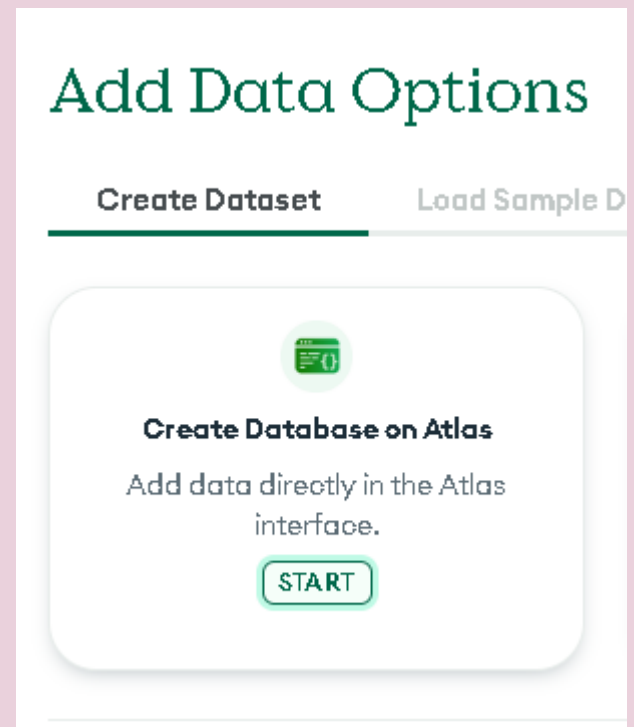
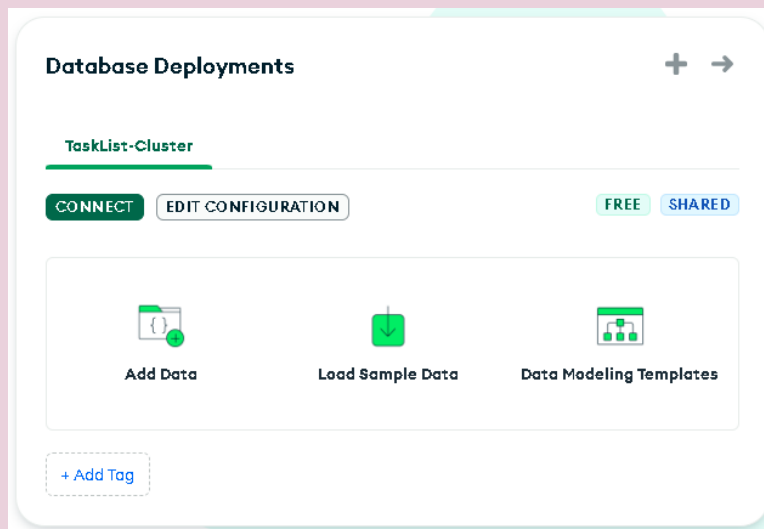


Clúster MongoDB Atlas

1. Crea un cluster usando [MongoDB Atlas](#).
2. Dentro del clúster crea una base de datos nueva para tu aplicación de lista de tareas.
 - Recuerda que en el proceso de creación debes definir también el nombre de tu primera colección



Add Data Options

Create DatasetLoad Sample DataMigrate DatabaseImport File

[← Create Database on Atlas](#)

Database name
Enter a name for your new database

Collection name
Enter a name for your new collection

[Learn more about database and collection naming](#)

Create Database

3. En base al modelo de datos que creaste en módulos anteriores debés crear las colecciones que definiste en dicho modelo.

| taskList | | | | | | | |
|--|-----------|-------------------|-------------------|--------------|---------|------------|----------------|
| LOGICAL DATA SIZE: 514B STORAGE SIZE: 36KB INDEX SIZE: 36KB TOTAL COLLECTIONS: 1 | | | | | | | |
| CREATE COLLECTION | | | | | | | |
| Collection Name | Documents | Logical Data Size | Avg Document Size | Storage Size | Indexes | Index Size | Avg Index Size |
| tasks | 5 | 514B | 103B | 36KB | 1 | 36KB | 36KB |

B

×

Create Collection

Database name ?

taskList

Collection name ?

users

Additional Preferences

Select



Cancel

Create

4. Para cada colección creada debes crear al menos un documento con datos de ejemplo de acuerdo a los campos que definiste que tendrían los documentos de tus colecciones en el modelo.

