**UNIDAD 1: MongoDB**

1. [UNIDAD 1: Introducción](https://learn.nextu.com/mod/page/view.php?id=10280&pid=P_WEB_DATABASE)

**Introducción**

**MongoDB**

MongoDB es un sistema manejador de base de datos NoSQL orientado a documentos, donde cada base de datos consiste de colecciones y las colecciones tienen documentos. En esta unidad comenzamos introduciendo las funciones para la creación de bases de datos MongoDB y las operaciones para el respaldo y recuperación de las bases de datos, así como también los mecanismos de importación y exportación de datos. Luego de ello, aprenderás a realizar consultas de mediana complejidad, con la finalidad de que explores el contenido de los documentos en las bases de datos. Uno de los conceptos más completos en MongoDB es el de agregación, que permite agrupar datos usando diversas opciones. De esta forma puedes generar desde vistas sencillas de tus datos hasta vistas más complejas resultantes del procesamiento de los datos usando las diversas funciones proporcionadas por el manejador. Finalmente, podrás definir usuarios y sus roles en tu base de datos a través de los mecanismos de autenticación soportados por MongoDB.

**Objetivos de aprendizaje**

1. Crear bases de datos NoSQL orientadas a documentos usando MongoDB que puedan ser respaldadas y recuperadas posteriormente.
2. Aplicar las consultas de agregación usando diversos operadores.
3. Implementar consultas de mediana complejidad sobre las bases de datos MongoDB con la finalidad de visualizar los datos de diversas formas.
4. Aplicar las funciones de autenticación de MongoDB para crear usuarios y sus correspondientes roles en el sistema, con la finalidad de validar los usuarios que pueden acceder las diferentes bases de datos.
5. Lección 1: Retomando MongoDB
   1. [Creando y consultando datos](https://learn.nextu.com/mod/lesson/view.php?id=10281&pid=P_WEB_DATABASE)

Comandos:

use database

show collections

db.collection.find()

db.collection.insert( 🡪 Se puede presionar tab para ver opciones

Ejemplo de insert:

db.tareas.insert ( [ { nombre: “Tarea2”},{ nombre: “Tarea3”}, detalle “dt” ])

Find con condición:

db.tareas.find({ detalle : “dt” })

db.tareas.find({ detalle : “dt”, nombre: “Tarea3” }) 🡪 es condición AND

Para hacer condición OR

db.tareas.find({ $or: [{ nombre: “Tarea2” },{ “Tarea3” }] })

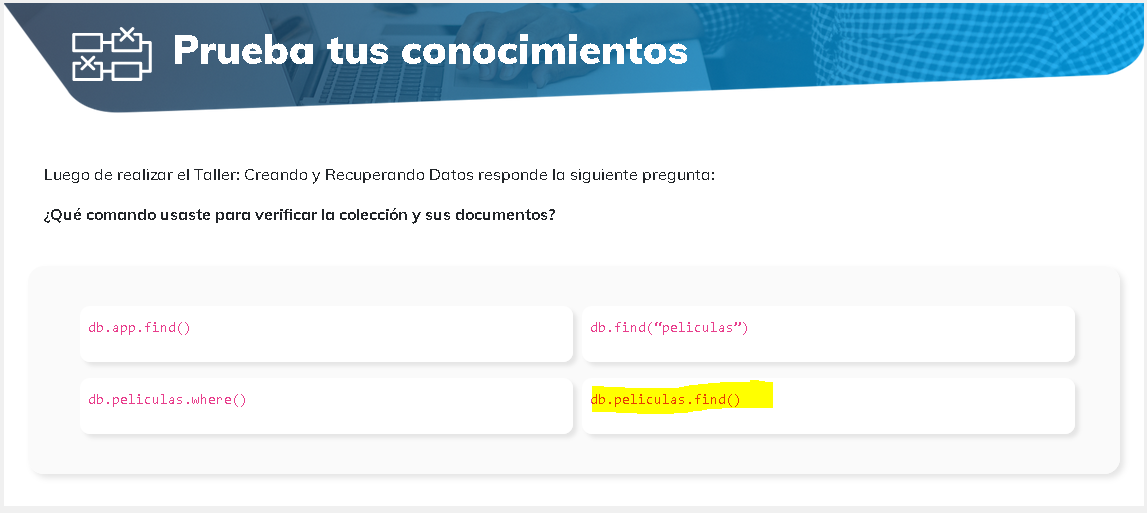
Otro ejemplo buscando documentos con valor fecha\_limite nulo:

db.tareas.find({ fecha\_limite: null })

Para eliminar documentos:

db.tareas.deleteMany ( { fecha\_limite: null } )

* 1. Taller 1
  2. Actividad Interactiva 1



* 1. Respaldando y recuperando datos

**Respaldo:**

mongodump (Respaldo todas las DBs)

mongodump -d productividad

Genera una carpeta que se llama dump 🡪 y se crea metadata y bson (binary son) 🡪 Datos en binario

