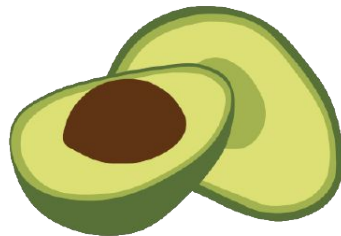


ArangoDB

Suas principais vantagens são: consolidação, performance simplificada, redução de complexidade, tolerância a falhas e velocity pack.



Enterprise

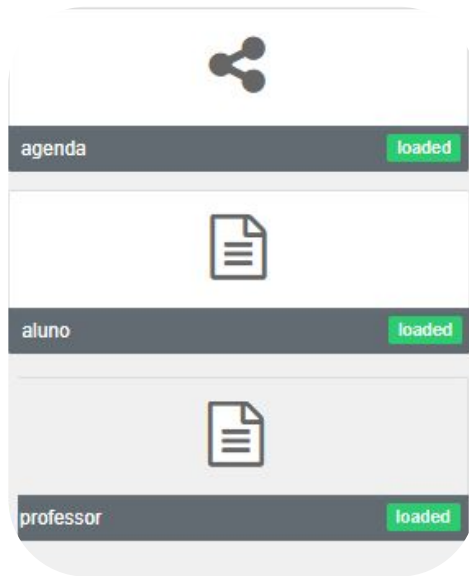
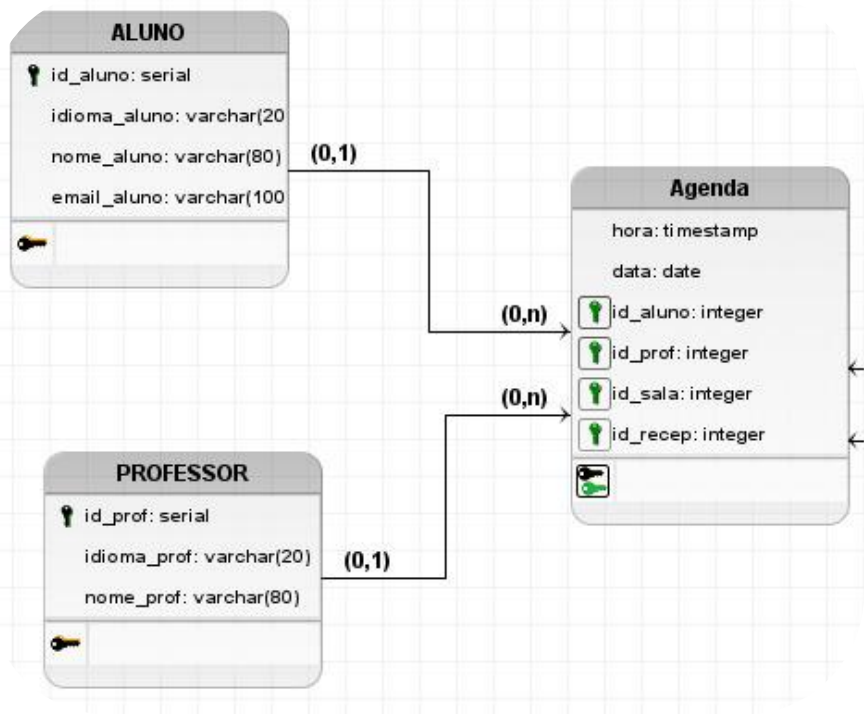
1. Alta segurança
2. Replicação
Datacenter-Datacenter
3. Recursos exclusivos de
scale-out
4. Livre para avaliação
5. Integração LDAP

Oasis

1. Totalmente gerenciado
2. Backups
automatizados e de
segurança nível
corporativo
3. Escalabilidade elástica
4. Teste grátis por duas
semanas

Community

1. Fornecem um pacote
contendo o servidor e
as ferramentas do
cliente (arangosh,
arangodump)
2. Gratuito
3. Replicação
desabilitada
4. Possui interface web



