

Como visto na aula

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto no vídeo para poder continuar com o próximo capítulo.

Dica : [aqui \(https://github.com/alura-cursos/criando-uma-webapp-com-java-e-mongodb/archive/04201002dc4877707bfe54c1594cb765e9c1c27e.zip\)](https://github.com/alura-cursos/criando-uma-webapp-com-java-e-mongodb/archive/04201002dc4877707bfe54c1594cb765e9c1c27e.zip) você pode baixar o projeto inicial para começar o curso.

Segue cada passo dessa aula. A primeira coisa a se fazer é subir o banco de dados Mongo. Para isso use o comando

```
./mongod --dbpath /Users/lazaro/Documents/Alura/mongodb/db
```

Depois dele iniciado, podemos dar início ao nosso projeto.

Para isso, crie um projeto novo no eclipse usando o Maven.

Com o projeto já criado, adicione a dependência do Mongo no `pom.xml` :

```
<dependencies>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongodb-driver</artifactId>
    <version>3.4.2</version>
  </dependency>
</dependencies>
```

Agora seu projeto já está pronto para se conectar no Mongo. Crie uma classe e um método `main` nessa classe:

```
public class Main {

    public static void main(String[] args) {

    }

}
```

Agora para se conectar com o Mongo, basta criar uma instância da classe `MongoClient` :

```
MongoClient cliente = new MongoClient();
```

Com o cliente do Mongo já instanciado, basta agora pegar o banco de dados que você desejar.

```
MongoDatabase bancoDeDados = cliente.getDatabase("test");
```

A partir do banco de dados que você obteve, já é possível pegar uma `Collection` :

```
MongoCollection<Document> alunos = bancoDeDados.getCollection("alunos");
```

Já isto, podemos buscar um documento lá no Mongo:

```
Document primeiroAluno = alunos.find().first();
```

Para ver o aluno que foi achado lá no banco, podemos usar o `System.out.println` :

```
System.out.println(primeiroAluno);
```

Prontinho, agora podemos fechar a conexão com o banco:

```
cliente.close();
```

Além de conectar no banco e buscar documento, você viu que podemos fazer as operações de CRUD (Create, Read, Update, Delete):

- Criar documento

```
Document novoAluno = new Document("nome", "Celina Prates")
    .append("data_nascimento", new Date(2009, 2, 9))
    .append("notas", Arrays.asList(10, 9, 8.5))
    .append("curso", new Document("nome", "História"))
    .append("habilidades", Arrays.asList(
        new Document("nome", "ingles").append("nível", "Avançado"),
        new Document("nome", "Espanhol").append("nível", "básico")));
alunos.insertOne(novoAluno);
```

- Atualizar um documento

```
alunos.updateOne(
    Filters.eq("nome", "Celina Prates"),
    new Document("$set", new Document("nome", "Celina Regina Salomão Prates")));
```

- Deletar um documento

```
alunos.deleteOne(Filters.eq("nome", "Celina Regina Salomão Prates"));
```