Criação do Cluster Servidor <https://cloud.mongodb.com/>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Modelagem <https://jsonformatter.curiousconcept.com/>

Modelando USUARIOS

{

"\_id": 1,

"nome": "Carlos Lopes",

"idade": 32,

"dt\_nascimento": "1992-10-05",

"endereco": "Rua teste, 100",

"enderecos": {

"logradouro": "Rua teste",

"numero": 100,

"bairro": "vila maria",

"cidade": "São Paulo"

}

}

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Site

Descrição gerada automaticamente

Colocamos arrays de interesses e reservas

Interface gráfica do usuário, Texto, Aplicativo, Site

Descrição gerada automaticamente

Caso fosse preciso ter vários endereço para o mesmo usuário poderia fazer dessas forma

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Modelando DESTINO

{

"\_id": 1,

"nome": "Parque Ibirapuera",

"descicao": "Principal Parque de SP",

"localizacao": {

"type": 'Point',

"coordinates": []

}

}

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Modelando RESERVAS

{

"\_id": ObjectId("123"),

"destino": ObjectId("456"),

"data": "2024-10-05",

"status": "pendente",

"usuario" ObjectId(345)

}

Texto

Descrição gerada automaticamente

CRIANDO BANCO DE DADOS

Usamos o atlas

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

CRIANDO UMA COLLECTION

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

INSERINDO DOCUMENTO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

CONSULTANDO DOCUMENTOS

db.usuarios.find({}); Traz a quantidade de registros que ele encontrar

db.usuarios.findOne({}); Traz a primeiro referencia

db.usuarios.findOneAndUpdate({}, {}); Primeira a busca depois atualiza e traz a atualizada

db.usuarios.findOneAndDelete({}); Primeiro busca depois deleta e traz o dado deletado