Aula 14 : Mangusto

# Clientes de banco de dados

**Clientes de banco de dados**

Toda vez que acessamos um banco de dados para realizar qualquer operação CRUD, nos tornamos clientes desse banco de dados.

Podemos ser diferentes tipos de clientes para acessar o mesmo banco de dados:

* Cliente CLI
* cliente GUI
* cliente da web
* aplicativo cliente

Já atuamos como clientes CLI, vamos visualizar brevemente os outros tipos de clientes.

**Clientes GUI:**

É o cliente que pode se conectar ao banco de dados a partir de um programa gráfico desenhado para ele. O cliente GUI por excelência para trabalhar com o MongoDB é **o MongoDB Compass** .

Isso geralmente é instalado quando instalamos o Mongo, mas se você não o tiver, [poderá baixá-lo](https://www.mongodb.com/try/download/compass) .

**Clientes de aplicativos:**

Este será o nosso forte: poder acessar o banco de dados de nosso aplicativo nos permite usá-lo a partir do código, a partir de um contexto.

Isso significa que, uma vez que o aplicativo esteja configurado corretamente, não dependeremos tanto da entrada manual, pois nosso programa saberá quando realizar as operações CRUD

**Clientes Web:**

Quando conseguimos ter nosso banco de dados na nuvem, podemos nos conectar a um servidor na web para poder analisar os dados e realizar operações desde qualquer computador.

É um dos modelos mais utilizados, pois não é necessário ter um computador específico (dependendo das licenças)

O cliente web por excelência para o MongoDB é o Atlas, além de atuar como **DBaaS**

DBaaS: banco de dados como serviço

**O problema: escalabilidade e viabilidade**

* O que acontece quando sua empresa está crescendo e precisamos armazenar uma grande quantidade de informações?
* Estamos dispostos a dedicar salas inteiras com bancos de dados?
* chegar os custos de ter que comprar **infra-estrutura física** para o negócio ? Valera a pena?
* E se houvesse uma forma de "alugar" aquele espaço para um provedor, para se preocupar apenas com os aspectos mais superficiais do banco de dados?

**A solução: DBaaS**

Usar um banco de dados como serviço implica poder usar um banco de dados, **sem se preocupar em ter que gerenciar todos os aspectos físicos que isso implica** , ou seja, podemos alugar espaços para hospedar as informações em nosso banco de dados.

Todos os aspectos físicos, de manutenção e segurança dos referidos bancos de dados são gerenciados pelo provedor do referido serviço.

Amazon, Google, Microsoft, **Mongo Atlas** são alguns exemplos.

**Vantagem do modelo DBaaS**

* **A infraestrutura física sai** da equação, economizando custos, já que o provedor é responsável pela manutenção e disponibilidade dos sistemas. Os usuários são responsáveis por seus próprios dados.
* **Economia geral de custos** . Além de dispensar investimentos físicos, com **o DBaaS** você pode ter menos funcionários dedicados a essa tarefa, economizar energia e aproveitar melhor o espaço físico.
* **escalabilidade** . Com **o DBaaS** podemos acessar diferentes taxas baseadas principalmente no desempenho desejado e em nossas necessidades.
* **Pessoal qualificado** . Através **do DBaaS** você tem acesso a especialistas em banco de dados que se encarregarão de todas as tarefas de manutenção, atualização, segurança e gerenciamento.

MongoDB AtlasName

## Atlas do MongoDB: destaques dos recursos

* **Automação** – Uma maneira fácil de criar, iniciar e dimensionar aplicativos no MongoDB.
* **Flexibilidade** : DBaaS com tudo o que é necessário para aplicações modernas.
* **Segurança** – Vários níveis de segurança disponíveis.
* **Escalabilidade** – Alta escalabilidade sem interrupção dos negócios.
* **Alta disponibilidade** – implantações tolerantes a falhas e autocorretivas por padrão.
* **Alto desempenho** – O desempenho necessário para cargas de trabalho exigentes.

## Vantagens:

**Execução**

* Inicie um cluster em segundos.
* Implantações replicadas e sem interrupções.
* Escalabilidade total: dimensione horizontalmente ou verticalmente sem interromper a atividade.
* Revisões automáticas e atualizações simplificadas.

**Proteção e segurança**

* Autenticação e criptografia.
* Backups contínuos com recuperação temporária.
* Monitoramento detalhado e alertas personalizados.

mangusto

Mongoose é um ODM (Object Document Mapping), que nos permitirá definir esquemas para gerenciar coleções e documentos entre um aplicativo nodejs e um banco de dados MongoDB

Uma vez que entendemos que o MongoDB permite gerenciar documentos semelhantes a um objeto. Além disso, o mangusto nos permitirá conectar ao banco de dados gerenciado do Mongo Atlas, para poder manter o gerenciamento do banco de dados na nuvem.