



**Certified Tech
Developer**
The Ultimate Degree

MongoDB

O MongoDB é um sistema de banco de dados NoSQL, open source, orientado a documentos.

É um sistema de gerenciamento de banco de dados de documentos projetado para persistência de dados de alto desempenho e alta disponibilidade.

Nele os dados são armazenados em uma representação binária chamada de BSON (Binary JSON). A codificação BSON estende a representação popular do JSON (JavaScript Object Notation) para incluir tipos adicionais, como: int, long, date, etc.

Os documentos BSON contêm um ou mais campos, cada um contendo um valor de um tipo de dado específico, incluindo matrizes, dados binários e subdocumentos.

Esses documentos MongoDB BSON estão fortemente alinhados com a estrutura de objetos na linguagem de programação.

A seguir veremos suas características.

MongoDB vs Banco de Dados Relacional

Em vez de usar tabelas e linhas, como nos bancos de dados relacionais, aqui utilizamos coleções e documentos. Uma base de dados é um espaço onde se armazena uma coleção.

Veremos isso com mais detalhes no gráfico a seguir.

Neste exemplo, você verá a diferença sobre como persistir os dados em um banco de dados relacional e no MongoDB.

Anotações Spring Data MongoDB

@Document: usada em classes, informa ao MongoDriver que ele pode considerar objetos dessa classe como Documents MongoDB em uma coleção em particular. O nome da coleção é deduzido do nome da classe, mas se eles tiverem nomes diferentes, é necessário adicionar o nome da coleção a ser associada. Sendo semelhante à anotação @Entity usada no JPA.

@Field: instrui o framework como persistir os atributos da classe no MongoDB. Podemos usar o atributo "name" para especificar o nome que este campo possui no Document MongoDB.

```
@Document(collection = "airplanes")
public class Aircraft {

    private String id;

    private String model;

    @Field(name="seats") private int nbSeats;

}
```

@Id: todo document em uma collection MongoDB deve ter um campo id. É único e está indexado. Sendo semelhante a uma chave primária no banco de dados relacional.

@Transient: exclui um atributo para que não seja persistido.

```
@Document
public class Aircraft {

    @Id private String id;

    private String model;
```

```
@Transient private int nbSeats;

}
```

@Indexed: se queremos executar consultas com filtros mais rapidamente. Como por exemplo, vamos fazer muitas queries sobre um field model, é bom indexá-lo. Pode-se especificar a direção do índice e se é único.

```
@Document
public class Aircraft {

    @Id private String id;

    @Indexed(direction = IndexDirection.ASCENDING, unique = false)

    private String model;

    private int nbSeats;
}
```

@DbRef: atua de forma semelhante a um join de um banco de dados relacional. Um documento é associado a outro de uma Collection diferente.

```
@Document
public class Manufacturer {

    @Id private String id;

    private String name;
```

```
}
```

```
@Document  
public class Aircraft {  
  
    @Id private String id;  
  
    private String model;  
  
    @DbRef  
    private Manufacturer man;  
}
```

Guia de instalação

Antes de começar a criar um banco de dados no MongoDB, disponibilizamos dois links importantes para acessar e realizar o download e poder trabalhar.

1. Instalação do MongoDB: a forma e a localização do instalador dependerá do sistema operacional. [Install MongoDB — MongoDB Manual](#)
2. Instalação do Robo 3T: é uma interface gráfica para o MongoDB. [Robomongo](#)

A seguir, fornecemos os guias de instalação de acordo com o sistema operacional que você possui.

