

Modelagem Inicial do Banco de Dados Não Relacional

Documentação referente ao modelo inicial do banco de dados

- O sistema:

Sistema se baseia em arrecadação de alimentos para distribuição a locais para população necessitada. Plataforma administrativa para arrecadação e distribuição de alimentos.

- A modelagem:

O sistema contém cadastro de Empresas e pessoas participantes do projeto, cada empresa pode ter suas próprias características, podendo ter atributos variados dentro do documento do banco correspondente ao seu cadastro.

Cada documento é independente.

Ex:

```
{  
  
  "Nome Empresa" : "Nome"  
  
  "cnpj" : 0000.0000/0000-0  
  
  "ramo" : "hortifruti"  
  
  "certificacao": "nome certificação"  
  
}
```

Os produtos também poderão ser doados por pessoas físicas:

```
{  
  
  "Nome pessoa":  
  
  "cpf": "000.000.000-00"
```

“dtNasc”: “00-00-0000”

“idade”: “00”

“cep”: 00000-00

“rua”: “Nome rua”

“bairro”: “nome bairro”

“cidade”: “nome cidade”

} Obs -> Cep obtido por meio da API ViaCEP para preenchimento dos dados de endereço

Uma empresa pode cadastrar a quantidade desejada de produtos a serem doados, de maneira que cada produto pode ter características variadas, como validade, data produção, validade embalagem, tipo de alimento, entre outros.

Cada produto cadastrado será um documento próprio e terá o cnpj da empresa cadastrada ou cpf da pessoa que doará o alimento

Ex:

```
{  
    "idProduto" : 00  
    "dataCadastro": 00-00-0000  
    "produto" : "nome produto",  
    "codBarra": 0000000000,  
    "dtProducao": 00-00-0000(opcional)  
    "marca": "nome marca"(opcional),  
    "validade": 00-00-0000 (se possuir),  
    "quantidade": 00,
```

```

        "tipo": "tipo do alimento",

        "cnpj": "cnpj da empresa doadora",

        "cpf":      "cpf      da      pessoa      doadora"
    }

```

Um relatório é gerado sempre que solicitado no sistema. Os relatórios servirão como documentos de Análise de Qualidade no funcionamento das doações e distribuição dos alimentos. O relatório poderá ser gerado por: Busca por Data, Busca por CNPJ, Busca por CPF, Busca por Nome Produto.

Cada relatório exibirá uma lista de resultados, contendo os produtos doados, as empresas ou pessoas doadoras, para onde serão redirecionados os produtos e a data de cadastro desses produtos.

Um relatório de produtos com datas próximas ao vencimento também será gerado.

Ex:

```

{

    "relatorio": [

        {

            "idProduto": 00,

            "dataCadastro": 00-00-0000,

            "CNPJ": 000000/0000-00,

            "nomeProduto": "nome",

            "validade": 00-00-0000,

            "idLocalDistribuicao": "local"

        },

    ]
}

```

```

{
    "idProduto": 00,
    "dataCadastro": 00-00-0000,
    "CPF": 0000000000-00,
    "nomeProduto": "nome",
    "validade": 00-00-0000,
    "idLocalDistribuicao": "local"
},
]
}

```

Como última coleção do Banco de Dados, têm-se os documentos relacionados aos locais de distribuição dos produtos, os quais, também serão relatórios a serem exibidos por cada local que receberá o produto. Portanto, determinado local somente terá acesso ao produtos que estiverem interligados com o CEP do mesmo, acessando todas as informações do produto e de quem foi o doador dele.

Ex:

```

{
    "nomeLocal":[
        {
            "idProduto": 00,
            "CPF": 0000000000-00,
            "dataEnvio": 00-00-0000

```

```
    },  
    {  
        "idProduto": 00,  
        "CNPJ": 000000/0000-00,  
        "dataEnvio": 00-00-0000  
    },  
]  
}
```

- A consulta de dados:

As operações de consulta de dados, estarão resumidas ao relatórios dispostos no sistema, para que o sistema não tenha muitas entradas que possam dificultar a manutenção e prevenir uma grande quantidade de erros.

Sendo o administrador do Sistema, o responsável por gerir e distribuir os alimentos cadastrados para os locais de recepção conforme demanda, será possível consultar todos os produtos cadastrados em um determinado período, via busca.

O Administrador poderá também remover produtos ou transferi-los para outros locais, provendo maior liberdade na manipulação desses dados.

Cabe também ao Administrador encerrar o cadastro de pessoas e empresas, caso os mesmos solicitem.

