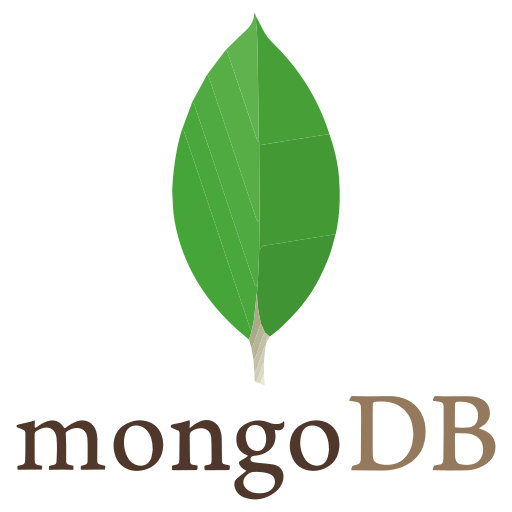
Mongo DB



Guillermo de Carlos Muñoz

23/24 - IABD - Big Data Aplicado

# 

[**Creación de contenedor docker 3**](#_3adognt5vhew)

[**Creación DB y Colecciones 5**](#_cdi3g1thrryp)

[Creación base de datos 5](#_i1uoxk4youuk)

[Creación colecciones 5](#_238je1siwupy)

[**OPERACIONES CRUD 6**](#_o4rzvp4vasb)

[Insertar documentos 6](#_oq6iz2eqkjrr)

[Colección: Productos 6](#_ke33ysk1nz68)

[Colección: Clientes 7](#_ln53x6kfp6ad)

[Colección: Pedidos 7](#_6nx06q74zdip)

[Colección: Empleados 8](#_lrzw27y39lwk)

[Colección: Proveedores 9](#_7tj9lbxmgq8r)

[Actualiza información en documentos 10](#_ktix0sq3zmj1)

[Colección: Proveedores 10](#_ba8201n9t9se)

[Colección: Clientes 10](#_udeym5z9zwvu)

[Colección: Empleados 10](#_93ud07uzpqav)

[Colección Productos 11](#_par7j2joy5pv)

[Eliminar Registros 11](#_d3978k9jjv65)

[Colección: Clientes 11](#_umyinev0xm6)

[Colección: Empleados 11](#_ryht5vjtta8p)

[Aggregation Pipelines 12](#_7d67wzoz9ejh)

[1. Encuentra el producto más vendido en la tienda 12](#_kxujk18a3umu)

[2. Identifica a los clientes que han realizado más de 3 pedidos en el último mes. 13](#_pp1xcuicixk)

[3. Calcula la ganancia total por empleado en los últimos 6 meses. 14](#_7d67wzoz9ejh)

[4. Encuentra el promedio de precios de productos por categoría. 14](#_3k0wf3iv0g6w)

[5. Agrupa los productos por proveedor y muestra el recuento de productos por proveedor. 15](#_ccz59fafhfn1)

[6. Encuentra los productos que han estado fuera de stock por más de 30 días. 15](#_fqvv6iaal40h)

[7. Identifica a los clientes que gastaron más de un cierto monto en pedidos el último trimestre. 16](#_kp4eymeqng4m)

[8. Encuentra los productos más populares basados en las calificaciones de los clientes. 17](#_5evrepkpr7zj)

[9. Realiza una agregación que muestre el salario promedio de los empleados por departamento. 17](#_uwuh1ud4adur)

[**Índices 18**](#_gs8ptlseh0np)

