

Banco de
Dados



MongoDB



MongoDB Atlas.

A plataforma de dados multi-cloud.
Um pacote integrado de banco de dados em nuvem com serviços para agilizar e simplificar a maneira como você cria com dados.



Objetivo

Criar mais rapidamente com um conjunto de serviços de dados integrados. Ele visa atender a uma variedade de casos de uso, permitindo que desenvolvedores criem aplicações ágeis, que possam se adaptar rapidamente a mudanças nos requisitos.



Características do MongoDB

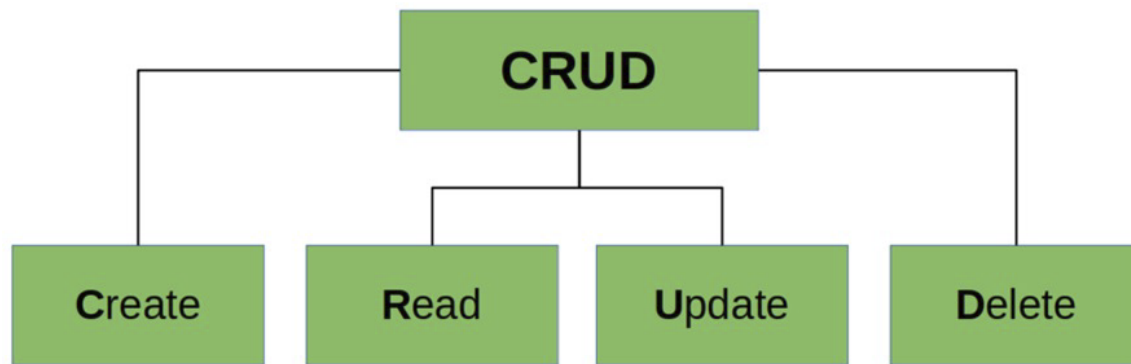
Modelo de Documento Flexível: Utiliza um formato semelhante ao JSON, permitindo um mapeamento direto para objetos no código, o que facilita o desenvolvimento e a modelagem para diversos casos de uso.

Gerenciamento Simplificado de Dados: A API de consulta unificada proporciona uma maneira intuitiva de trabalhar com dados, suportando pesquisas de texto, análises em tempo real e experiências orientadas a eventos, enquanto mantém uma arquitetura de dados simplificada.

Escalabilidade e Segurança: Oferece uma base confiável com padrões robustos de segurança e recuperação de dados. O MongoDB Atlas permite executar aplicações globalmente, com suporte multinuvem, garantindo segurança e confiança no dimensionamento de cargas de trabalho.

O que é crud?

- Create (Criar): Inserir novos documentos (registros) no banco.
- Read (Ler): Buscar e visualizar documentos no banco.
- Update (Atualizar): Modificar documentos existentes no banco.
- Delete (Excluir): Remover documentos do banco.



Atlas

Lucas's Org ...

Access Manager

Billing

All Clusters

Get Help

Lucas

adotaeu

Data Services

Charts

Overview

DEPLOYMENT

Database

SERVICES

Search

Vector Search

Stream Processing

Triggers

Migration

Data Federation

Programmatic Access

Data API

SECURITY

Backup

Database Access

Network Access

Advanced

Overview

Real Time

Metrics

Collections

Atlas Search

Performance Advisor

Online Archive

Programmatic Access

DATABASES: 1

COLLECTIONS: 1

VISUALIZE YOUR DATA

REFRESH

+ Create Database

Search Namespaces

adotaeu

pets

adotaeu

LOGICAL DATA SIZE: 424B

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
pets	5	424B	85B	36KB	1	36KB	36KB

localhost:3000/pets

Apps

Google

Email - lucas ferreir...

Gastos pessoais.xlsx...

CHATGPT

Chatgot

Meu Curso | Portal...

Home.js - nodebox...

TV.com.VC

Intensa Eventos - P...

Início | Disney+

Todos os favoritos

Estilos de formatação

[{"_id": "6646813c9e8c1e529eb6fe33", "titulo": "CAT", "veterinario": "carlos", "petSister": "Lucas"}, {"_id": "6646836e9e8c1e529eb6fe39", "titulo": "lucas", "veterinario": "lucas", "petSister": "lucas"}, {"_id": "6648bd71b91bda0897d80670", "titulo": "burro", "veterinario": "lucas", "petSister": "igor"}, {"_id": "6648bf5cb91bda0897d80675", "titulo": "macaco", "veterinario": "lucas", "petSister": "igor"}, {"_id": "66f340d9b2dc3b835ceea78e", "titulo": "Gato", "veterinario": "Lucas", "petSister": "Luciano"}]



GET http://localhost:3000/pets

HTTP http://localhost:3000/pets Save Share

GET http://localhost:3000/pets Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results 200 OK • 219 ms • 744 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "_id": "6646813c9e8c1e529eb6fe33",
4     "titulo": "CAT",
5     "veterinario": "carlos",
6     "petSistter": "Lucas"
7   },
8   {
9     "_id": "6646836e9e8c1e529eb6fe39",
10    "titulo": "lucas",
11    "veterinario": "lucas",
12    "petSistter": "lucas"
13  },
14  {
15    "_id": "6648bd71b91bda0897d80670",
16    "titulo": "burro",
17    "veterinario": "lucas",
18    "petSistter": "igor"
19  },
20  {
21    "_id": "6648bf5cb91bda0897d80675",
22    "titulo": "macaco",
23    "veterinario": "lucas",
```



HTTP **http://localhost:3000/pets** Save Share

POST ▼ **http://localhost:3000/pets** Send ▼

Params Authorization Headers (9) **Body** ● Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Beautify

```
1 {
2
3   "titulo": "Papagaio",
4   "veterinario": "Virgulino",
5   "petSister": "Jadson"
6 }
```

Body Cookies Headers (8) Test Results 201 Created • 203 ms • 424 B • 🌐 📄 Save Response ⋮

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 {
2   "message": "Pet cadastrado com sucesso!",
3   "pets": {
4     "titulo": "Papagaio",
5     "veterinario": "Virgulino",
6     "petSister": "Jadson",
7     "_id": "66f345f0f538d1b96ba59e9d"
8   }
9 }
```



HTTP <http://localhost:3000/pets/66f345f0f538d1b96ba59e9d> Save Share

DELETE ▼ <http://localhost:3000/pets/66f345f0f538d1b96ba59e9d> Send ▼

Params Authorization Headers (7) **Body** Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

1

Body Cookies Headers (8) Test Results 200 OK • 144 ms • 307 B • Save Response ⋮

Pretty Raw Preview Visualize JSON ▼

```
1 {  
2   "message": "Pet excluído com sucesso!"  
3 }
```




Procura após Delete

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:3000/pets/66f345f0f538d1b96ba59e9d`
- Method:** `GET`
- Body:** Empty
- Response:** `404 Not Found` (132 ms, 307 B)
- Response Body (JSON):**

```
{
  "message": "Pet não encontrado"
}
```