Cada empresa ou pessoa, terá um cadastro, o qual poderão cadastrar produtos a serem doados, porém, somente terão acesso aos produtos que lançarem ao

sistema, ficando a cargo do Administrador a distribuição dos mesmos em lotes de entrega.



Será possível que excluam os produtos a serem doados desde que não estejam em situação de "confirmado" pelo sistema. Se ele estiver confirmado, não será possível excluir o cadastro ou alterar esse produto.

Haverá um botão de confirmar doação para que o usuário possa excluir ou alterar o produto antes de encerrar a solicitação.

Informações adicionadas ao DB

O banco de dados contém 5 Collections, cada uma delas contém, no mínimo, 10.000 documentos com informações que foram adicionadas por meio de um programa em Python, disponível em: <https://github.com/MarquesFabiano/Gerador-de-dados-para-mongoDB.git>.

localhost:27017 > foodshare

+ Create collection	Refresh	View 	Sort by <input type="text" value="Collection Name"/>	
empresas				
<u>Storage size:</u> 700.42 kB	<u>Documents:</u> 10 K	<u>Avg. document size:</u> 134.00 B	<u>Indexes:</u> 1	<u>Total index size:</u> 131.07 kB
loaisDistribuicao				
<u>Storage size:</u> 937.98 kB	<u>Documents:</u> 10 K	<u>Avg. document size:</u> 290.00 B	<u>Indexes:</u> 1	<u>Total index size:</u> 131.07 kB
peessoas				
<u>Storage size:</u> 1.05 MB	<u>Documents:</u> 10 K	<u>Avg. document size:</u> 207.00 B	<u>Indexes:</u> 1	<u>Total index size:</u> 131.07 kB
produtos				
<u>Storage size:</u> 1.31 MB	<u>Documents:</u> 10 K	<u>Avg. document size:</u> 281.00 B	<u>Indexes:</u> 1	<u>Total index size:</u> 131.07 kB
relatorios				
<u>Storage size:</u> 1.51 MB	<u>Documents:</u> 10 K	<u>Avg. document size:</u> 425.00 B	<u>Indexes:</u> 1	<u>Total index size:</u> 131.07 kB

Algumas consultas e os resultados:

- 1) Buscando as pessoas mais velhas que doaram:

```
db.pessoas.find().sort('dtNasc', -1).limit(10)
```

```
}
}
{
  _id: ObjectId('66735da3bc8b02e4db3a9ec5'),
  nome: 'Jacob Guerrero',
  cpf: '019.481.860-72',
  dtNasc: '1943-07-08',
  idade: 35,
  endereco: {
    cep: '81961',
    rua: 'Brown Fall',
    bairro: 'Lopez Drives',
    cidade: 'Swansonland'
  }
}
{
  _id: ObjectId('66735da3bc8b02e4db3aa480'),
  nome: 'Daniel Ramos',
  cpf: '736.675.185-01',
  dtNasc: '1943-07-08',
  idade: 54,
  endereco: {
    cep: '51024',
    rua: 'Fred Junctions',
    bairro: 'Frederick Rue',
    cidade: 'Tylerburgh'
  }
}
```

- 2) Média da idade das pessoas que doaram:

```
db.pessoas.aggregate([
  {
    '$group': {
      '_id': 'medialdade',
      'medialdade': {'$avg': '$idade'}
```

```
    }  
  }  
])
```

```
< {  
  _id: 'mediaIdade',  
  mediaIdade: 48.94660533946605  
}
```

3) Buscar os 5 ramos que mais se repetem

```
db.empresas.aggregate([  
  {  
    '$group': {  
      '_id': '$ramo',  
      'count': {'$sum': 1}  
    }  
  },  
  {  
    '$sort': {'count': -1}  
  },  
  {  
    '$limit': 5  
  }  
])
```



```

< {
  _id: 'Aeronautical engineer',
  count: 31
}
{
  _id: 'English as a second language teacher',
  count: 29
}
{
  _id: 'Surveyor, building',
  count: 28
}
{
  _id: 'Archivist',
  count: 28
}
{
  _id: 'Systems developer',
  count: 28
}

```

4) Buscar os produtos mais doados

```

db.relatorios.aggregate([
  {
    '$unwind': '$relatorio'
  },
  {
    '$group': {
      '_id': '$relatorio.nomeProduto',
      'totalDoacoes': {'$sum': 1}
    }
  },
  {
    '$sort': {'totalDoacoes': -1}
  },
  {
    '$limit': 5
  }
]

```

])

```
])
[
  {
    _id: 'leader',
    totalDoacoes: 37
  },
  {
    _id: 'strategy',
    totalDoacoes: 35
  },
  {
    _id: 'need',
    totalDoacoes: 34
  },
  {
    _id: 'site',
    totalDoacoes: 33
  },
  {
    _id: 'five',
    totalDoacoes: 32
  }
]
```