Insert Document

To Collection tasks

VIEW  

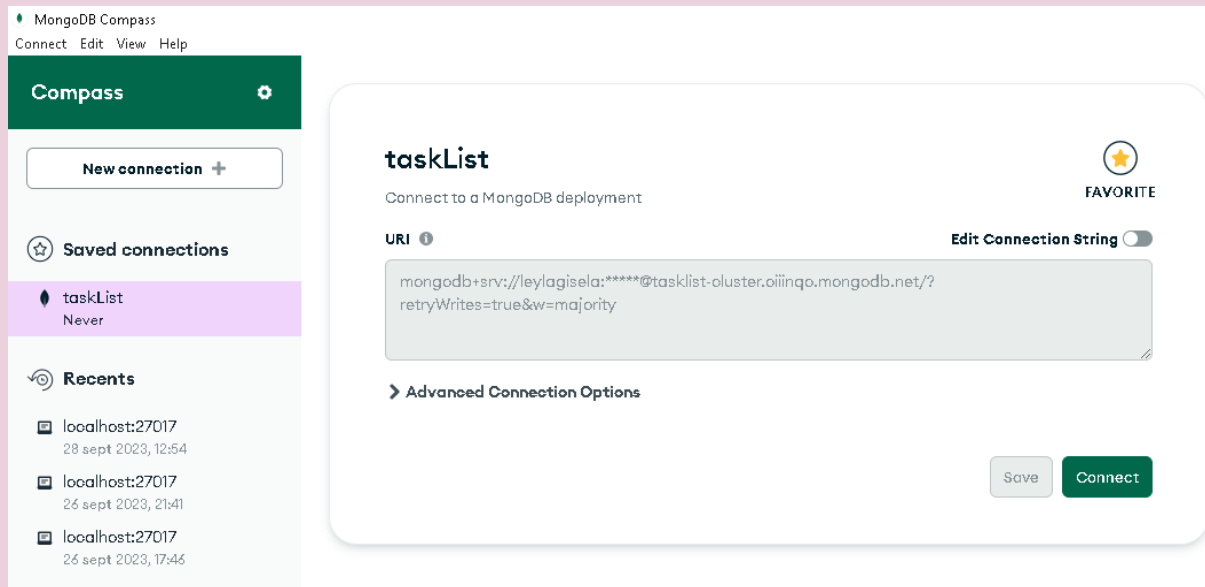
```
1 {
2   "_id": { "$oid": "651eb68d22fdc2731bea83e1" },
3   "name": "Estudiar Git",
4   "description": "Realizar las actividades de la plataforma",
5   "isCompleted": true
6 }
```

Cancel Insert

| taskList.tasks | |
|--|----------------------------------|
| STORAGE SIZE: 4KB | LOGICAL DATA SIZE: 0B |
| TOTAL DOCUMENTS: 0 | INDEXES TOTAL SIZE: 4KB |
| Find | Indexes |
| Schema Anti-Patterns 0 | Aggregation |
| Search Indexes | |
| Filter | Type a query: { field: 'value' } |
| QUERY RESULTS: 1-5 OF 5 | |
| <pre>_id: ObjectId('651eb68d22fdc2731bea83e1') name: "Estudiar Git" description: "Realizar las actividades de la plataforma" isCompleted: true</pre> | |
| <pre>_id: ObjectId('651eb78422fdc2731bea83e2') name: "Aprender HTML Y CSS" description: "Crear mi CV" isCompleted: true</pre> | |
| <pre>_id: ObjectId('651eb83422fdc2731bea83e4') name: "React Fundamentos" description: "Estudiar Librerías UI en React" isCompleted: false</pre> | |
| <pre>_id: ObjectId('651eb84522fdc2731bea83e5') name: "Ejercicios"</pre> | |

5. Para cada colección añade en el documento de Google Docs como serían los comandos para las operaciones del CRUD usando el [driver de acuerdo a tu stack](#), teniendo en cuenta el documento de ejemplo o el documento creado con el comando Create, para las operaciones de Read, Update y Delete.

Conexión a MongoDB Atlas desde Compass



CRUD en MongoDB se refiere a las operaciones básicas que puedes realizar en una base de datos, las cuales son:

1. Crear (Create):

- Operación: insert o insertMany
- Descripción: Esta operación se utiliza para añadir nuevos documentos a una colección. Puedes insertar un solo documento con insertOne o varios documentos con insertMany.

Ejemplo de insertOne:

- user

```
> db.users.insertOne({user:"legive", password:"1234", email:"leylagisela@gmail.com"})
< {
  acknowledged: true,
  insertedId: ObjectId("651ec4603030acdf23123add")
}
> db.users.find()
< {
  _id: ObjectId("651ec4603030acdf23123add"),
  user: 'legive',
  password: '1234',
  email: 'leylagisela@gmail.com'
}
```

taskList.users

STORAGE SIZE: 8KB LOGICAL DATA SIZE: 91B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 8KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-1 OF 1

_id: ObjectId('651ec4603030acdf23123add')

user: "legive"

password: "1234"

email: "leylagisela@gmail.com"

- task

```
>_MONGOSH
> db.tasks.insertOne({name:"Base de datos", description:"Operarla en MongoDb", isComplete:false})
< {
  acknowledged: true,
  insertedId: ObjectId("651ed2063030acdf23123ae0")
}
```



```
_id: ObjectId('651ed2063030acdf23123ae0')
name: "Base de datos"
description: "Operarla en MongoDb"
isComplete: false
```

