**Actividad:**

**creación de los objetos de la base de datos**

**GA6-220501096-AA1-EV04**

**Aprendiz:**

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

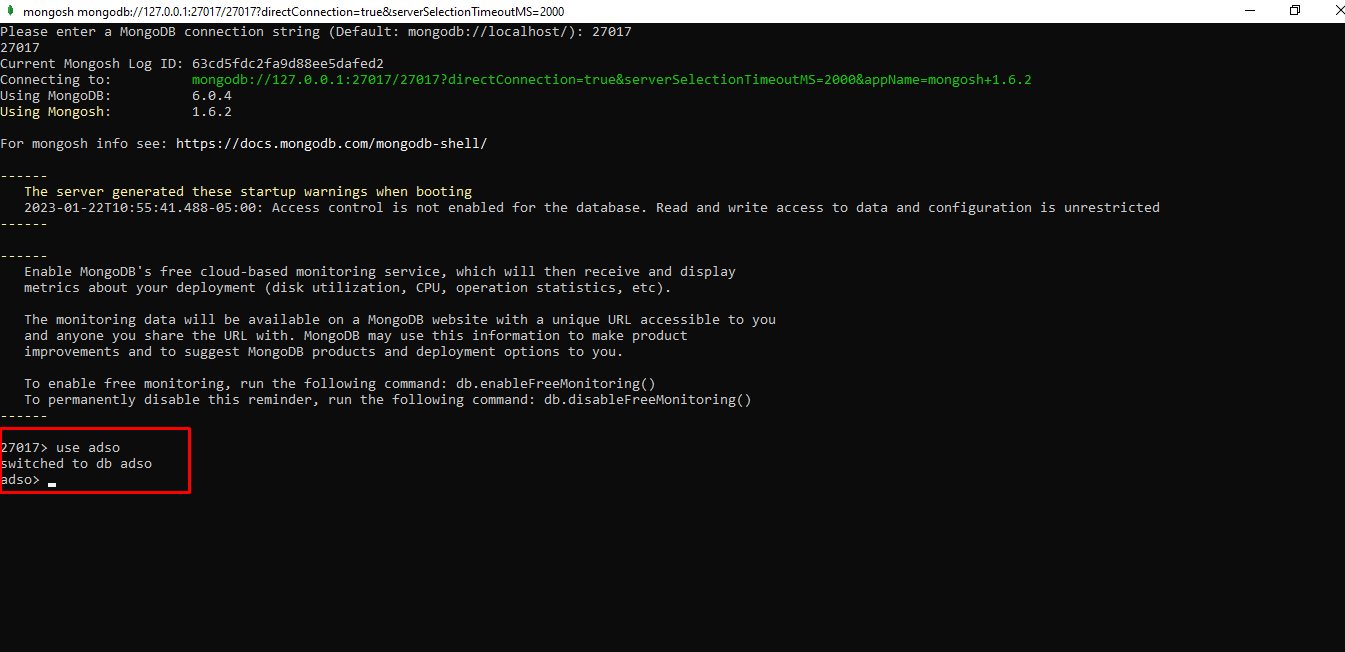
Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Con base en las características del *software* a desarrollar realice el modelo relacional indicando la cardinalidad y cumpliendo con las reglas de normalización según los conceptos y ejemplos vistos en el componente Modelo entidad relación – Modelo relacional.

● Cree una base de datos NoSQL.

**RTA**: Para crear una base de datos en MongoDB llamada ADSO, puedes usar el comando **use** en la consola de MongoDB. Por ejemplo:

 Logotipo

Descripción generada automáticamente con confianza media

Si la base de datos ya existe, esto simplemente cambiará a esa base de datos. Si no existe, se creará automáticamente al agregar una colección o al insertar un documento en ella.

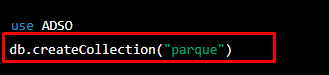
Si prefieres crear la base de datos antes de usarla, puedes crearla usando el comando **db.createDatabase()** en la consola de MongoDB:



Es importante tener en cuenta que, para MongoDB 3.4 y versiones anteriores, el comando **db.createDatabase()** no está disponible y debería usar **use** y **db.createCollection()** en su lugar.

* Cree una colección de datos llamada “parque”.

RTA: Para crear una colección de datos llamada "parque" en MongoDB, puedes usar el comando **db.createCollection()**. Por ejemplo: **db.createCollection('parque')**



Con el primer comando estamos seleccionando la base de datos y con el segundo creamos la colección "parque" dentro de la base de datos seleccionada.

