CONSULTAS

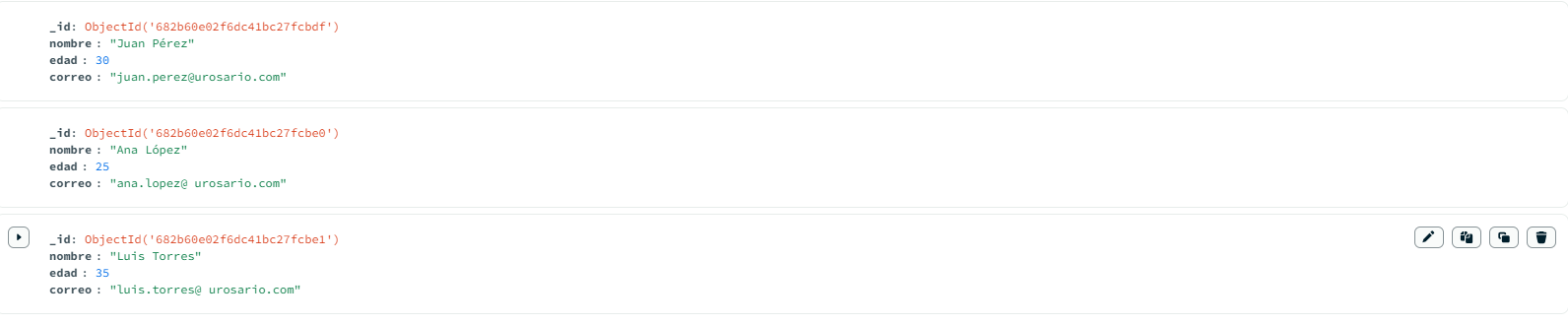
--------------------------------------------------------------------------------------------------------------------

• Encuentra todos los usuarios.

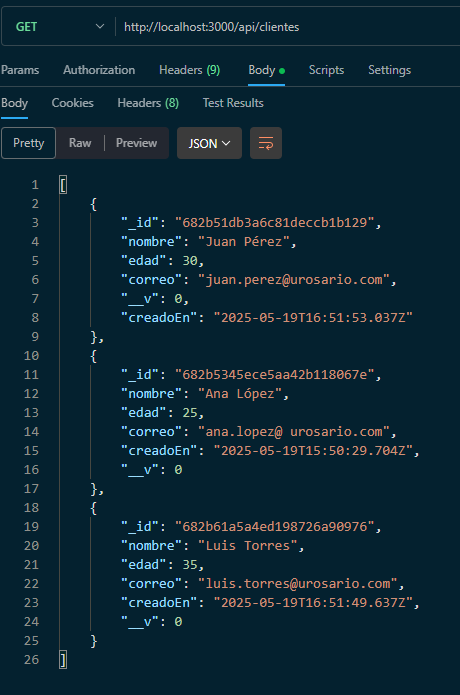
db.usuarios.find()



Abriendo la interfaz grafica tenemos todos los usuarios.



Consulta PostMan: http://localhost:3000/api/clientes con un GET

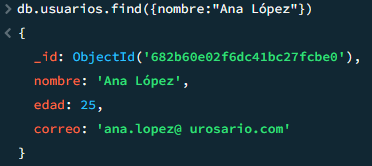


--------------------------------------------------------------------------------------------------------------------

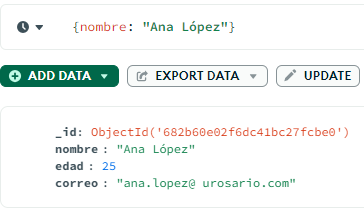
• Encuentra el usuario con nombre Ana López.

