

Estrutura da base de dados do projeto Alimenta Ação

1. Introdução

Este documento tem por finalidade especificar os detalhes técnicos da base de dados do projeto “Alimenta Ação” tais como suas características, especificações, modelagem e utilização.

2. Projeto

O projeto “Alimenta Ação” tem como objetivo unificar todas as centrais de arrecadação de alimentos e ONGs, definir objetivos e metas de arrecadação e gerenciar o cadastro de famílias que se colocaram na fila de espera para o recebimento do auxílio ou para usuários que queiram contribuir de alguma forma, com o intuito de facilitar a distribuição, divulgação e arrecadação de alimentos de um local, cidade e regiões.

3. Gerenciador de banco de dados

Para este projeto, foi escolhido um a utilização do NoSQL com o gerenciador MongoDB, um banco de dados não relacional baseado em documentos do tipo JSON.

O motivo de se utilizar um banco de dado não relacional neste projeto é devido aos benefícios que este modelo oferece, detalhados a seguir:

- **Escalabilidade:** Bancos de dados não relacionais tendem a ser altamente escaláveis. Eles permitem o armazenamento de dados em uma estrutura distribuída, facilitando a expansão à medida que o número de cadastro de alimentos, ONGs e famílias cresce.

- **Integração de dados desestruturados:** Em campanhas de arrecadação de alimentos, as ONGs podem gerar e coletar uma variedade de tipos de dados, incluindo textos, imagens, vídeos e dados geográficos. Um banco de dados não relacional pode armazenar e gerenciar esses dados desestruturados de forma mais eficiente do que um sistema relacional tradicional.

- **Performance em consultas complexas:** para consultas que tiverem grandes volumes de dados como doações, distribuições ou participações em campanhas a performance através de um banco de dados não relacional é superior ao modelo tradicional.

- **Flexibilidade de esquema:** como este projeto tem o intuito de já utilizar informações de ONGs e centrais existentes, os dados compartilhados podem estar em diversos formatos e estruturas. Para essa situação um banco de dados não relacional possui um modelo flexível de armazenamento de informações permitindo a obtenção dos mais variados tipos de dados sem muita dificuldade.

4. Modelo das coleções

A seguir é especificado as coleções que serão utilizados no projeto, sua definição, estrutura e organização:

- **ONGS:** Existe a necessidade de manter o cadastro das ONGs flexível para suportar futuras mudanças, como a adição ou remoção de atributos, conforme a necessidade de cada instituição. Essa coleção conterá os dados da instituição, localidade e cidades de cobertura de ações beneficentes. Cada ONG possuíra um acesso exclusivo no sistema onde poderão divulgar campanhas, metas de arrecadação, informações de eventos entre outros. Por esse motivo é armazenado a senha criptografada. A utilização do documento como modelo supri as necessidades além de simplificar a gestão e atualização dos dados relacionados.

```
{
  "cnpj": "12.345.678/0001-99",
  "nome": "Empresa Exemplo",
  "endereco": {
    "rua": "Rua Exemplo",
    "numero": 100,
    "cidade": "São Paulo",
    "estado": "SP",
    "cep": "01000-000"
  },
  "telefone": "(11) 1234-5678",
  "email": "contato@empresaexemplo.com",
  "links": [
    "http://www.empresaexemplo.com",
    "http://www.outrasite.com"
  ],
  "senha": "aec342fb38b8173b8e898229990ed6a4f196d9bed8428f6d2b5a7010fe85781b",
  "cidades_cobertura": [
    "São Paulo",
    "Rio de Janeiro",
    "Belo Horizonte"
  ]
}
```

- **Campanhas:** coleção que especificará as campanhas arrecadação de alimentos, ligadas a uma ou mais instituição. Campanhas podem conter diversos tipos de informações, como detalhes da campanha, comentários dos usuários, locais de arrecadação o que eleva a necessidade de uma estrutura mais flexível para suportar diferentes tipos de dados. Nessa coleção é utilizado objetos aninhados como “comentário” que permite usuários comentarem ou questionarem sobre algum evento de arrecadação e também e “locais_arrecadacao” que informa os locais onde os alimentos serão distribuídos, que podem incluir tanto centros permanentes quanto eventos pontuais.