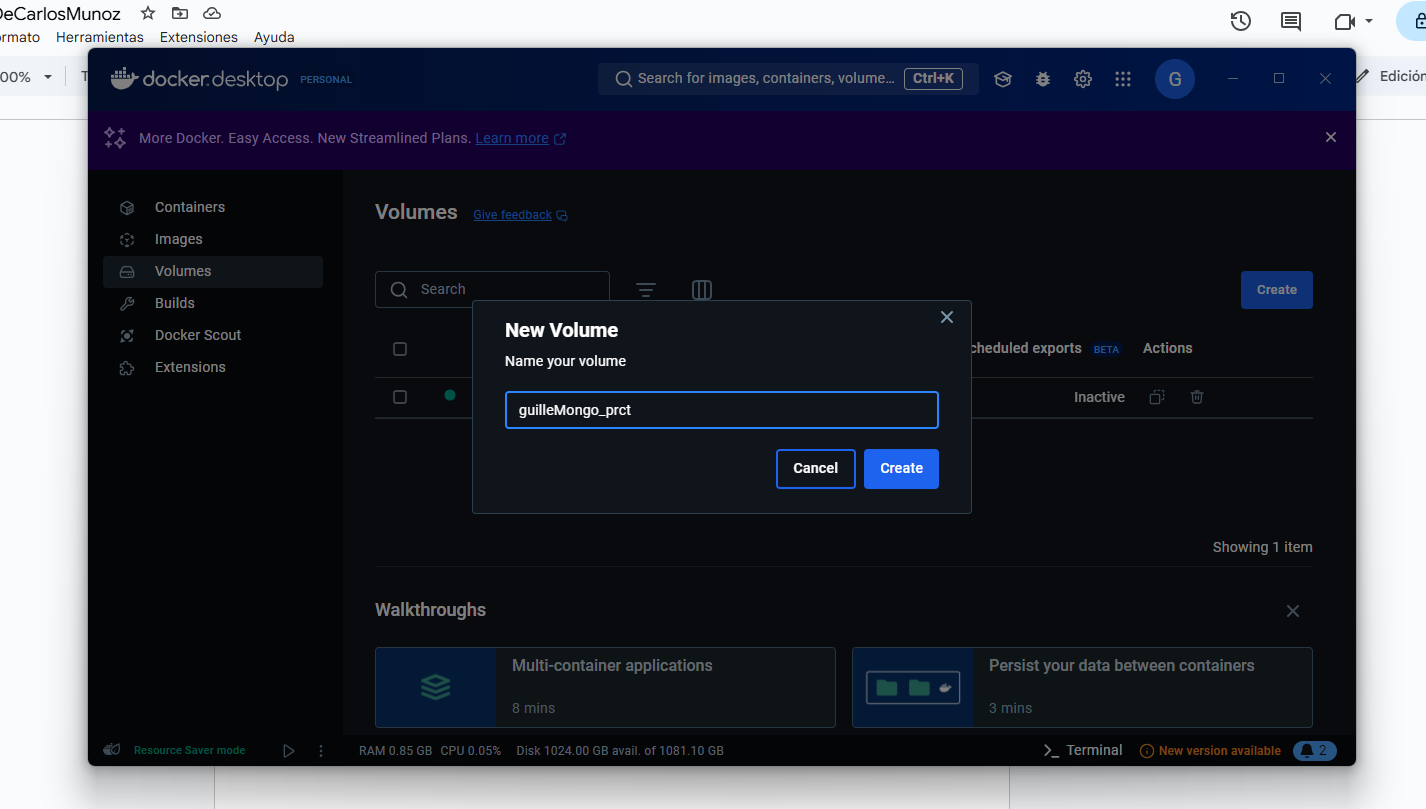
[Crea un índice simple en la colección "productos" para acelerar las búsquedas por nombre del producto. 18](#_gr7uhs9tovpj)

[Implementa un índice compuesto en la colección "pedidos" que incluya el cliente y la fecha del pedido. 18](#_rjft0ajxhfyy)

[Crea un índice de texto en la colección "productos" para habilitar búsquedas de texto completo en la descripción de los productos. 18](#_ehfwdzfjije6)