--------------------------------------------------------------------------------------------------------------------

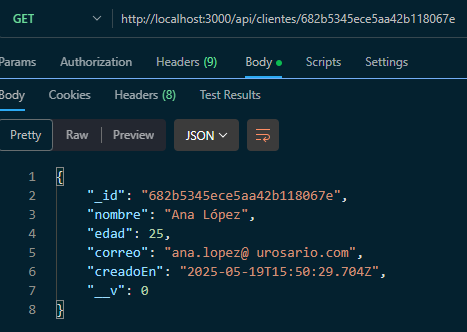
db.usuarios.find({ nombre: "Ana López" })



{nombre: "Ana López" }



<http://localhost:3000/api/clientes/682b5345ece5aa42b118067e>

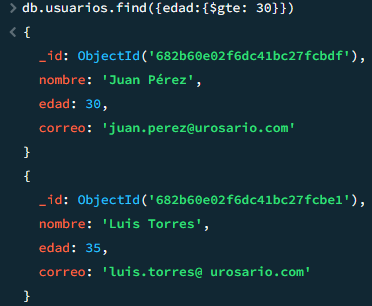


--------------------------------------------------------------------------------------------------------------------

• Encuentra todos los usuarios mayores o iguales a 30 años.

--------------------------------------------------------------------------------------------------------------------

db.usuarios.find({ edad: {$gte:30} })



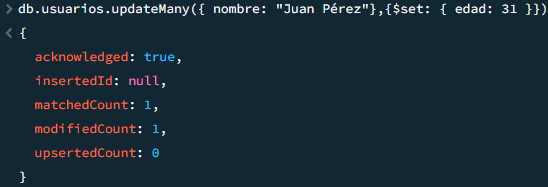
{ edad: {$gte:30} }



ACTUALIZAR

--------------------------------------------------------------------------------------------------------------------

db.usuarios.updateMany({ nombre: "Juan Pérez"},{$set: { edad: 31 }})

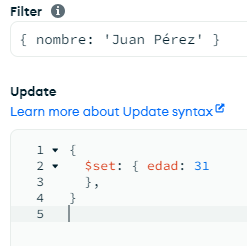


{

$set: { edad: 31

},

}



METODO PUT

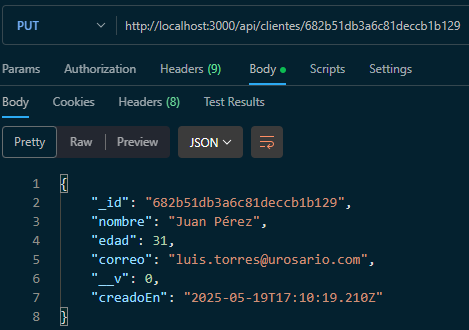
{

"nombre" : "Juan Pérez",

"edad": 31,

"correo" : "luis.torres@urosario.com"

}



Añadir campo

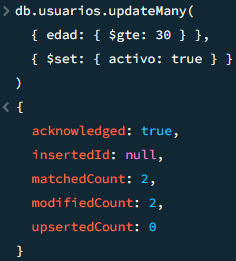
--------------------------------------------------------------------------------------------------------------------

db.usuarios.updateMany(

{ edad: { $gte: 30 } },

{ $set: { activo: true } }

)

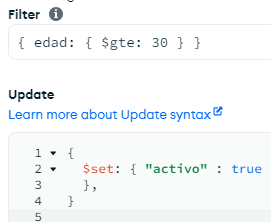


{

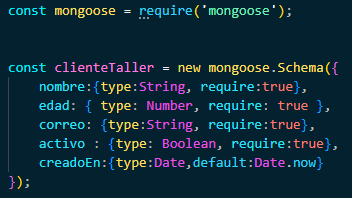
$set: { "activo" : true

},

}



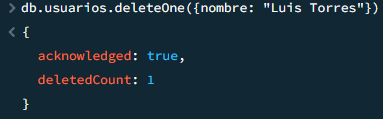
SE EDITA PRINCIPALMENTE DESDE LOS MODELOS



ELIMINACION

--------------------------------------------------------------------------------------------------------------------

db.usuarios.deleteOne({nombre: "Luis Torres"})