**Recuperacion:**

mongorestore

**Importacion:**

Json archive

Con MongoDB Compass 🡪 click en collection 🡪 Import Data

Formato JSON o CSV

Y se selecciona el archivo

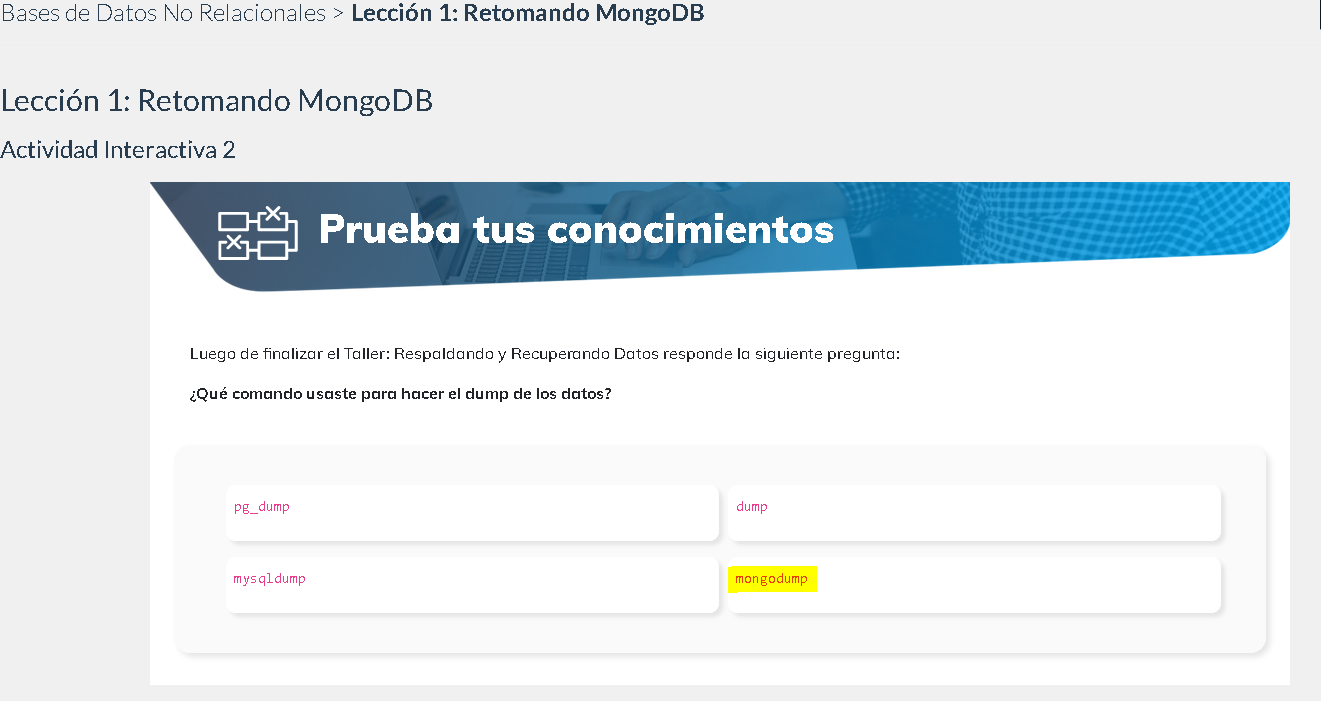
**Exportación:**

Con MongoDB Compass 🡪 click en collection 🡪 Export Data

Formato JSON o CSV

Seleccionar el nombre de archivo destino

* 1. Taller 2
  2. Actividad Interactiva 2



* 1. Ejercicio Práctico
  2. ¿Sabías qué?

**Cápsula de conocimiento**

**Resumen de comandos en MongoDB: Consultas Respaldo y Recuperación**

Este recurso te ofrece una guía rápida de los comandos en Mongo que hemos trabajado en esta lección.

* **Insertar un documento:**

> db.<coleccion>.insert({})

* **Consultas OR de dos condiciones:**

> db.<coleccion>.find({$or: [ , , ...]})

* **Consultar documentos sin un atributo:**

> db.<coleccion>.find({ : null })

* **Eliminar muchos:** Donde se eliminarán todos los documentos con el patrón indicado por {}.

> db.<coleccion>.deleteMany({})

* **Respaldo:** La operación de respaldo crea una carpeta llamada `dump` desde donde se ejecuta.

terminal> mongodump -d <base\_de\_datos>

* **Recuperación:** La operación de recuperación se realiza usando los archivos de la carpeta llamada `dump`.

terminal> mongorestore

Esperamos que este resumen sea de utilidad para ti en el futuro cercano. Recuerda que estas no son las únicas formas de utilizar estos comandos, existen otras, sin embargo, son las más comunes.

1. Lección 2: Consultas
   1. [Consultando y ordenando atributos](https://learn.nextu.com/mod/lesson/view.php?id=10282&pid=P_WEB_DATABASE)

Db.collection.find().pretty() 🡪 Una forma más ordenada

Db.collection.find().sort({ nombre: 1 }).pretty() 🡪 Ordernar, 1 ordernar de forma ascendente, -1 ordenar de forma descendente, 0 no ordenar

Db.collection.find().sort({ nombre: 1 }).limit(2).pretty() 🡪 Solo 2 documentos

Db.collection.find( {}, { nombre: 1 } ) 🡪 1 significa que se muestre el campo, y 0 que no.

Db.collection.find( {}, { nombre: 1 , \_id: 0 } ) 🡪 Para desactivar el \_id

Db.collection.find({ nombre: “AlgunNombre” }) 🡪 Busca los documentos con nombre AlgunNombre

Db.collection.find({ nombre: { $ne: “AlgunNombre” } }) 🡪 Busca los documentos que no tenga nombre AlgunNombre

* 1. Taller 1
  2. Actividad Interactiva
  3. Consultando textos
  4. Taller 2
  5. Ejercicio Práctico
  6. ¿Sabías qué?

1. Lección 3: Agrupando los datos
2. Lección 4: Aplicando Seguridad y Respaldando Datos
3. UNIDAD 1: Prueba