

CARACTERÍSTICAS

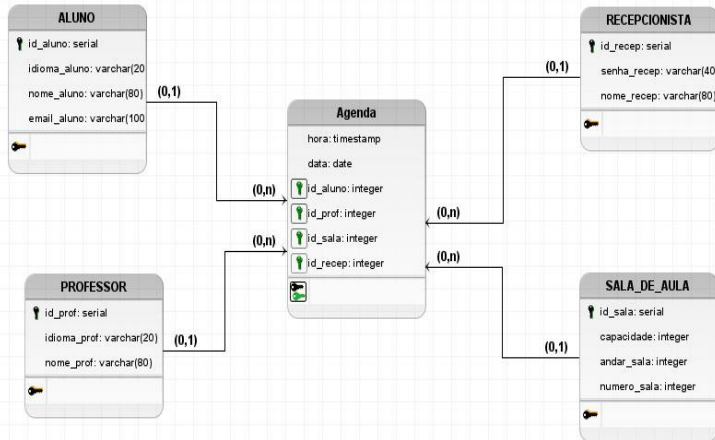
- Versão gratuita escalável de preço
 - Limitação em RAM e Armazenamento
 - Multi-regiões para grandes empresas
-
- Servidor sempre ativo
 - Várias linguagens de programação



REPRESENTAÇÕES DE MODELOS



Postgres



Mongo

```
[ ] from datetime import date, time
import datetime
from backports.datetime_fromisoformat import MonkeyPatch
MonkeyPatch.patch_fromisoformat()

bd.agenda.insert_one({"hora": datetime.datetime.now(), "id_aluno": 1, "id_prof": 10101, "id_sala": 202, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now(), "id_aluno": 4, "id_prof": 16161, "id_sala": 101, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now(), "id_aluno": 6, "id_prof": 15151, "id_sala": 105, "id_recep": 2020})
bd.agenda.insert_one({"hora": datetime.datetime.now(), "id_aluno": 10, "id_prof": 20202, "id_sala": 101, "id_recep": 3030})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=1), "id_aluno": 2, "id_prof": 10101, "id_sala": 103, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=1), "id_aluno": 3, "id_prof": 10101, "id_sala": 101, "id_recep": 4040})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=1), "id_aluno": 7, "id_prof": 15151, "id_sala": 205, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=1), "id_aluno": 8, "id_prof": 20202, "id_sala": 101, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=2), "id_aluno": 9, "id_prof": 19191, "id_sala": 104, "id_recep": 1010})
bd.agenda.insert_one({"hora": datetime.datetime.now() + datetime.timedelta(hours=2), "id_aluno": 5, "id_prof": 18181, "id_sala": 201, "id_recep": 4040})
```

```
[ ] bd.aluno.insert_one({"id_aluno": 1, "nome_aluno": "Ana", "email_aluno": "carol@gmail.com", "idioma_aluno": "ingles"})
bd.aluno.insert_one({"id_aluno": 2, "nome_aluno": "Carol", "email_aluno": "carol@gmail.com", "idioma_aluno": "ingles"})
bd.aluno.insert_one({"id_aluno": 3, "nome_aluno": "Bruna", "email_aluno": "bruna@gmail.com", "idioma_aluno": "ingles"})
bd.aluno.insert_one({"id_aluno": 4, "nome_aluno": "Alex", "email_aluno": "alex@gmail.com", "idioma_aluno": "espanhol"})
bd.aluno.insert_one({"id_aluno": 5, "nome_aluno": "Fernando", "email_aluno": "fernando@gmail.com", "idioma_aluno": "espanhol"})
bd.aluno.insert_one({"id_aluno": 6, "nome_aluno": "Mateus", "email_aluno": "mateus@gmail.com", "idioma_aluno": "ingles"})
bd.aluno.insert_one({"id_aluno": 7, "nome_aluno": "Camila", "email_aluno": "camila@gmail.com", "idioma_aluno": "ingles"})
bd.aluno.insert_one({"id_aluno": 8, "nome_aluno": "Lucas", "email_aluno": "lucas@gmail.com", "idioma_aluno": "francês"})
bd.aluno.insert_one({"id_aluno": 9, "nome_aluno": "Arnaldo", "email_aluno": "arnaldo@gmail.com", "idioma_aluno": "francês"})
bd.aluno.insert_one({"id_aluno": 10, "nome_aluno": "Charles", "email_aluno": "charles@gmail.com", "idioma_aluno": "francês"})
```

```
[ ] bd.recepcionista.insert_one({"id_recep": 1010, "nome_recep": "Gloria", "senha_recep": "senha1"})
bd.recepcionista.insert_one({"id_recep": 2020, "nome_recep": "Debora", "senha_recep": "senha2"})
bd.recepcionista.insert_one({"id_recep": 3030, "nome_recep": "Larissa", "senha_recep": "senha3"})
bd.recepcionista.insert_one({"id_recep": 4040, "nome_recep": "Geovanna", "senha_recep": "senha4"})
```

