

# MongoDB

EJ1:

Crear BD 'domotica':

```
liga> use domotica
switched to db domotica
```

Crear collection 'mediciones':

```
domotica> db.createCollection("mediciones")
{ ok: 1 }
```

Insertar 3 mediciones con fechas distintas:

```
domotica> db.mediciones.insertMany([
...     {
...         "sensorId": "sensor_12345",
...         "ts": new Date(2024, 8, 15),
...         "tipo": "hum",
...         "valor": 60,
...         "ubicacion": "Ordes (A Coruña)"
...     },
...     {
...         "sensorId": "sensor_67890",
...         "ts": new Date(2024, 8, 20),
...         "tipo": "temp",
...         "valor": 24.0,
...         "ubicacion": "Santiago de Compostela (A Coruña)"
...     },
...     {
...         "sensorId": "sensor_54321",
...         "ts": new Date(2026, 1, 22),
...         "tipo": "temp",
...         "valor": 21.0,
...         "ubicacion": "A Coruña (A Coruña)"
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6971f203c729723e378ce5be'),
    '1': ObjectId('6971f203c729723e378ce5bf'),
    '2': ObjectId('6971f203c729723e378ce5c0')
  }
}
```

### Consultar por sensorId y rango ts (últimas 24h)

```
domotica> db.mediciones.find({ "ts": { $gt: ISODate("2026-01-21") } }, { _id:1})  
[ { _id: ObjectId('6971f203c729723e378ce5c0') } ]
```

### Consultar mediciones por encima de umbral.

```
domotica> db.mediciones.find({ "valor": { $gt: 50.0 } })  
[  
  {  
    _id: ObjectId('6971f203c729723e378ce5be'),  
    sensorId: 'sensor_12345',  
    ts: ISODate('2024-09-15T00:00:00.000Z'),  
    tipo: 'hum',  
    valor: 60,  
    ubicacion: 'Ordes (A Coruña)'  
  }  
]
```

### Corregir mediciones erróneas (updateMany).

```
domotica> db.mediciones.updateMany({ "valor": { $gt: 50.0 } }, { $set: { "valor":50 } })  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

### Eliminar mediciones anteriores a una fecha (deleteMany).

```
domotica> db.mediciones.deleteMany( { "ts": { $gt: ISODate("2025-01-01") } } )  
{ acknowledged: true, deletedCount: 1 }
```

## EJ2:

### Crear BD 'tienda':

```
domotica> use tienda  
switched to db tienda
```

### Colecciones: clientes, pedidos:

```
tienda> db.createCollection("clientes")  
{ ok: 1 }  
tienda> db.createCollection("pedidos")  
{ ok: 1 }
```

## Insertar 2 clientes:

```
tienda> db.clientes.insertMany([
...   {
...     "idCliente": "cliente_001",
...     "nombre": "Juan Pérez",
...     "emails": [{
...       "emailPrincipal": "juan.perez@example.com",
...       "secundario": "jperez@example.com"}
...   ],
...   "direcciones": [
...     {
...       "calle": "Calle Falsa 123",
...       "ciudad": "Madrid",
...       "codigoPostal": "28013",
...       "pais": "España"
...     },
...     {
...       "calle": "Avenida Siempre Viva 742",
...       "ciudad": "Barcelona",
...       "codigoPostal": "08001",
...       "pais": "España"
...     }
...   ]
... },
... {
...   "idCliente": "cliente_002",
...   "nombre": "María Gómez",
...   "emails": [
...     {"emailPrincipal": "maria.gomez@example.com"}
...   ],
...   "direcciones": [
...     {
...       "calle": "Calle Luna 45",
...       "ciudad": "Valencia",
...       "codigoPostal": "46001",
...       "pais": "España"
...     }
...   ]
... }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6973360ebe8e9562298ce5b6'),
    '1': ObjectId('6973360ebe8e9562298ce5b7')
  }
}
```

### Insertar 3 pedidos:

```
tienda> db.pedidos.insertMany([
...   {
...     "idPedido": 1,
...     "idCliente": "cliente_001",
...     "fechaPedido": new Date(2024, 8, 16),
...     "lineas": [{
...       "sku": "producto_123",
...       "nombre": "Libro de Ciencia Ficción",
...       "cantidad": 2,
...       "precio": 15.50
...     }
...   ],
...   "estado": "Enviado"
... },
...   {
...     "idPedido": 2,
...     "idCliente": "cliente_002",
...     "fechaPedido": new Date(2024, 8, 21),
...     "lineas": [{
...       "sku": "producto_456",
...       "nombre": "Juego de Mesa Estratégico",
...       "cantidad": 1,
...       "precio": 35.00
...     }
...   ],
...   "estado": "enviado"
... },
...   {
...     "idPedido": 3,
...     "idCliente": "cliente_002",
...     "fechaPedido": new Date(2026, 1, 23),
...     "lineas": [{
...       "sku": "producto_789",
...       "nombre": "Auriculares Inalámbricos",
...       "cantidad": 120,
...       "precio": 85.75
...     }
...   ],
...   "estado": "Procesando"
... }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6973341fbe8e9562298ce5b3'),
    '1': ObjectId('6973341fbe8e9562298ce5b4'),
    '2': ObjectId('6973341fbe8e9562298ce5b5')
  }
}
```

### Pedidos de un cliente por idCliente.

```
tienda> db.pedidos.find({"idCliente": "cliente_001"})
[
  {
    _id: ObjectId('6973341fbe8e9562298ce5b3'),
    idPedido: 1,
    idCliente: 'cliente_001',
    fechaPedido: ISODate('2024-09-16T00:00:00.000Z'),
    lineas: [
      {
        sku: 'producto_123',
        nombre: 'Libro de Ciencia Ficción',
        cantidad: 2,
        precio: 15.5
      }
    ],
    estado: 'Enviado'
  }
]
```

### Pedidos con cantidad > 100

```
tienda> db.pedidos.find({"lineas.cantidad": {$gt:100} })
[
  {
    _id: ObjectId('6973341fbe8e9562298ce5b5'),
    idPedido: 3,
    idCliente: 'cliente_002',
    fechaPedido: ISODate('2026-02-23T00:00:00.000Z'),
    lineas: [
      {
        sku: 'producto_789',
        nombre: 'Auriculares Inalámbricos',
        cantidad: 120,
        precio: 85.75
      }
    ],
    estado: 'Procesando'
  }
]
```

### Cambiar el estado de un pedido.

```
tienda> db.pedidos.updateOne({"idPedido":{$eq:3}},{$set: {"estado" : "enviado"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

## Cambiar el email principal de un cliente.

```
tienda> db.clientes.updateOne(
...   { idCliente: "cliente_001" },
...   { $set: { "emails.0.emailPrincipal": "juan.perez.rodrigo@example.com" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```