Modelo Lógico vs ArangoDB

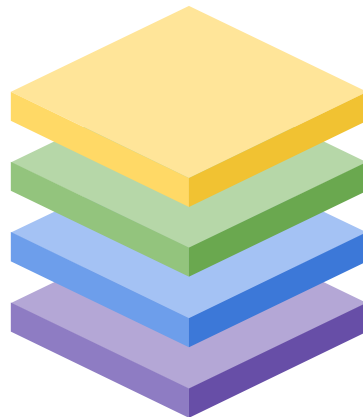
Conexão com o Banco

Sintaxe de Conexão

```
arangosh> db._createDatabase("exemplo");  
arangosh> var users = require("@arangodb/users");  
arangosh> users.save("root@exemplo", "senha");  
arangosh> users.grantDatabase("root@exemplo", "exemplo");  
shell> arangosh --server.username "root@exemplo" --server.database exemplo
```

Exemplo de Conexão com o Banco

```
arangosh> db._createDatabase("_system");  
arangosh> var users = require("@arangodb/users");  
arangosh> users.save("root@_system", "1234");  
arangosh> users.grantDatabase("root@_system", "_system");  
shell> arangosh --server.username "root@_system" --server.database _system
```



Insert nas Tabelas no ArangoDB

Sintaxe de Inserção

```
For x in [  
{"_nome":"Luis Felipe","_email":"luis@gmail.com"}}] insert x in teste1
```

Exemplo de Inserção

```
For alu in [  
{"_nome_aluno":"Carol","_email_aluno":"carol@gmail.com","_idioma_aluno":"inglês"},  
] insert alu in aluno
```

```
For prof in [  
{"_nome_prof": "Jorge", "_idioma_prof": "inglês"}] insert prof in Professor
```

```
For ag in [  
{"_hora": "15:00:00", "_data": "2020-07-30", "_from": "aluno/10871", "_to": "professor/6847"}] insert ag in agenda
```

Select nas Tabelas no ArangoDB

Sintaxe de Seleção

```
FOR x IN teste1
  FOR y IN teste2
    FILTER y.key == x.key
  RETURN { x, y }
```

Exemplo de Seleção

```
for alu in aluno
  for prof in professor
    for ag in agenda
      filter ag.from == alu.id
      filter ag.to == prof.id

return distinct {
  Aluno: alu,
  Professor: prof,
  Agenda: ag
}
```

Exclusão nas Tabelas no ArangoDB

Sintaxe de Exclusão

```
LET doc = DOCUMENT('teste1/0000')  
REMOVE doc IN teste1
```

Exemplo de Exclusão

```
LET doc = DOCUMENT('professor/7045')  
REMOVE doc IN professor
```

```
LET doc = DOCUMENT('professor/6939')  
REMOVE doc IN professor
```

Atualização nas Tabelas no ArangoDB

Sintaxe de Exclusão

```
LET key = PARSE_IDENTIFIER("teste1/0000").key  
UPDATE key WITH { _nome: "Luis Felipe Avelar" } IN teste1
```

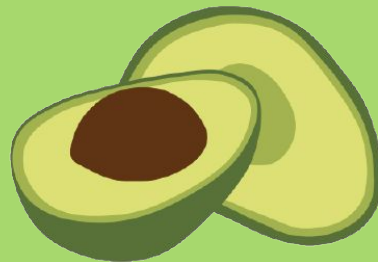
Exemplo de Exclusão

```
LET key = PARSE_IDENTIFIER("professor/6847").key  
UPDATE key WITH { _idioma_prof: "francês" } IN professor
```

```
LET key = PARSE_IDENTIFIER("aluno/10871").key  
UPDATE key WITH { _idioma_aluno: "francês" } IN aluno
```

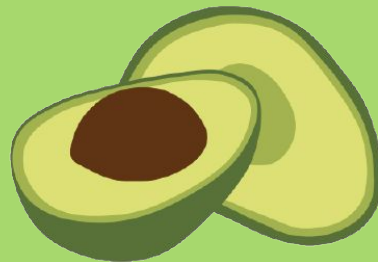
Pontos Positivos

1. Linguagem AQL é simples de entender.
2. Documentação oficial é boa e fácil de encontrar.
3. Por ser um pouco conhecido, tem vários vídeos no Youtube de tutoriais.
4. Por ter a versão community fica melhor de testar a aplicação antes de adquirir a versão paga.



Pontos Negativos

1. Quando precisa fazer joins com várias tabelas, não é muito bom, vários valores vêm repetidos e é difícil visualizá-los.



Material de Referência

<https://www.arangodb.com/docs/stable/>

<https://www.arangodb.com/docs/stable/aql/>

<https://youtu.be/4C4zqhXwCKs>