CONEXÃO VIA PYMONGO

É necessária a instalação das bibliotecas pymongo, dnspython e backports para datetime.

```
[ ] pip install pymongo
```

```
[ ] pip install dnspython
```

```
[ ] !pip install backports-datetime-fromisoformat
```

```
[ ] from pymongo import MongoClient
```

```
client = MongoClient("mongodb+srv://root:1505@cluster0.g4qte.gcp.mongodb.net/BD1?retryWrites=true&w=majority")  
bd = client.test
```

INCLUSÃO DE DADOS

```
[ ] bd.professor.insert_one({"id_prof": 10101, "nome_prof": "Jorge", "idioma_prof": "ingles"})
    bd.professor.insert_one({"id_prof": 12121, "nome_prof": "Luiz", "idioma_prof": "ingles"})
    bd.professor.insert_one({"id_prof": 13131, "nome_prof": "Vitoria", "idioma_prof": "ingles"})
    bd.professor.insert_one({"id_prof": 14141, "nome_prof": "Ingrid", "idioma_prof": "ingles"})
    bd.professor.insert_one({"id_prof": 15151, "nome_prof": "Maristela", "idioma_prof": "ingles"})
    bd.professor.insert_one({"id_prof": 16161, "nome_prof": "Carla", "idioma_prof": "espanhol"})
    bd.professor.insert_one({"id_prof": 17171, "nome_prof": "Brenda", "idioma_prof": "espanhol"})
    bd.professor.insert_one({"id_prof": 18181, "nome_prof": "Andre", "idioma_prof": "espanhol"})
    bd.professor.insert_one({"id_prof": 19191, "nome_prof": "Luna", "idioma_prof": "frances"})
    bd.professor.insert_one({"id_prof": 20202, "nome_prof": "Stella", "idioma_prof": "frances"})
```

```
[ ] list(bd.aluno.find())
```

```
[ ] list(bd.recepcionista.find())
```

```
[ ] list(bd.professor.find())
```

```
[ ] list(bd.sala de aula.find())
```

```
[ ] list(bd.agenda.find())
```

```
19) list(bd.agenda.find({})

[{'_id': ObjectId('5f79e0a4b2719d45a246dc7a'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 4, 6470000),
  'id_aluno': 1,
  'id_prof': 10101,
  'id_recep': 1010,
  'id_sala': 202},
 {'_id': ObjectId('5f79e0a4b2719d45a246dc7b'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 4, 7930000),
  'id_aluno': 4,
  'id_prof': 16161,
  'id_recep': 1010,
  'id_sala': 101},
 {'_id': ObjectId('5f79e0a4b2719d45a246dc7c'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 4, 9160000),
  'id_aluno': 6,
  'id_prof': 15151,
  'id_recep': 2020,
  'id_sala': 105},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc7d'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 380000),
  'id_aluno': 10,
  'id_prof': 20202,
  'id_recep': 3030,
  'id_sala': 101},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc7e'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 1600000),
  'id_aluno': 2,
  'id_prof': 10101,
  'id_recep': 1010,
  'id_sala': 103},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc7f'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 2820000),
  'id_aluno': 4,
  'id_prof': 10101,
  'id_recep': 4040,
  'id_sala': 101},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc80'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 4050000),