5) Buscar os maiores doadores

```
db.relatorios.aggregate([
  {
    '$unwind': '$relatorio'
  },
  {
    '$group': {
      '_id': '$relatorio.doador',
      'totalDoacoes': {'$sum': 1}
    }
  },
  {
    '$sort': {'totalDoacoes': -1}
  },
])
```

```

{
  '$limit': 5
}
])

```

```

  _id: {
    tipo: 'empresa',
    cnpj: '05.941.111/0871-23'
  },
  totalDoacoes: 1
}
{
  _id: {
    tipo: 'empresa',
    cnpj: '28.137.169/5353-04'
  },
  totalDoacoes: 1
}
{
  _id: {
    tipo: 'empresa',
    cnpj: '20.766.445/6019-02'
  },
  totalDoacoes: 1
}
{
  _id: {
    tipo: 'pessoa',
    cpf: '947.822.693-63'
  },
  totalDoacoes: 1
}

```

6) Buscar os 3 produtos mais doados

```

db.produtos.aggregate([
  {
    '$group': {
      '_id': '$produto',
      'totalDoacoes': {'$sum': 1}
    }
  }
])

```

```

    }
  },
  {
    '$sort': {'totalDoacoes': -1}
  },
  {
    '$limit': 3
  }
])

```



```

< {
  _id: 'hotel',
  totalDoacoes: 22
}
{
  _id: 'although',
  totalDoacoes: 19
}
{
  _id: 'old',
  totalDoacoes: 19
}

```

7) Buscar o produto com a maior quantidade total doada

```

db.produtos.aggregate([
  {
    '$group': {
      '_id': '$produto',
      'totalQuantidade': {'$sum': '$quantidade'}
    }
  },
  {
    '$sort': {'totalQuantidade': -1}
  },
  {
    '$limit': 1
  }
])

```

])

```
< {  
  _id: 'cost',  
  totalQuantidade: 1073  
}  
foodshare> |
```

Por que utilizamos um banco de dados não relacional?

No desenvolvimento de sistemas modernos como o Foodshare, um aplicativo fictício para compartilhamento de alimentos, a escolha do banco de dados desempenha um papel crucial na eficiência e na capacidade de escala da aplicação. Optamos por utilizar o MongoDB, um banco de dados não relacional, devido às suas características distintas que atendem às necessidades específicas do projeto.

Flexibilidade no Esquema de Dados

Uma das vantagens fundamentais do MongoDB é sua flexibilidade no esquema de dados. Ao contrário dos bancos de dados relacionais tradicionais que exigem esquemas rígidos e predefinidos, o MongoDB permite que cada documento tenha uma estrutura diferente. Isso é particularmente útil no Foodshare, onde informações sobre pessoas, empresas, produtos doados e relatórios de distribuição podem variar significativamente em termos de conteúdo e formato.

Escalabilidade Horizontal

A capacidade de escalabilidade horizontal do MongoDB é outra razão decisiva para sua escolha no Foodshare. À medida que a base de usuários e a quantidade de dados aumentam, podemos simplesmente adicionar novos nós ao cluster MongoDB para lidar com o aumento da carga de trabalho. Isso garante que o sistema permaneça responsivo e eficiente mesmo em períodos de crescimento rápido.

Modelagem de Dados Complexos e Hierárquicos

Bancos de dados não relacionais são especialmente adequados para lidar com dados complexos e hierárquicos, como os encontrados no Foodshare. Documentos aninhados e arrays de documentos facilitam a representação e a recuperação de informações detalhadas sobre doadores, produtos doados e locais de distribuição. Essa estrutura simplifica o desenvolvimento de consultas complexas e a manutenção de uma estrutura de dados coesa e intuitiva.

Exemplo de Aplicação no Foodshare

Imagine que precisamos gerenciar informações detalhadas sobre milhares de doadores e suas doações. Com o MongoDB, podemos estruturar cada documento de forma única para capturar todas as nuances e detalhes das interações de cada doador com o sistema. Além disso, podemos facilmente escalar nosso banco de dados conforme a demanda cresce, adicionando mais capacidade de armazenamento e processamento conforme necessário.

Conclusão

A decisão de adotar o MongoDB no Foodshare não apenas facilita a implementação inicial do sistema, mas também prepara o aplicativo para um crescimento sustentável no futuro. Com flexibilidade, escalabilidade e capacidade de modelar dados complexos, o MongoDB se mostra uma escolha ideal para aplicações modernas que exigem agilidade e eficiência no gerenciamento de informações.