Texto

Descripción generada automáticamente

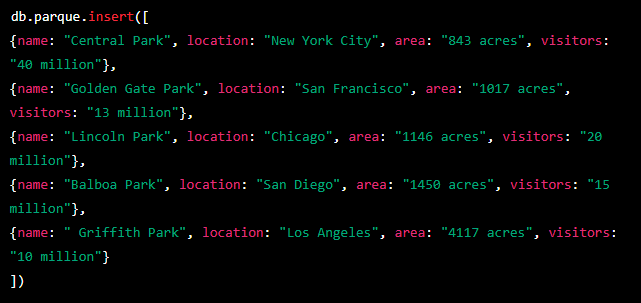
* Inserte cinco (5) documentos con la estructura JSON Creada.

**RTA:**Para insertar cinco documentos con una estructura JSON específica en la colección "parque" en MongoDB, puedes usar el comando db.collection.insert(). Por ejemplo, si tu estructura JSON es la siguiente:

Texto

Descripción generada automáticamente

Podemos insertar 5 documentos con la siguiente sintaxis:



Después de insertarse podemos validar la siguiente información

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

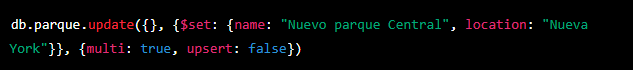
Cada uno de los documentos creados tiene las propiedades "name", "location", "area", "visitors" con sus respectivos valores.

Cabe mencionar que el comando **insert()** inserta uno o varios documentos en una colección, especificando el conjunto de documentos con una sintaxis similar a la de un objeto JSON.

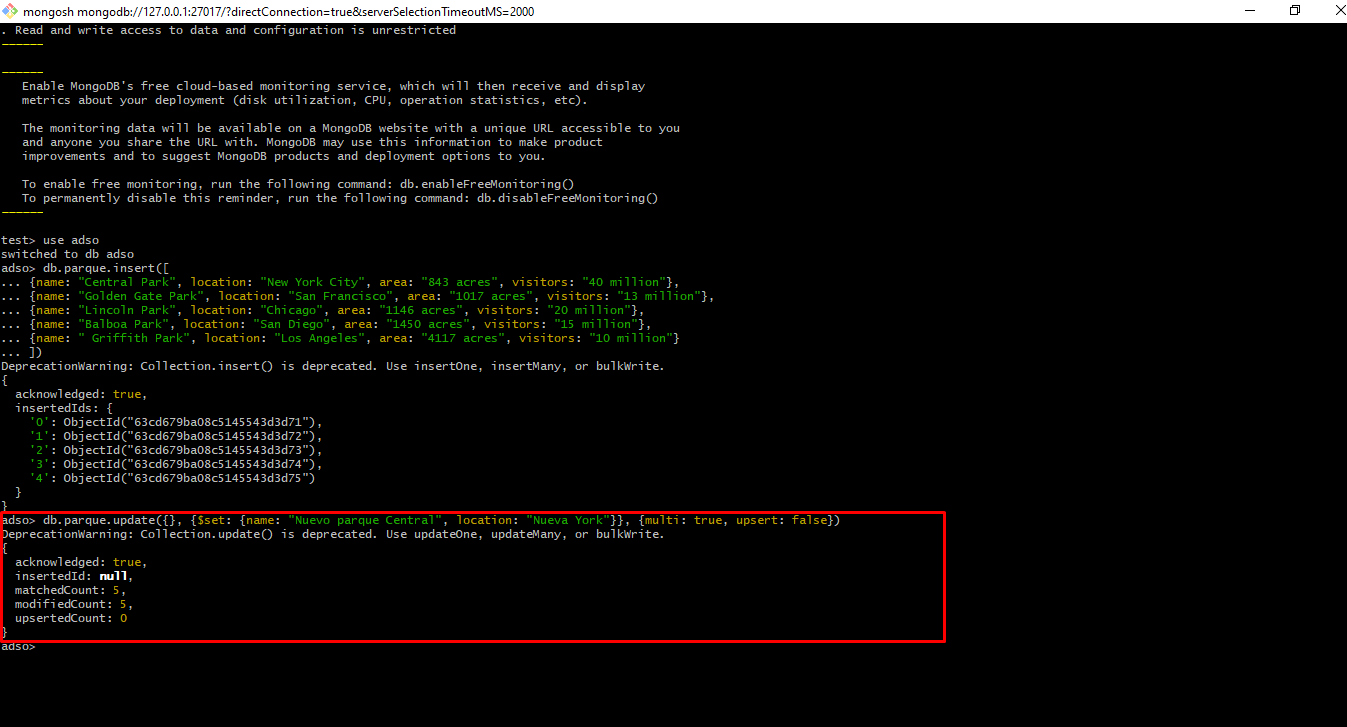
* Actualizar los datos del primer y último registro.

