes de datos; en estas nuevas circunstancias, muchos de los sistemas actuales necesitan tener un alto nivel de rendimiento y una alta escalabilidad, con necesidades de despliegue inmediato, por lo que se acoplan mejor a los sistemas de almacenamiento NoSQL.

El concepto NoSQL agrupa dentro de sí diferentes tipos de bases de datos, cada una de ellas con características muy diferentes y particulares, lo que brinda un abanico adicional de opciones que pueden adaptarse mejor a soluciones requeridas en diferentes entornos.

Según AWS (2021), las bases de datos NoSQL más representativas son las siguientes:

* **Bases de datos clave-valor**

Este tipo es una de las bases de datos NoSQL más simples que existen. Su estructura, generalmente, se representa como una tabla de dos columnas:



A continuación, se presenta un ejemplo de la estructura de este tipo de base de datos:

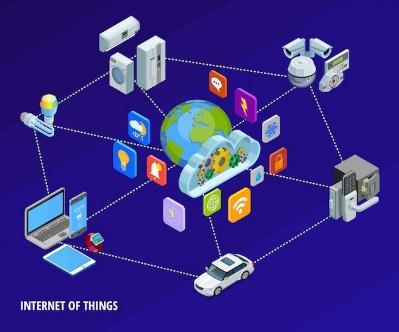
**Figura 1**

*Representación de una base de datos Clave-Valor*



Este tipo de estructura de almacenamiento es similar a otros tipos de estructura que se utilizan en programación para el procesamiento de colecciones en memoria temporal, como son los diccionarios y las tablas Hash.





.

En la siguiente tabla, se muestran algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura de almacenamiento clave-valor:

**Tabla 1**

*Ejemplos de gestores de bases de datos tipo clave-valor*

|  |  |
| --- | --- |
| **Nombre y logo** | **URL sitio oficial** |
|  | <https://redis.io> |
|  | <https://couchdb.apache.org> |
| Project Voldemort | <https://www.project-voldemort.com/voldemort/> |
|  | <https://aws.amazon.com/es/dynamodb/> |
|  | <https://riak.com> |
|  | <https://www.oracle.com/co/database/technologies/related/berkeleydb.html> |
|  | <https://aerospike.com> |

* **Bases de datos documentales**

El nombre documental hace referencia al tipo de estructura principal usada para representar la información, el cual se denomina documento.





Estas bases de datos semiestructuradas facilitan la versatilidad, por lo que pueden evolucionar según las necesidades de las aplicaciones, y, entre sus ventajas, ofrece la posibilidad de gestión de información semiestructurada, que puede albergar gran cantidad de información con diversos tipos de datos; ya que no tiene un esquema formal o rígido, simplifican las tareas de inserción y actualización de datos.

Las aplicaciones web y móviles actuales están sometidas a gran cantidad de cambios en las estructuras de información que gestionan, por lo cual este tipo de bases de datos NoSQL se adecúan muy bien a este tipo de contextos.

A continuación, se observa un ejemplo de una posible estructura para una base de datos tipo documental.

**Figura 2**

*Representación gráfica de bases de datos tipo documental*



De acuerdo con la información ya proporcionada, ¿dónde es posible encontrar este tipo de base de datos?



En la siguiente tabla 2, se muestran algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura de almacenamiento documental. Recuerde que los vistos en la anterior tabla fueros los de clave-valor:

**Tabla 2**

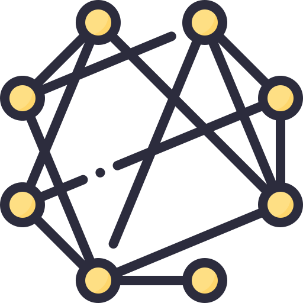
*Ejemplos de gestores de bases de datos tipo documental.*

|  |  |
| --- | --- |
| Nombre y logo | URL sitio oficial |
|  | <https://www.mongodb.com> |
|  | <https://ravendb.net> |
|  | <https://www.marklogic.com> |
|  | <https://azure.microsoft.com/es-es/services/cosmos-db/> |
|  | <https://orientdb.org> |

* **Bases de datos de tipo grafo**

Esta base de datos se caracteriza por representar la información por medio de vértices y aristas, lo que la asocia directamente con la teoría de grafos, ampliamente estudiada en la matemática y las ciencias de la computación.

Entonces, ¿qué se entiende por nodo o vértice?