  'id_aluno': 7,
  'id_prof': 15151,
  'id_recep': 1010,
  'id_sala': 208},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc81'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 5280000),
  'id_aluno': 8,
  'id_prof': 20202,
  'id_recep': 1010,
  'id_sala': 101},
 {'_id': ObjectId('5f79e0a5b2719d45a246dc82'),
  'hora': datetime.datetime(2020, 10, 4, 14, 48, 5, 6500000),
  'id_aluno': 9,
  'id_prof': 19191,
  'id_recep': 1010,
```

```
{5} list(bd.aluno.find())
```

```
[{'_id': ObjectId('5f79d694b2719d45a246dc52'),
  'email_aluno': 'carol@gmail.com',
  'id_aluno': 1,
  'idioma_aluno': 'ingles',
  'nome_aluno': 'Ana'},
 {'_id': ObjectId('5f79d696b2719d45a246dc53'),
  'email_aluno': 'carol@gmail.com',
  'id_aluno': 2,
  'idioma_aluno': 'ingles',
  'nome_aluno': 'Carol'},
 {'_id': ObjectId('5f79d696b2719d45a246dc54'),
  'email_aluno': 'bruna@gmail.com',
  'id_aluno': 3,
  'idioma_aluno': 'ingles',
  'nome_aluno': 'Bruna'},
 {'_id': ObjectId('5f79d696b2719d45a246dc55'),
  'email_aluno': 'alex@gmail.com',
  'id_aluno': 4,
  'idioma_aluno': 'espanhol',
  'nome_aluno': 'Alex'},
 {'_id': ObjectId('5f79d696b2719d45a246dc56'),
  'email_aluno': 'fernando@gmail.com',
  'id_aluno': 5,
  'idioma_aluno': 'espanhol',
  'nome_aluno': 'Fernando'},
 {'_id': ObjectId('5f79d696b2719d45a246dc57'),
  'email_aluno': 'mateus@gmail.com',
  'id_aluno': 6,
  'idioma_aluno': 'ingles',
  'nome_aluno': 'Mateus'},
 {'_id': ObjectId('5f79d696b2719d45a246dc58'),
  'email_aluno': 'camila@gmail.com',
  'id_aluno': 7,
  'idioma_aluno': 'ingles',
  'nome_aluno': 'Camila'},
 {'_id': ObjectId('5f79d696b2719d45a246dc59'),
  'email_aluno': 'lucas@gmail.com',
  'id_aluno': 8,
  'idioma_aluno': 'francês',
  'nome_aluno': 'Lucas'},
 {'_id': ObjectId('5f79d696b2719d45a246dc5a'),
  'email_aluno': 'arnaldo@gmail.com',
  'id_aluno': 9,
  'idioma_aluno': 'francês',
  'nome_aluno': 'Arnaldo'},
 {'_id': ObjectId('5f79d697b2719d45a246dc5b'),
  'email_aluno': 'charles@gmail.com',
  'id_aluno': 10,
  'idioma_aluno': 'francês',
  'nome_aluno': 'Charles'}]
```

EXCLUSÃO DE DADOS

```
[ ] bd.aluno.delete_one({"nome_aluno": "Ana"})  
    bd.professor.delete_one({"nome_prof": "Ingrid"})
```

ATUALIZAÇÃO DE DADOS

```
[ ] bd.sala_de_aula.update_many({"capacidade":5}, {"$inc":{"capacidade":10}})
    bd.agenda.update_many({"id_prof":10101}, {"$inc":{"id_prof":12121}})
```

PONTOS POSITIVOS

- Escalabilidade de recursos
- Trabalho em conjunto
- Estável e sempre ativo
- Várias linguagens disponíveis



PONTOS NEGATIVOS

- Curva de aprendizado inicial é baixa
- Não é possível implementar chaves estrangeiras
- Poucos recursos para contas gratuitas
- Duplicata de documentos

Referências

Documentação PyMongo

<https://pymongo.readthedocs.io/en/stable/>

Documentação MongoDB

<https://docs.mongodb.com/>

Stack Overflow

<https://stackoverflow.com>