{"nombre": "Luis Torres"}

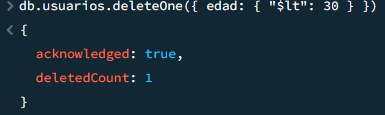


DELETE http://localhost:3000/api/clientes/682b5362ece5aa42b1180680

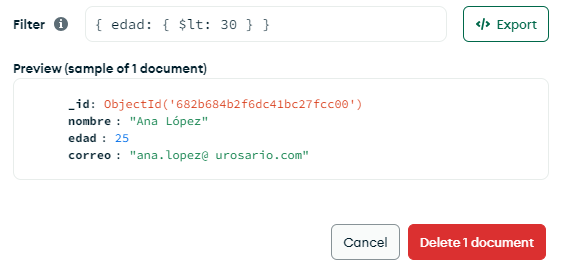


--------------------------------------------------------------------------------------------------------------------

db.usuarios.deleteOne({edad: { "$lt": 30 } })



{edad: {"$lt”: 30}} Esto va en la interfaz y posterior a esto se hacia el delete.



Parte 2: Gestión de Productos

--------------------------------------------------------------------------------------------------------------------

1. Crea una nueva colección llamada productos e inserta al menos 5 productos con los

siguientes campos:

db.createCollection("productos")



const mongoose = require('mongoose');

const productos = new mongoose.Schema({

nombre:{type:String, require:true},

precio: { type: Number, required: true },

categoria:{type:String,require: true}

});

module.exports=mongoose.model("productos", productos)

const mongoose = require('mongoose');

const clienteTaller = new mongoose.Schema({

nombre:{type:String, require:true},

edad: { type: Number, require: true },

correo: {type:String, require:true},

creadoEn:{type:Date,default:Date.now}

});

module.exports=mongoose.model("Cliente", clienteTaller)



INSERCION DE DATOS

[{

"\_id": {

"$oid": "682a7adc64ee13d0c2283970"

},

"nombre": "Televisor",

"precio": 150,

"categoria": "Electrónica"

},

{

"\_id": {

"$oid": "682a7adc64ee13d0c2283971"

},

"nombre": "Celular",

"precio": 200,

"categoria": "Electrónica"

},

{

"\_id": {

"$oid": "682a7adc64ee13d0c2283972"

},

"nombre": "Camiseta Roja",

"precio": 20,

"categoria": "Ropa"

},

{

"\_id": {

"$oid": "682a7adc64ee13d0c2283973"

},

"nombre": "Camiseta Negra",

"precio": 20,

"categoria": "Ropa"

},

{

"\_id": {

"$oid": "682a7adc64ee13d0c2283974"

},

"nombre": "Nevera",

"precio": 600,

"categoria": "Hogar"

},

{

"\_id": {

"$oid": "682a7adc64ee13d0c2283975"

},

"nombre": "Sofa",

"precio": 200,

"categoria": "Hogar"

}]



POSTMAN

{

"nombre" : "Televisor",

"precio": 150,

"categoria": "Electronica"

}

{

"nombre" : "Celular",

"precio": 200,

"categoria": "Electronica"

}

{

"nombre" : "Camiseta Roja",

"precio": 20,

"categoria": "Ropa"

}

{

"nombre" : "Camiseta Negra",

"precio": 20,

"categoria": "Ropa"

}

{

"nombre" : "Nevera",

"precio": 600,

"categoria": "Hogar"

}

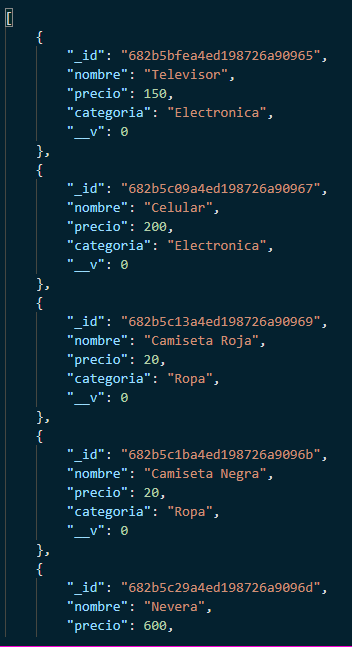
{

"nombre" : "Sofa",

"precio": 200,

"categoria": "Hogar"