RTA: Para actualizar el primer registro, puedes usar el comando "db.parque.updateOne({}, {$set: {datos\_nuevos}})"

ejemplo:



Y esta acción la podemos validar de la siguiente manera



Este comando actualizará el primer registro de la colección "parque" estableciendo el nombre del parque a "Nuevo parque Central" y la ubicación a "Nueva York". el primer argumento es una consulta vacía, es decir selecciona todos los documentos, el segundo es el nuevo valor de los campos que queremos cambiar y el tercer argumento es una opción para cambiar todos los documentos.

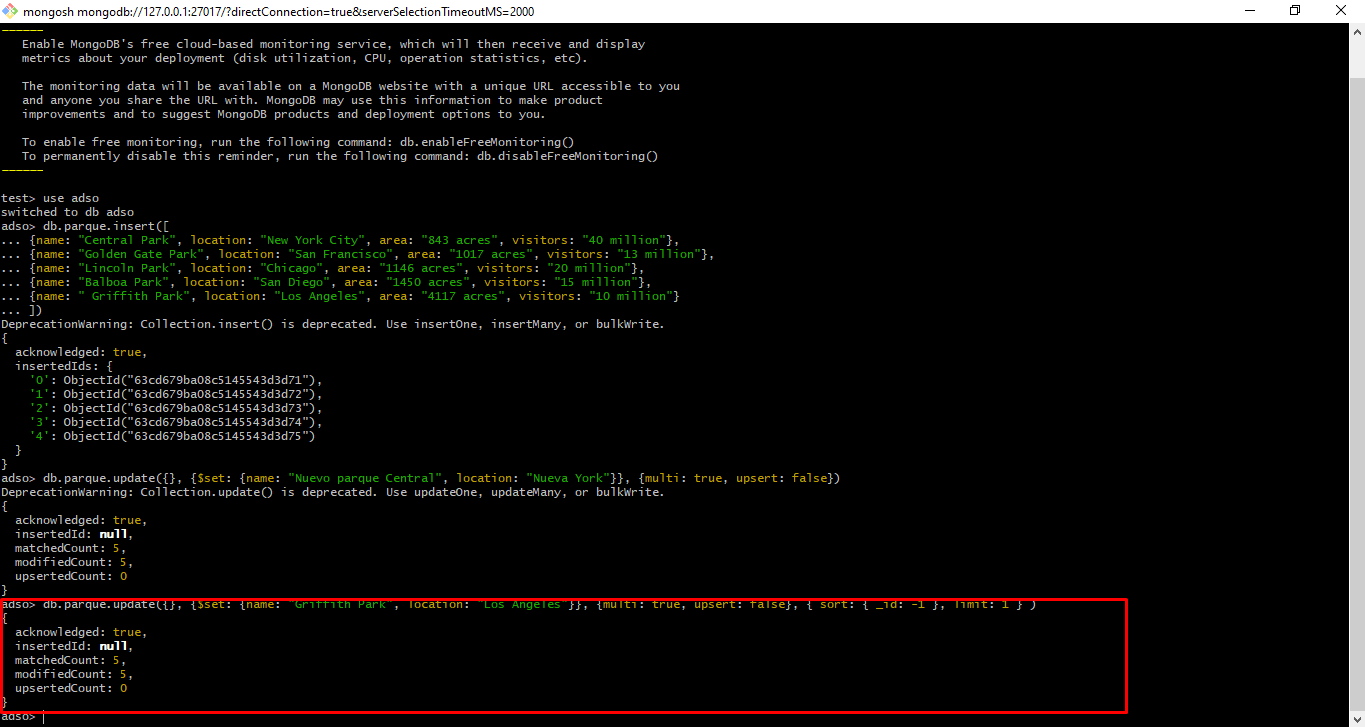
Para actualizar el último registro, puedes usar el comando "db.parque.updateOne({}, {$set: {datos\_nuevos}}, {sort: {\_id: -1}, limit: 1})"

ejemplo:

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

Y esta acción la podemos visualizar de la siguiente manera



* Liste la colección completa.

**RTA:**Para listar todos los documentos en una colección en MongoDB, puedes usar el comando db.collection.find() sin ningún argumento. Por ejemplo, para listar todos los documentos en la colección "parque":



Y se visualiza de la siguiente manera



También puedes utilizar el comando **db.collection.find().pretty()** para mostrar los resultados en un formato más legible:



En ambos casos, se mostrarán todos los documentos en la colección "parque", con todos sus campos y valores.