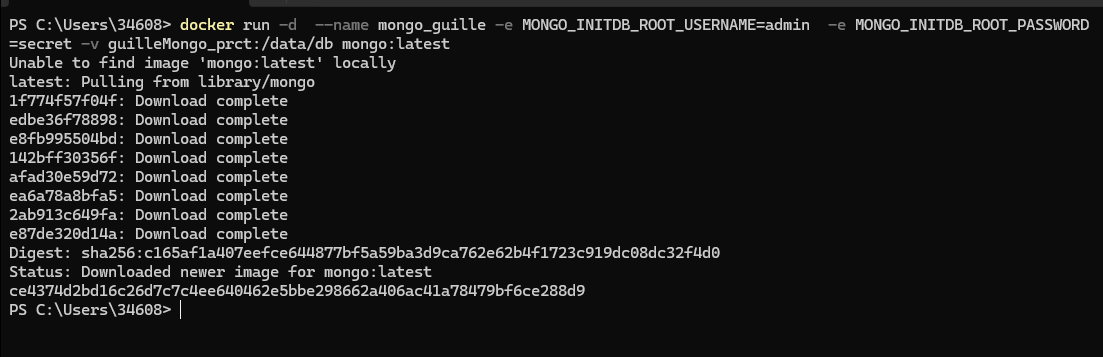
# 

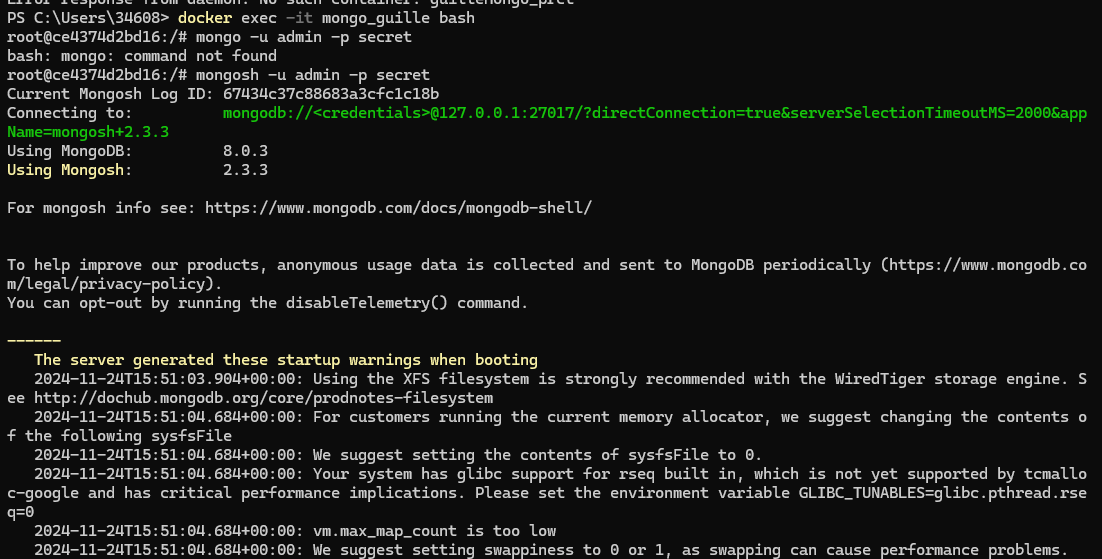
# Creación de contenedor docker

Creo el volumen al que asignare el container:  


Comando de container mongo con nombre, usuario, contraseña y dirección de volumen, usando la última versión de mongo:

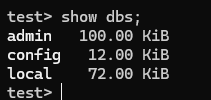
docker run -d --name mongo\_guille -e MONGO\_INITDB\_ROOT\_USERNAME=admin -e MONGO\_INITDB\_ROOT\_PASSWORD=secret -v guilleMongo\_prct:/data/db mongo:latest





Ya dentro de la bash del container de mongo.

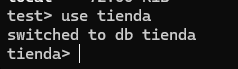
Para ver las dbs actualmente creadas.



# Creación DB y Colecciones

## Creación base de datos

use tienda



## Creación colecciones

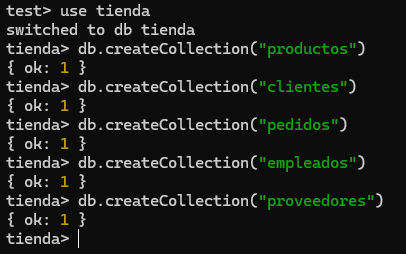
db.createCollection("productos")

db.createCollection("clientes")

db.createCollection("pedidos")

db.createCollection("empleados")

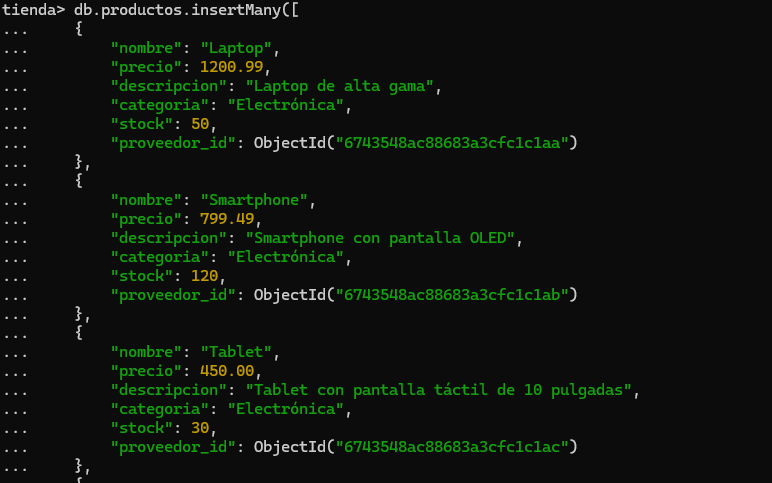
db.createCollection("proveedores")



