

- [Documentación oficial](#)
- [MongoDB University](#)
- [Operaciones CRUD en MongoDB](#)

Insert

Insertar un documento

```
db.mi_coleccion.insertOne({
  pais: 'España',
  ciudad: 'Madrid',
  nombreCliente: 'Ana',
  pizzas: [
    { pizza: 'Jamón', puntos: 3 }
  ],
  precio: 100,
  fecha: new Date()
})
```

Insertar múltiples documentos

```
db.mi_coleccion.insertMany([
  {
    pais: 'España',
    ciudad: 'Madrid',
    nombreCliente: 'Ana',
    pizzas: [
      { pizza: 'Jamón', puntos: 3 }
    ],
    precio: 100,
    fecha: new Date()
  },
  {
    pais: 'España',
    ciudad: 'Ourense',
    nombreCliente: 'Jose',
    pizzas: [
      { pizza: 'Piña', puntos: 2 }
    ],
    precio: 21,
    fecha: new Date()
  }
])
```

Buscar

Buscar un documento

```
db.mi_coleccion.findOne({pais: 'España'})
```

Buscar múltiples documentos

```
db.mi_coleccion.find({pais: 'España'})
```

Búsquedas extendidas.

Busca documentos con pizza Napolitana de 5 puntos o más

```
db.mi_coleccion.find({
  pizzas: {
    $elemMatch: { pizza: 'Napolitana', puntos: { $gte: 5 } }
  }
})
```

Busca todos los documentos, pero sólo muestra las pizzas Napolitana de 5 puntos o más

```
db.mi_coleccion.find(
  {}, // query
  { // projection
    pizzas: {
      $elemMatch: { pizza: 'Napolitana', puntos: { $gte: 5 } }
    }
  }
)
```

Actualizar

Reemplazar UN documento entero

```
db.mi_coleccion.updateOne(
  { nombreCliente: 'Ana' }, // query
  { ciudad: 'Ourense' }    // nuevo documento
)
```

Actualizar propiedad específica de UN documento

```
db.mi_coleccion.update(
  { nombreCliente: 'Ana' }, // query
```

```
{ $set: { ciudad: 'Ourense' } } // nuevo documento
)
```

Actualizar propiedad específica de TODOS los documento con nombre Ana

```
db.mi_coleccion.update(
  { nombreCliente: 'Ana' }, // query
  { $set: { ciudad: 'Ourense' } }, // nuevo documento
  { multi: true }
)
```

Geolocalización

Tutorial geo espacial

<https://docs.mongodb.com/manual/tutorial/geospatial-tutorial/>

Agregación

<https://docs.mongodb.com/manual/aggregation/>

```
db.mi_coleccion.aggregate([
  {
    $group: {
      _id: '$pais', // $pais
      total: { $sum: '$precio' },
      media_precio: { $avg: '$precio' },
      precio_minimo: { $min: '$precio' },
      precio_maximo: { $max: '$precio' }
    }
  },
  {
    $project: {
      _id: 1,
      total: 1
    }
  }
])
```

```
db.mi_coleccion.aggregate.aggregate([
  {
    $lookup:
    {
      from: 'inventory',
      localField: 'item',
      foreignField: 'sku',
```

```
        as: 'inventory_docs'
      }
    }
  })
}
```

Map reduce

```
// Map:
// { "España": ["Piña", "Piña", "Jamón", "Napolitana"] }
// { "España-Ourense": ["Piña", "Piña"] }
// ...

let map = function() {
  this.pizzas.map(p => {
    emit(`${this.pais}`, p.pizza)
    emit(`${this.pais}-${this.ciudad}`, p.pizza)
  })
}

// Reduce:
// { "España": { "Piña": 2, "Jamon": 1, "Napolitana": 1 } }
// { "España-Ourense": { "Piña": 2 } }
let reduce = function(key, values) {
  let sumaDePizasPorSabor = {}
  values.map(k => {
    sumaDePizasPorSabor[k] ?
      sumaDePizasPorSabor[k]++ :
      sumaDePizasPorSabor[k] = 1
  })
  return sumaDePizasPorSabor
}

let options = { out: {inline: 1} }

db.mi_coleccion.mapReduce(map, reduce, options)
```