Es importante mencionar que el comando **find()** devuelve un cursor, el cual es un objeto especial que permite iterar sobre los resultados de una consulta. si deseas mostrar todos los resultados del cursor en una sola vez, puedes utilizar el método **toArray()** sobre el cursor.



También puedes limitar los resultados utilizando el método limit().



Este comando devolverá solo los primeros 3 documentos de la colección.

* Borre el tercer documento de la colección parque.

RTA: Para eliminar un documento específico en una colección en MongoDB, puedes usar el comando **db.collection.remove()** y especificar un criterio para identificar el documento a eliminar. Por ejemplo, para eliminar el tercer documento de la colección "parque", puedes usar la siguiente sintaxis:



Este comando eliminará el tercer documento en la colección "parque". El primer argumento es una consulta vacía, es decir selecciona todos los documentos, el segundo argumento indica que solo queremos eliminar un solo documento y el tercer argumento especifica que queremos seleccionar el tercer documento ordenado por \_id y saltando los primeros dos documentos.

Es importante mencionar que el comando **remove()** elimina de forma permanente los documentos y no tiene opción de deshacer la acción. Si quieres mantener un registro de los documentos eliminados, puedes considerar utilizar el comando **update()** y agregar un campo "eliminado" con el valor "si" o similar.

Interfaz de usuario gráfica, Texto

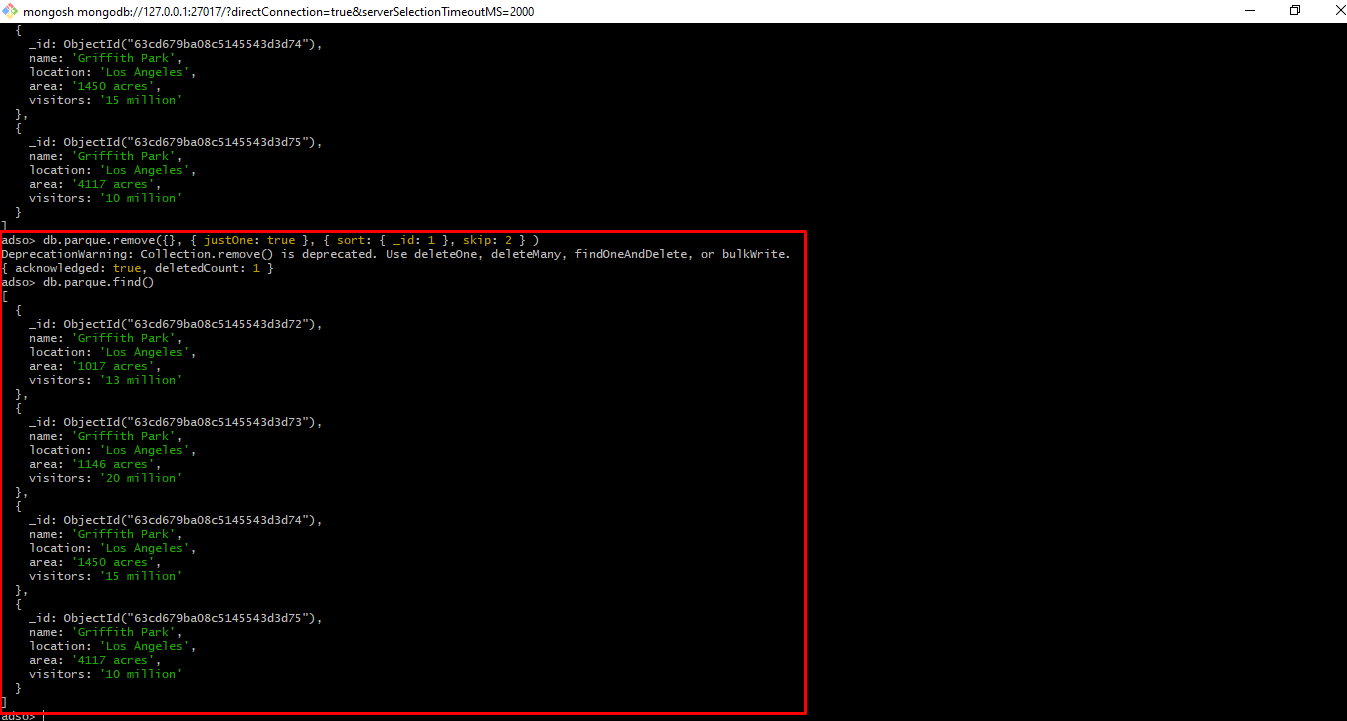
Descripción generada automáticamente

* Liste la colección de datos completa.