# OPERACIONES CRUD

## Insertar documentos

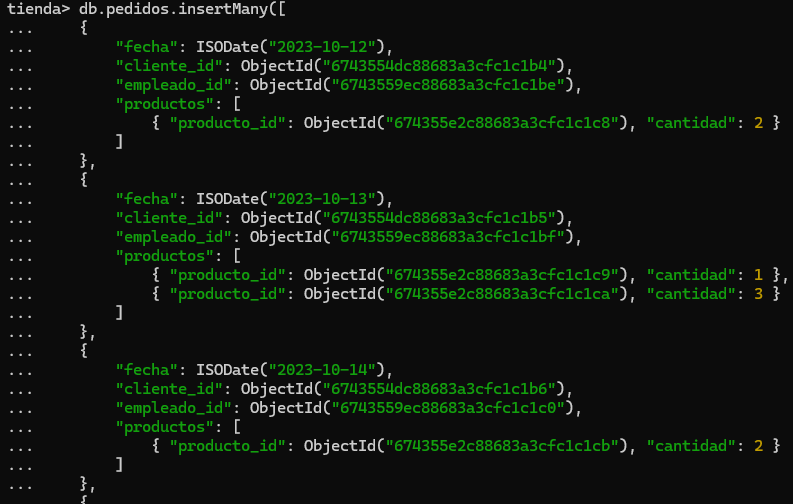
### Colección: Productos



### Colección: Clientes



### Colección: Pedidos



### Colección: Empleados

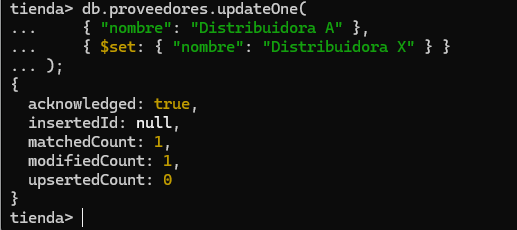


### Colección: Proveedores

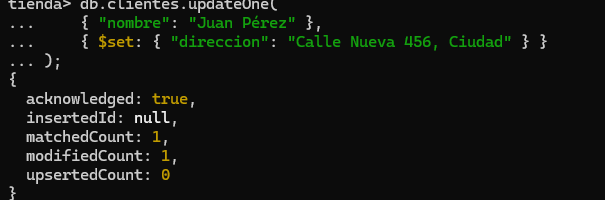


## Actualiza información en documentos

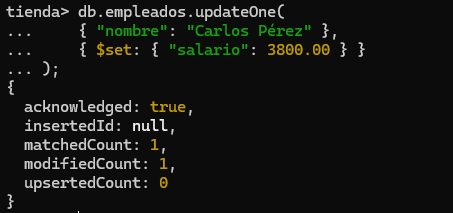
### Colección: Proveedores



### Colección: Clientes



### Colección: Empleados

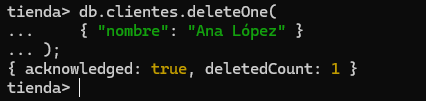


### Colección Productos

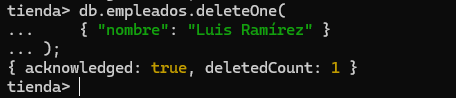


## Eliminar Registros

### Colección: Clientes



### Colección: Empleados



### 

# Aggregation Pipelines

## 1. Encuentra el producto más vendido en la tienda

db.pedidos.aggregate([

{ $unwind: "$productos" },

{ $group: { \_id: "$productos.producto\_id", total\_vendido: { $sum: "$productos.cantidad" } } },

{ $sort: { total\_vendido: -1 } },

{ $limit: 1 },

{

$lookup: {

from: "productos",

localField: "\_id",

foreignField: "\_id",

as: "producto"