Este tipo de base de datos carece de un esquema fijo y los nodos, a su vez, pueden tener diferentes tipos y cantidades de atributos, por lo que se dice que son **multidimensionales.**



Otra característica clave es que las relaciones pueden almacenar información adicional y no existen restricciones sobre la forma en la que se establecen dichas relaciones, es decir, pueden ser sin dirección, unidireccionales o bidireccionales. Esta característica le permite tener un altísimo rendimiento en la búsqueda de resultados y el descubrimiento de nueva información.

A continuación, un ejemplo gráfico de la representación de una base de datos de tipo grafo:

**Figura 3**

*Representación gráfica de bases de datos tipo grafo*



Nota. Tomado de Almagro (2017).

Este tipo de base de datos facilita la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos son los que incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento.

Ya vimos algunos de los sistemas gestores de bases de datos NoSQL más famosos en el mercado, que permiten el uso de la estructura tipo clave-valor y de almacenamiento documental; ahora, es momento de ver los de tipo grafo:

**Tabla 3**

*Ejemplos de gestores de bases de datos tipo grafo*

|  |  |
| --- | --- |
| **Nombre y logo** | **URL sitio oficial** |
|  | <https://neo4j.com/> |
|  | <https://www.tigergraph.com/> |
| Graph Engine | <https://www.graphengine.io/> |
|  | <https://www.arangodb.com/> |
| Titan Distributed Graph Database | <https://titan.thinkaurelius.com/> |

**Paralelo entre SQL y NoSQL**

Las bases de datos SQL y NoSQL cumplen con la labor de gestionar información y existe gran cantidad de sistemas gestores de bases de datos en el mercado, que facilitan este proceso con altos estándares de calidad. Sencillamente, estos dos modelos utilizan estructuras de almacenamiento y estándares de acceso a la información que son transversalmente diferentes. El uso o no de un modelo SQL o NoSQL dependerá del contexto de la solución que se busca resolver y de las necesidades de infraestructura requeridas por el cliente.

A continuación, se hace un paralelo de algunas de las características más importantes que diferencian un sistema SQL de un sistema NoSQL (AWS, 2021).



1. **MongoDB**

MongoDB es uno de los sistemas gestores de bases de datos NoSQL de tipo documental más famosos del mercado; su nombre viene del término en inglés *humongous,* que significa gigantesco. Ofrece un conjunto de soluciones, entre las que se incluye la posibilidad de trabajar con servidores que se pueden descargar y configurar en el equipo físico o servicios alojados en la nube por medio del gestor Atlas.

Este gestor de bases de datos NoSQL fue escrito en lenguaje C ++, pero todas las consultas usadas para la gestión de la información utilizan el formato BSON, que es similar al formato JSON, ampliamente utilizado por los desarrolladores web y móviles, para los procesos de transferencia de información a través de Internet.

Se presenta, entonces, la forma de hacer su instalación; luego, su configuración; y finalmente, su administración:

* **Instalación de MongoDB**

A continuación, en el video, se muestra la forma en la que se realiza una instalación exitosa de una instancia gratuita de MongoDB y el servicio de gestión *Cloud* *Atlas*, accediendo a la dirección [www.mongodb.com](http://www.mongodb.com)



* **Configuración de MongoDB**

Posteriormente, el sistema ingresará al *dashboard* de configuración general de MongoDB, Atlas, en el que se recomienda realizar las configuraciones iniciales de seguridad. En este espacio, se establecen las credenciales de acceso necesarias para la gestión de la base de datos. Para conocer cómo se realiza este proceso, le invitamos a ver el siguiente video:



* **Administración de MongoDB**

Una vez terminada la configuración básica de seguridad, podrá acceder al *dashboard* para la administración general de la base de datos NoSQL. Tenga en cuenta que este servicio de base de datos se encuentra administrado por el proveedor *Cloud* seleccionado y que la base de datos se alberga en un clúster. Para conocer cómo se realizan los pasos de administración, vea lo siguiente:



.

* 1. **Documentos**

Como ya se ha mencionado, MongoDB es un gestor de base de datos NoSQL, con una estructura basada en documentos. Por ello, es importante aclarar la jerarquía completa de esta estructuración de la información, la cual se muestra a través de la siguiente información:



Para un mejor entendimiento de esta jerarquía, la siguiente figura presenta gráficamente la relación de los elementos ya descritos:

**Figura 4**

*Jerarquías de almacenamiento en MongoDB*



Los documentos son abstracciones de una entidad u objeto del mundo real que expresa sus características por un conjunto de pares clave-valor. La clave o nombre del campo siempre son cadenas de texto y los valores pueden ser de cualquier tipo de dato primitivo o tipos de datos complejos.

