

1. Insertar 2 documentos en la colección clientes con '_id' no repetidos

```
JS libros.js JS clientes.js X
JS clientes.js > ...
1 db = db.getSiblingDB("base1")
2 db.clientes.drop()
3 db.clientes.insertOne({ _id:1, nombre:'Lopez Marcos', domicilio:'Colon 111', provincia:'Cordoba' })
4 db.clientes.insertOne({ _id:2, nombre:'Perez Ana', domicilio:'San Martin 222', provincia:'Santa Fe' })

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS MongoDB Shell

base1> load("clientes.js")
true
base1> db.clientes.find()
[
  {
    _id: 1,
    nombre: 'Lopez Marcos',
    domicilio: 'Colon 111',
    provincia: 'Cordoba'
  },
  {
    _id: 2,
    nombre: 'Perez Ana',
    domicilio: 'San Martin 222',
    provincia: 'Santa Fe'
  }
]
base1> 
```

2. Insertar otro documento con clave repetida

```
JS libros.js JS clientes.js X
JS clientes.js > ...
1 db = db.getSiblingDB("base1")
2 db.clientes.drop()
3 db.clientes.insertOne({ _id:1, nombre:'Lopez Marcos', domicilio:'Colon 111', provincia:'Cordoba' })
4 db.clientes.insertOne({ _id:2, nombre:'Perez Ana', domicilio:'San Martin 222', provincia:'Santa Fe' })
5 db.clientes.insertOne({ _id:1, nombre:'Sanchez Javier', domicilio:'Belgrano 444', provincia:'Jujuy' })

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS MongoDB Shell

base1> load("clientes.js")
true
base1> db.clientes.find()
[
  {
    _id: 1,
    nombre: 'Lopez Marcos',
    domicilio: 'Colon 111',
    provincia: 'Cordoba'
  },
  {
    _id: 2,
    nombre: 'Perez Ana',
    domicilio: 'San Martin 222',
    provincia: 'Santa Fe'
  }
]
base1> load("clientes.js")
MongoServerError: E11000 duplicate key error collection: base1.clientes index: _id_ dup key: { _id: 1 }
base1> 
```

3. Mostrar todos los documentos de la colección libros

```
JS libros.js  X  JS clientes.js

JS libros.js > ...
1 db = db.getSiblingDB("base1")
2 db.libros.drop()
3 db.libros.insertOne({ codigo:1, nombre:'El aleph', autor:'Borges', editoriales:['El Planeta','Siglo XXI'] })
4 db.libros.insertOne({ codigo:2, nombre:'Martin Fierro', autor:'Jose Hernandez', editoriales:['Planeta'] })
5 db.libros.insertOne({ codigo:3, nombre:'Aprenda PHP', autor:'Mario Molina', editoriales:['Planeta'] })
6 db.libros.insertOne({ codigo:4, nombre:'Java en 10 minutos', autor:'Barros Sergio', editoriales:['Planeta','Sig

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  MongoDB Shell + v []

base1> db.libros.find()
[
  {
    _id: ObjectId('68d2e95a821e817cad735191'),
    codigo: 1,
    nombre: 'El aleph',
    autor: 'Borges',
    editoriales: [ 'El Planeta', 'Siglo XXI' ]
  },
  {
    _id: ObjectId('68d2e95a821e817cad735192'),
    {
      _id: ObjectId('68d2e95a821e817cad735192'),
      codigo: 2,
      nombre: 'Martin Fierro',
      autor: 'Jose Hernandez',
      editoriales: [ 'Planeta' ]
    },
    {
      _id: ObjectId('68d2e95a821e817cad735193'),
      codigo: 3,
      nombre: 'Aprenda PHP',
      autor: 'Mario Molina',
      editoriales: [ 'Planeta' ]
    },
  ],
]
```

```
JS libros.js  X  JS clientes.js

JS libros.js > ...
1 db = db.getSiblingDB("base1")
2 db.libros.drop()
3 db.libros.insertOne({ codigo:1, nombre:'El aleph', autor:'Borges', editoriales:['El Planeta','Siglo XXI'] })
4 db.libros.insertOne({ codigo:2, nombre:'Martin Fierro', autor:'Jose Hernandez', editoriales:['Planeta'] })
5 db.libros.insertOne({ codigo:3, nombre:'Aprenda PHP', autor:'Mario Molina', editoriales:['Planeta'] })
6 db.libros.insertOne({ codigo:4, nombre:'Java en 10 minutos', autor:'Barros Sergio', editoriales:['Planeta','Sig

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  MongoDB Shell + v []

{
  _id: ObjectId('68d2e95a821e817cad735192'),
  codigo: 2,
  nombre: 'Martin Fierro',
  autor: 'Jose Hernandez',
  editoriales: [ 'Planeta' ]
},
{
  _id: ObjectId('68d2e95a821e817cad735193'),
  codigo: 3,
  nombre: 'Aprenda PHP',
  autor: 'Mario Molina',
  editoriales: [ 'Planeta' ]
},
{
  _id: ObjectId('68d2e95a821e817cad735194'),
  codigo: 4,
  nombre: 'Java en 10 minutos',
  autor: 'Barros Sergio',
  editoriales: [ 'Planeta', 'Siglo XXI' ]
}
]
base1> |
```