}

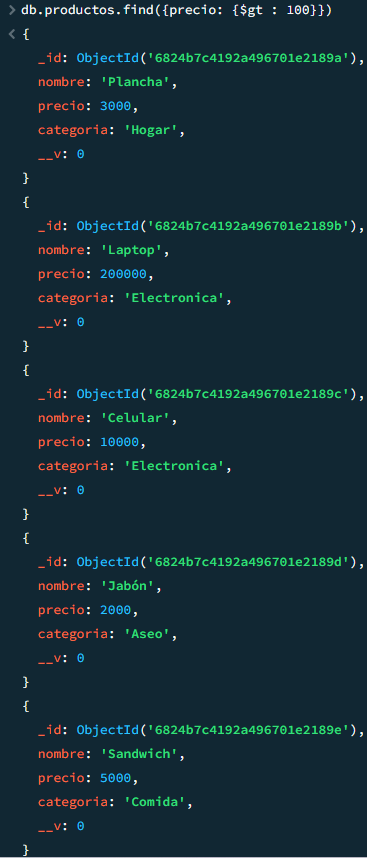


CONSULTAS

-------------------------------------------------------------------------------------------

• Encuentra todos los productos con precio mayor a $100.

db.productos.find({precio: {$gt : 100}})



{precio: {$gt: 100}}



• Ordena los productos por precio de manera descendente.

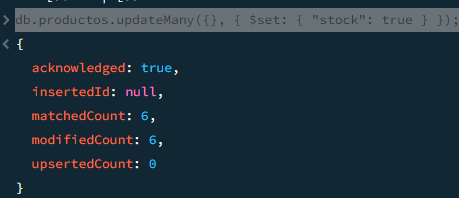
db.productos.find().sort({ 'precio': -1 });



3. Actualiza los datos:

-------------------------------------------------------------------------------------------

db.productos.updateMany({}, { $set: { "stock": true } });

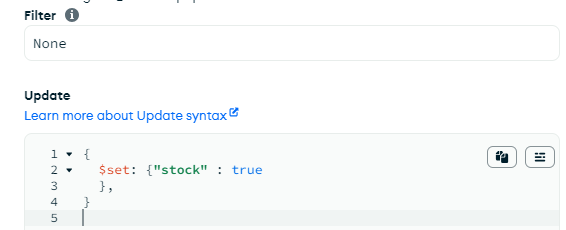


{

$set: {"stock" : true

},

}



CREAR DESDE LOS MODELOS, se cree un nuevo campo.

4. Elimina Registros

-------------------------------------------------------------------------------------------

• Elimina todos los productos cuyo precio sea menor a $50.

db.productos.delete({ precio: { $lt: 50 }})

{precio: { $lt: 50 }}

-------------------------------------------------------------------------------------------

PARTE 3: AGREEGACION Y ANALISIS

-------------------------------------------------------------------------------------------

1. Realiza una agregación para calcular:

• El precio promedio de los productos agrupados por categoría.

db.productos.aggregate([

{

$group: {

\_id: "$categoria",

precioPromedio: { $avg: "$precio" }

}

}

])

{

\_id: "$categoria",

precioPromedio: { $avg: "$precio" }

}

const Producto = require('./models/productos');

const resultado = await Producto.aggregate([

{

$group: {

\_id: "$categoria",

precioPromedio: { $avg: "$precio" }

}

}

]);

console.log(resultado);

-------------------------------------------------------------------------------------------

Punto 2

-------------------------------------------------------------------------------------------

db.productos.aggregate([

{

$group: {

\_id: "$categoria",

precioPromedio: { $avg: "$precio" }

}

},

{ $sort: { precioPromedio: -1 } },

{ $limit: 1 }

])

{ precioPromedio: -1 }

{ $limit: 1 }

const resultado = await Producto.aggregate([

{

$group: {

\_id: "$categoria",

precioPromedio: { $avg: "$precio" }

}

},

{ $sort: { precioPromedio: -1 } },

{ $limit: 1 }

]);

console.log("Categoría con mayor precio promedio:", resultado);