A continuación, se presentan algunos ejemplos de documentos caracterizados por diferentes tipos de datos:



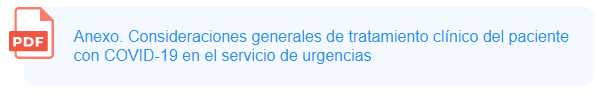
* 1. **Comandos básicos**

Los comandos básicos que se pueden utilizar para la manipulación y gestión de las bases de datos en MongoDB pueden ser ejecutados una vez se establece la conexión con el servidor, ya sea desde su cliente web o desde una aplicación externa.

A diferencia de las bases de datos relacionales, en MongoDB no existe una separación de comandos para la creación de estructuras como en el caso de las sentencias DDL de SQL y comandos para la manipulación de las estructuras creadas como las sentencias DML en SQL. Este tipo de base de datos busca tener mayor flexibilidad y adaptabilidad, por lo que una consulta sobre una estructura inexistente puede hacer que el motor de base de datos cree dicha estructura de forma automática.

Los comandos en MongoDB se pueden escribir tanto en una sola línea como ordenados por medio de espacios, saltos de línea o indentación, para hacer más legible su estructura, lo que no afecta la interpretación del motor de base de datos de MongoDB.

Le invitamos a conocer los comandos básicos más utilizados junto con su descripción:



* 1. **Operadores**

En las bases de datos relacionales, los comandos básicos pueden ampliarse para elaborar consultas mucho más complejas, además de restringir de una manera más adecuada. En MongoDB, estos elementos se procesan por medio de operadores, que se clasifican en cuatro tipos:

* **Operadores de comparación**

Los operadores de comparación, como su nombre lo indica, sirven para restringir las condiciones que deben ser evaluadas por el motor para determinar el conjunto de datos que puede ser afectado por el comando. Normalmente, son utilizados para ejecutar comandos de consulta o de manipulación de información en las colecciones.

En la siguiente tabla, se muestran los operadores de MongoDB, su equivalente en un ambiente relacional y su significado.

**Tabla 4**

*Operadores de comparación en MongoDB*

|  |  |  |
| --- | --- | --- |
| Operador MongoDB | Operador relacional equivalente | Significado |
| $eq | = | igual que (*equal*) |
| $ne | <> | distinto de (*not equal*) |
| $gt | > | valores mayores que (*greater than*) |
| $gte | >= | valores mayores o iguales que (*greater than or equal*) |
| $lt | < | valores menores que (*lower than*) |
| $lte | <= | valores menores o iguales que (*lower than or equal*) |
| $in | IN | sus valores se encuentran dentro del *array* especificado |
| $nin | NOT IN | operación contraria a $in, obtiene los documentos cuyo valor NO estén en el *array* |

* **Operadores de elemento**

Estos operadores existen para identificar si una clave/campo se define dentro de su estructura y valida el tipo de dato asociado a un campo particular, esto teniendo en cuenta que las estructuras en MongoDB se construyen y evolucionan de forma dinámica, ya que no tienen un esquema rígido de almacenamiento.

MongoDB utiliza el operador **$exists** para determinar la existencia de un campo dentro de la definición de un documento en una colección y el operador **$type** para comprobar el tipo de dato asociado a un campo particular;

Observe algunos ejemplos del uso de este tipo de operadores:



* **Operadores lógicos**

Estos operadores permiten crear condiciones más complejas en el proceso de manipulación de datos en colecciones de una base de datos MongoDB; funcionan y se interpretan igual que los operadores lógicos de cualquier lenguaje de programación y siguen la lógica proposicional con todas sus tablas de verdad.