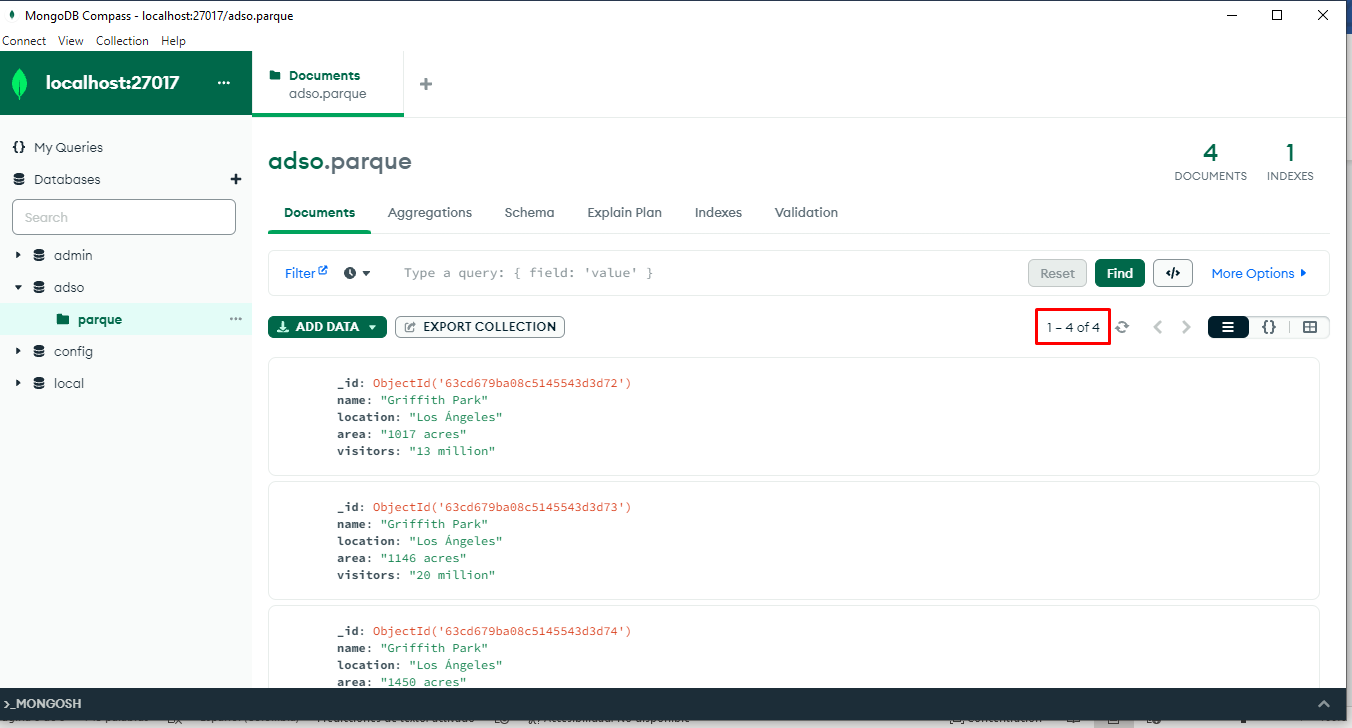
**RTA:** Para listar todos los documentos en una colección en MongoDB, puedes usar el comando db.collection.find() sin ningún argumento. Por ejemplo, para listar todos los documentos en la colección "parque":



Y al ejecutar el comando el resultado es el siguiente



Al validar el proceso por Mongo DB Compass podemos visualizar que el proceso se realizo con éxito



También puedes utilizar el comando **db.collection.find().pretty()** para mostrar los resultados en un formato más legible:



En ambos casos, se mostrarán todos los documentos en la colección "parque", con todos sus campos y valores.

Es importante mencionar que el comando **find()** devuelve un cursor, el cual es un objeto especial que permite iterar sobre los resultados de una consulta. si deseas mostrar todos los resultados del cursor en una sola vez, puedes utilizar el método **toArray()** sobre el cursor.

Final del formulario

También puedes limitar los resultados utilizando el método limit().



Este comando devolverá solo los primeros 3 documentos de la colección.

Cabe mencionar que si no deseas mostrar todos los campos de cada documento, puedes especificar los campos específicos que deseas mostrar en el comando **find()** utilizando el parametro **projection** y especificando los campos que deseas mostrar.



Este comando devolverá solo los campos "name" y "location" de todos los documentos en la colección "parque" sin mostrar el campo "\_id".