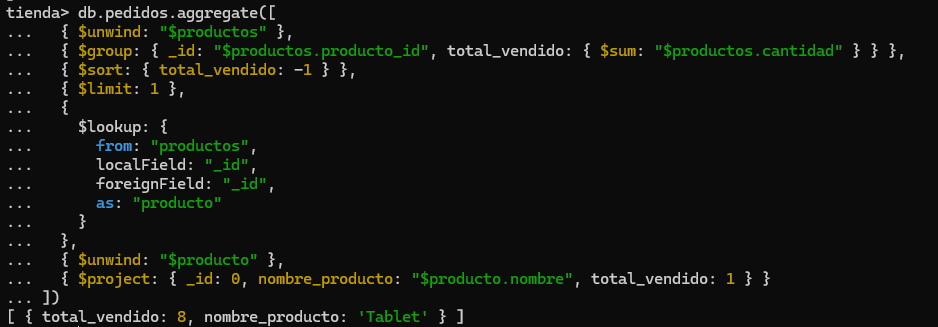
}

},

{ $unwind: "$producto" },

{ $project: { \_id: 0, nombre\_producto: "$producto.nombre", total\_vendido: 1 } }

])



## 

## 2. Identifica a los clientes que han realizado más de 3 pedidos en el último mes.

db.pedidos.aggregate([

{

$match: {

fecha: { $gte: new Date(new Date().setMonth(new Date().getMonth() - 1)) }

}

},

{

$group: {

\_id: "$cliente\_id",

totalPedidos: { $sum: 1 }

}

},

{

$match: {

totalPedidos: { $gt: 3 } // Más de 3 pedidos

}

},

{

$lookup: {

from: "Clientes",

localField: "\_id",

foreignField: "\_id",

as: "cliente"

}

},

{

$project: {

cliente: { $arrayElemAt: ["$cliente", 0] },

totalPedidos: 1

}

}

]);

## 

## 3. Calcula la ganancia total por empleado en los últimos 6 meses.

db.pedidos.aggregate([

{

$match: {

fecha: { $gte: new Date(new Date().setMonth(new Date().getMonth() - 6)) } }

},

{

$group: {

\_id: "$empleado\_id",

gananciaTotal: { $sum: "$total" }

}

},

{

$lookup: {

from: "Empleados",

localField: "\_id",

foreignField: "\_id",

as: "empleado"

}

},

{

$project: {

empleado: { $arrayElemAt: ["$empleado", 0] },

gananciaTotal: 1

}

}

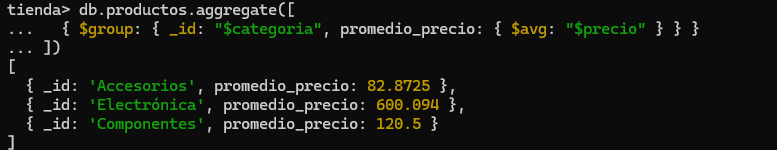
]);

## 4. Encuentra el promedio de precios de productos por categoría.

db.productos.aggregate([

{ $group: { \_id: "$categoria", promedio\_precio: { $avg: "$precio" } } }

])



## 5. Agrupa los productos por proveedor y muestra el recuento de productos por proveedor.

db.productos.aggregate([

{ $group: { \_id: "$proveedor\_id", total\_productos: { $sum: 1 } } },

{

$lookup: {

from: "proveedores",

localField: "\_id",

foreignField: "\_id",

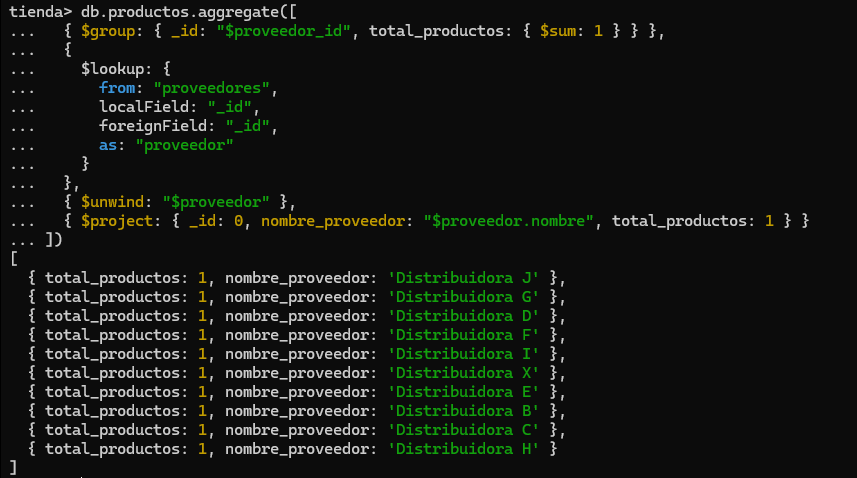
as: "proveedor"