Observe, en la siguiente tabla, algunos de los operadores lógicos más utilizados y su equivalente:

**Tabla 5**

*Operadores lógicos en MongoDB*

|  |  |  |
| --- | --- | --- |
| Operador MongoDB | Operador lógico equivalente | Ejemplo |
| $or | OR | Filtra de los documentos de la colección productos que cumplan alguna de las condiciones del *array*  db.productos.find({ $or: [ {"id":{$exists:"true"}},{"valor":{$gt:10}}] }).pretty()  En este caso, las dos condiciones a evaluar son que el campo id exista {"id":{$exists:"true"} o que el campo valor sea mayor que 10 {"valor":{$gt:10}} |
| $and | AND | Similar a $or, pero, en este caso, los documentos deben cumplir todas las condiciones del *array*  db.productos.find({$and:[ {"id":{$exists:"true"}},{"valor":{$gt:10}}]}).pretty()  Su uso puede resultar superfluo, teniendo en cuenta que en un filtro normal las condiciones indicadas se juntan mediante operadores lógicos Y  db.productos.find({"id":{$exists:"true"},"valor":{$gt:10}}).pretty() |

* **Operadores sobre *arrays***

Como los documentos pueden tener valores complejos que pueden ser de tipo lista, de datos primitivos o de listas de otros tipos de documentos, es muy útil usar operadores que permitan validar la información dentro de estas listas de elementos. De ahí la importancia de los operadores de *arrays* que facilitan esta labor.

A continuación, conozca algunos de los más comunes:



**C.** **SÍNTESIS**

En la actualidad, se presenta un universo muy amplio en variedad de tecnología para bases de datos, como es el caso de NoSQL que se creó con la finalidad de dar respuesta a las necesidades de desarrollo detectadas asociadas a estas aplicaciones actuales; en este sentido el presente componente formativo tiene como objetivo el abordaje de temas como las bases de datos no relacionales y la plataforma MongoDB. Estos temas se resumen a través del siguiente mapa conceptual, esto es:



**D. ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Conceptos clave relacionados con bases de datos NoSQL |
| Objetivo de la actividad | Afianzar algunos de los conceptos más importantes asociados a las bases de datos NoSQL para determinar la pertinencia de cada tipo, de acuerdo con los requerimientos del cliente. |
| Tipo de actividad sugerida | Arrastrar y soltar el término con la definición que corresponde |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 1 |

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Repaso de comandos básicos. |
| Objetivo de la actividad | Apropiar los comandos MongoDB básicos para el manejo adecuado de esta base de datos, teniendo en cuenta los requerimientos técnicos. |
| Tipo de actividad sugerida | Arrastrar y soltar el requerimiento con el comando que le corresponde. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 2 |

**E. MATERIAL COMPLEMENTARIO**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| MongoDB | MongoDB. (s. f.). *Welcome to the MongoDB Documentation*. | Página web | <https://www.mongodb.com/docs/> |

**F.** **GLOSARIO**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Clúster: | agrupación de varios equipos de cómputo especializados para ofrecer servicios. |
| Escalamiento horizontal: | permite el crecimiento de una manera muy amplia y técnicamente sin límites. Implica infraestructura de varios servidores, conocidos como nodos, que trabajan como una única unidad y pueden equilibrar cargas y agregar más componentes a su estructura, robusteciendo su capacidad sin afectar sus servicios. |
| Infraestructura tecnológica: | equipos de cómputo, sistemas de red y cualquier otro componente de tipo *hardware* requerido para el montaje y puesta en marcha de un sistema software. |
| *Shell*: | ventana de comandos. |

**H.** **REFERENCIAS BIBLIOGRÁFICAS**

Almagro-Blanco, P. y Sancho-Caparrini, F. (2017). *Generalized Graph Pattern Matching*. Cornell University. <https://arxiv.org/abs/1708.03734>

Amazon Web Services. (s. f.). *¿Qué es NoSQL?* AWS. [https://aws.amazon.com/es/NoSQL/](https://aws.amazon.com/es/nosql/)

Graph Everywhere. (2021). *Las top 10 Bases de datos nativas de grafos*. <https://www.grapheverywhere.com/las-top-10-bases-de-datos-nativas-de-grafos/>

**I. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha |
| Autor(es) | Jonathan Guerrero Astaiza | Experto Temático | Regional Cauca – Centro de Teleinformática y Producción Industrial | Mayo de 2021 |
| María Fernanda Chacón Castro | Diseñadora Instruccional | Regional Distrito Capital - Centro de Gestión Industrial | Mayo de 2022 |
| Andrés Felipe Velandia Espitia | Asesor Metodológico | Regional Distrito Capital - Centro de Diseño y Metrología | Mayo de 2022 |
| Rafael Neftalí Lizcano Reyes. | Responsable Equipo Desarrollo Curricular | Regional Santander - Centro Industrial del Diseño y la Manufactura | Mayo de 2022 |
| Darío González | Corrector de Estilo | Regional Distrito Capital - Centro de Diseño y Metrología | Agosto de 2022 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del cambio |
| Autor(es) |  |  |  |  |  |