



Mongoose e MongoDB Atlas

Discentes:

Anderson Azevedo
Erickson Camilo
Marlon Silva
Melque Rodrigues
Thomas Almeida

Introdução ao MongoDB



- Banco de Dados NoSQL;
- Vem da palavra “humongous”, que em inglês significa “gigantesco”;
- Não utiliza tabelas e relações fixas;
- Armazenar e gerenciar grandes quantidades de dados.

Características Principais do MongoDB



- Web Analytics;
- Estruturas de Dados Complexas;
- Redundância de Dados.

Como o MongoDB Armazena Dados?



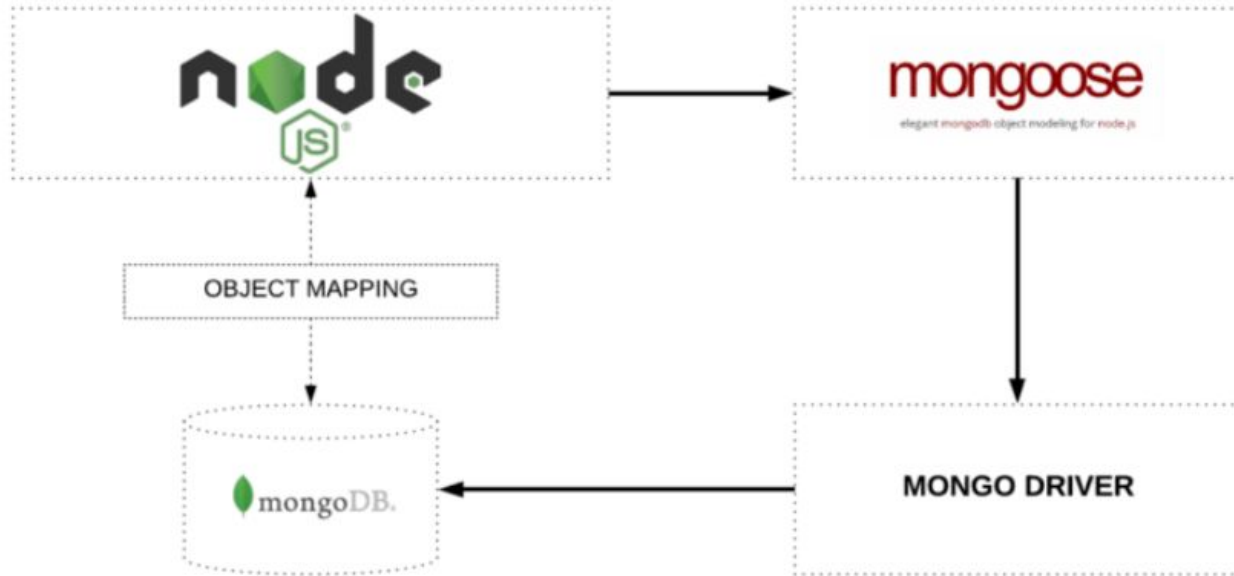
- JSON (JavaScript Object Notation) e BSON (Binary JSON);
- CSV (Comma Separated Values) e XML (eXtensible Markup Language);
- O MongoDB escolhe utilizar JSON/BSON

O que é o Mongoose?



- Mongoose é um biblioteca de Modelagem de Dados de Objeto (ODM) para MongoDB e Node.js

Mapeamento de objetos entre o Node e o MongoDB, gerenciado por meio do Mongoose



Características do Mongoose



- Modelagem de Objetos
- Validação
- Tipos de Dados
- Queries

Vantagens do Mongoose



Facilidade de uso: A sintaxe do Mongoose é mais próxima do JavaScript, tornando mais fácil trabalhar com dados do MongoDB.

Produtividade: A modelagem de objetos e a validação integrada aumentam a produtividade e reduzem o número de erros.

Integração com Node.js: Se integra com o Node.js, tornando-o a escolha ideal para aplicações back-end construídas com essa tecnologia.

Desvantagens do Mongoose



Curva de aprendizado: Requer um certo aprendizado para dominar todas as suas funcionalidades.

Performance: Em alguns cenários muito específicos, o Mongoose pode ter um leve impacto no desempenho em comparação com o MongoDB nativo. Ex: Consultas complexas e personalizadas.