}

},

{ $unwind: "$proveedor" },

{ $project: { \_id: 0, nombre\_proveedor: "$proveedor.nombre", total\_productos: 1 } }

])

## 6. Encuentra los productos que han estado fuera de stock por más de 30 días.

db.productos.aggregate([

{

$match: {

stock: { $eq: 0 }, // Productos fuera de stock

ultima\_actualizacion\_stock: {

$lt: new Date(new Date().setDate(new Date().getDate() - 30))

}

}

}

]);

## 7. Identifica a los clientes que gastaron más de un cierto monto en pedidos el último trimestre.

db.Pedidos.aggregate([

{

$match: {

fecha: { $gte: new Date(new Date().setMonth(new Date().getMonth() - 3)) }

}

},

{

$group: {

\_id: "$cliente\_id",

gastoTotal: { $sum: "$total" }

}

},

{

$match: {

gastoTotal: { $gt: 1000 }

}

},

{

$lookup: {

from: "Clientes",

localField: "\_id",

foreignField: "\_id",

as: "cliente"

}

},

{

$project: {

cliente: { $arrayElemAt: ["$cliente", 0] },

gastoTotal: 1

}

}

]);

## 8. Encuentra los productos más populares basados en las calificaciones de los clientes.

db.productos.aggregate([

{ $lookup: { from: "pedidos", localField: "\_id", foreignField: "productos.producto\_id", as: "pedidos" } },

{ $unwind: "$pedidos" },

{ $group: { \_id: "$\_id", total\_calificaciones: { $sum: 1 } } },

{ $sort: { total\_calificaciones: -1 } },

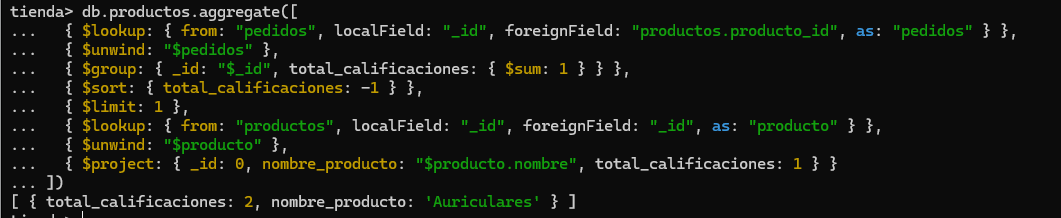
{ $limit: 1 },

{ $lookup: { from: "productos", localField: "\_id", foreignField: "\_id", as: "producto" } },

{ $unwind: "$producto" },

{ $project: { \_id: 0, nombre\_producto: "$producto.nombre", total\_calificaciones: 1 } }

])

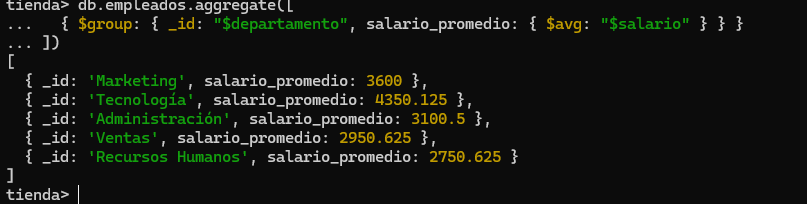


## 9. Realiza una agregación que muestre el salario promedio de los empleados por departamento.

db.empleados.aggregate([

{ $group: { \_id: "$departamento", salario\_promedio: { $avg: "$salario" } } }

])



# Índices

## Crea un índice simple en la colección "productos" para acelerar las búsquedas por nombre del producto.

db.productos.createIndex({ nombre: 1 });



## Implementa un índice compuesto en la colección "pedidos" que incluya el cliente y la fecha del pedido.

db.pedidos.createIndex({ cliente\_id: 1, fecha: 1 });



## Crea un índice de texto en la colección "productos" para habilitar búsquedas de texto completo en la descripción de los productos.

db.productos.createIndex({ descripcion: "text" });

## 