Ejemplo de insertMany:

- user

```
>_MONGOSH
> db.users.insertMany([(user:"user2", password:"user21234", email:"user2@gmail.com"), (user:"user3", password:"user31234", email:"user3@gmail.com")])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("651ed2a73030acdf23123ae1"),
    '1': ObjectId("651ed2a73030acdf23123ae2")
  }
}
```

taskList.users

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 269B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 20KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('651ec4603030acdf23123add')
user: "legive"
password: "1234"
email: "leylagisela@gmail.com"
```

```
_id: ObjectId('651ed2a73030acdf23123ae1')
user: "user2"
password: "user21234"
email: "user2@gmail.com"
```

```
_id: ObjectId('651ed2a73030acdf23123ae2')
user: "user3"
password: "user31234"
email: "user3@gmail.com"
```

- tasks

```
> db.tasks.insertMany([
  {name:"Terminar curso de bases de la programación", description:"plataforma de ADA", isComplete:false},
  {name:"web Fundamental", description:"Terminar temas y actividades", isComplete:false}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("651ed1723030acdf23123ade"),
    '1': ObjectId("651ed1723030acdf23123adf")
  }
}
```

taskList.tasks

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 841B TOTAL DOCUMENTS: 8 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Filter

Type a query: { field: 'value' }

```
_id: ObjectId('651eb85622fd021810e883e6')
name: "Express"
description: "Desarrollar las actividades de la plataforma"
isCompleted: false
```

```
_id: ObjectId('651eb85622fdc2731bea83e6')
name: "Mongo DB"
description: ""
isCompleted: false
```

```
_id: ObjectId('651ed1723030acdf23123ade')
name: "Terminar curso de bases de la programación"
description: "plataforma de ADA"
isComplete: false
```

```
_id: ObjectId('651ed1723030acdf23123adf')
name: "web Fundamental"
description: "Terminar temas y actividades"
isComplete: false
```

2. Leer (Read):

- Operación: find
- Descripción: La operación de lectura se utiliza para recuperar datos de la base de datos. Puedes especificar criterios de búsqueda para filtrar los resultados.

Ejemplo básico:

User

```
>_MONGOSH
>
> db.users.find()
< [
  {
    _id: ObjectId("651ec4603030acdf23123add"),
    user: 'legive',
    password: '1234',
    email: 'leylagisela@gmail.com'
  },
  {
    _id: ObjectId("651ed2a73030acdf23123ae1"),
    user: 'user2',
    password: 'user21234',
    email: 'user2@gmail.com'
  },
  {
    _id: ObjectId("651ed2a73030acdf23123ae2"),
    user: 'user3',
    password: 'user31234',
    email: 'user3@gmail.com'
  }
]
```

Ejemplo con criterios de búsqueda:

```
>_MONGOSH
> db.users.findOne({user:"legive"})
< {
  _id: ObjectId("651ec4603030acdf23123add"),
  user: 'legive',
  password: '1234',
  email: 'leylagisela@gmail.com'
}
```


3. Actualizar (Update):

- Operación: update o updateMany
- Descripción: La operación de actualización se utiliza para modificar documentos existentes en la base de datos. Puedes actualizar un solo documento con updateOne o varios documentos con updateMany.

Ejemplo de updateOne:

```
> db.users.updateOne({user:"legive"},{$set:{email:"leylagisela@yahoo.com"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.users.findOne({user:"legive"})
< {
  _id: ObjectId("651ec4603030acdf23123add"),
  user: 'legive',
  password: '1234',
  email: 'leylagisela@yahoo.com'
}
Atlas atlas-bnawkh-shard-0 [primary] taskList>|
```

Ejemplo de updateMany:

```
> db.tasks.updateMany({},{$set:{isComplete:true}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 5,
  upsertedCount: 0
}
```

4. Eliminar (Delete):

- Operación: deleteOne o deleteMany
- Descripción: La operación de eliminación se utiliza para eliminar documentos de la base de datos. Puedes eliminar un solo documento con deleteOne o varios documentos con deleteMany.

Ejemplo de deleteOne:

```
> db.users.deleteOne({user:"user2"})
< {
  acknowledged: true,
  deletedCount: 1
}
```

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('651ec4603030acdf23123add')
user: "legive"
password: "1234"
email: "leylagisela@yahoo.com"
```

```
_id: ObjectId('651ed2a73030acdf23123ae2')
user: "user3"
password: "user31234"
email: "user3@gmail.com"
```

Ejemplo de deleteMany:

```
> _MONGOSH
MongoDB Shell v2.1.0
> db.tasks.deleteMany({isComplete:false})
< {
  acknowledged: true,
  deletedCount: 2
}
```