Configuração inicial do Mongoose



```
npm install mongoose --save
```

```
// getting-started.js
```

```
const mongoose = require('mongoose');  
main().catch(err => console.log(err));  
async function main() {  
  await mongoose.connect('mongodb://127.0.0.1:27017/test');  
}
```

Definindo os Schemas



```
const questaoSchema = new mongoose.Schema({  
  enunciado: String;  
  alternativas: {  
    type: Array,  
    required: true,  
  }  
});  
  
const Questao = mongoose.model('Questao', questaoSchema);  
  
export default Questao;
```

Operações básicas no Model



```
await Questao.create({ enunciado: 'Questão um' });
```

```
await Questao.create([  
    { enunciado: 'Questão dois' },  
    { enunciado: 'Questão três' },  
]);
```

```
await Questao.deleteOne({ name: 'Questão um' });
```

```
await Questao.deleteMany({ name: /Questão/ });
```

```
await Person.updateOne({ name: 'Jean-Luc Picard' }, { ship: 'USS Enterprise' });
```

Queries no Mongoose



```
const pessoa = await Pessoa.findOne({ 'nome.last': 'Dantas' }, 'nome idade');
```

```
console.log('%s %s tem %d anos.', pessoa.nome.first, pessoa.nome.last, pessoa.idade);
```

```
const query = Pessoa.findOne({ 'nome.last': 'Dantas' });
```

```
query.select('nome idade');
```

```
const pessoa = await query.exec();
```

findOne()

find()

count()

Queries no Mongoose



await **Person**.

```
find({
  occupation: /host/,
  'name.last': 'Ghost',
  age: { $gt: 17, $lt: 66 },
  likes: { $in: ['vaporizing', 'talking'] }
}).
limit(10).
sort({ occupation: -1 }).
select({ name: 1, occupation: 1 }).
exec();
```

await **Person**.

```
find({ occupation: /host/ }).
where('name.last').equals('Ghost').
where('age').gt(17).lt(66).
where('likes').in(['vaporizing', 'talking']).
limit(10).
sort('-occupation').
select('name occupation').
exec();
```

Queries no Mongoose



```
await Pessoa.find().sort({ age: -1 });
```

```
await Pessoa.find().sort({ age: 1 });
```

```
await Pessoa.find().sort({ age: 1, weight: -1 });
```

MongoDB Atlas



MongoDB Atlas

The multi-cloud developer data platform available on AWS, Azure, and Google Cloud



Serverless

from \$0.10/million reads

[Sign Up](#)

(i) Per 1 million reads

For serverless applications with variable or infrequent traffic. Minimal configuration required.

- ✓ Up to 1TB of storage
- ✓ Resources scale seamlessly to meet your workload
- ✓ Pay only for the operations you run
- ✓ Always-on security and backups



Dedicated

from \$57/month

[Sign Up](#)

(i) Estimated based on \$0.08 per hour

For production applications with sophisticated workload requirements. Advanced configuration controls.

- ✓ 10GB to 4TB of storage
- ✓ 2GB to 768GB RAM
- ✓ Network isolation and fine-grained access controls
- ✓ Multi-region and multi-cloud options available



Shared

from \$0/month

[Try for Free](#)

(i) Free forever for free clusters

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

- ✓ 512MB to 5GB of storage
- ✓ Shared RAM
- ✓ Upgrade to dedicated clusters for full functionality
- ✓ No credit card required to start

Referências



Mardan, A., & Mardan, A. (2018). Boosting node. Js and MongoDB with mongoose. Practical Node. js: Building Real-World Scalable Web Apps, 239-276.

Documentação do Mongoose. Disponível em: <https://mongoosejs.com/>.

Documentação do MongoDB. Disponível em: <https://www.mongodb.com/docs/>

Introdução ao Mongoose para MongoDB. Disponível em:
<https://www.freecodecamp.org/portuguese/news/introducao-ao-mongoose-para-mongodb/>

MongoDB Atlas. Disponível em: <https://www.mongodb.com/pt-br/lp/cloud/atlas/>

Tutorial de CRUD com Mongoose. Disponível em:
<https://medium.com/@skhans/how-to-build-a-basic-node-js-crud-app-with-mongoose-and-mongodb-3e958a36001d>

Entendendo o MongoDB: Uma Introdução aos Bancos de Dados NoSQL. Disponível em:
https://medium.com/@nelson.miranda_40644/entendendo-o-mongodb-uma-introdução-aos-bancos-de-dados-nosql-c5850dd6cf6b