Um exemplo de flexibilidade que a utilização de um banco de dados não relacional pode oferecer nessa situação seria o objeto “locais_arrecadacao”. Como é uma informação inteiramente relacionada a campanha, não houve a necessidade no momento de separar em uma outra coleção específica. Caso, em algum momento, esse objeto passe a conter uma grande quantidade de dados e diversas relações, inclusive com outras campanhas, a migração

desse objeto para uma nova coleção seria muito simples. Isso evitaria redundância e repetição de informações, bastando atualizar a coleção da campanha para referenciá-lo pelo seu "id".

```
{
  "id": 1,
  "titulo": "Campanha 1",
  "descricao": "Descricao 1",
  "data_inicio": "2023-01-01",
  "data_fim": "2023-12-31",
  "status": "Ativa",
  "meta": 1000,
  "comentarios": [
    {
      "usuario": "User 1",
      "comentario": "Comentário 1",
    }
  ],
  "locais_arrecadacao": [
    {
      "endereco": "Local 1",
      "horario_inicio": "08:00",
      "horario_termino": "18:00"
    }
  ],
  "cnpj_ong": "12.345.678/0001-99"
}
```

- **Doacoes:** Registros de doações feitas, vinculadas a campanhas específicas e/ou ONGs. Usar uma coleção para "Doações" é vantajoso para lidar com o grande número de transações. A flexibilidade de armazenar diferentes tipos de dados, como quantidade, data, detalhes dos alimentos ou dinheiro arrecadados e informações do doador, facilita a adaptação e expansão conforme o volume de doações cresce. Além disso, essa estrutura permite consultas rápidas e eficientes, e possibilita a realocação futura de bases de dados dependendo da região.

Outro exemplo de flexibilidade que utilizando a Doação como uma coleção é a distinção de doação de alimentos e outros de dinheiro. Ambos têm a mesma finalidade que é informar a doação/arrecadação e somente as informações necessárias são preenchidas, sem a necessidade de informar campos que não serão utilizados. O mesmo ocorre nos dados do doador, caso o doador queira se manter anônimo, não é necessário o preenchimento dos campos.

```
[
  {
    "id_campanha": 2,
    "alimento": "Feijão",
    "qtde": 15,
    "doador": {
      "nome": "João da Silva",
      "email": "joao.silva@example.com",
      "telefone": "(11) 98765-4321"
    }
  },
  {
    "id_campanha": 2,
    "dinheiro": 50.0,
    "doador": {
      "nome": "Maria Oliveira",
      "email": "maria.oliveira@example.com",
      "telefone": "(21) 91234-5678",
      "endereco": {
        "rua": "Rua das Flores",
        "numero": 123,
        "cidade": "Rio de Janeiro",
        "estado": "RJ",
        "cep": "12345-678"
      }
    }
  }
]
```

5. Consultas

- **Login ONG:** O banco de dados não relacionado fornece funções simples que permite uma consulta fácil e ágil ao banco de dados, nesta consulta é realizado a busca de uma ONG de acordo com as credenciais fornecidas.

```
db.ongs.findOne(
  {
    "cnpj": "12.345.678/0001-99",
    "senha": "aec342fb38b8173b8e898229990ed6a4f196d9bed8428f6d2b5a7010fe85781b"
  });
```

A função “FindOne” retorna apenas um registro que corresponda aos critérios de consulta especificados.

- **Detalhes ONG:** nesta consulta é realizada uma agregação a coleção de ongs e também de campanhas que participam

```

db.ongs.aggregate([
  {
    "$lookup": {
      "from": "campanhas",
      "localField": "cnpj",
      "foreignField": "cnpj_ong",
      "as": "campanhas_detalhes"
    }
  },
  {
    "$unwind": {
      "path": "$campanhas_detalhes",
      "preserveNullAndEmptyArrays": true
    }
  },
  {
    "$group": {
      "_id": "$cnpj",
      "nome": { "$first": "$nome" },
      "enderecos": { "$first": "$endereco" },
      "email": { "$first": "$email" },
      "links": { "$first": "$links" },
      "campanhas": { "$push": "$campanhas_detalhes.titulo" }
    }
  },
  {
    "$project": {
      "_id": 0,
      "cnpj": "$_id",
      "nome": 1,
      "enderecos": 1,
      "email": 1,
      "links": 1,
      "campanhas": 1
    }
  }
]);

```

Para unir informações da coleção **ong** e **campanha**, é utilizado a função de agregação \$lookup e \$unwind, responsáveis para selecionar outra coleção e o campo que fará a junção de ambas as coleções, então a consulta é agrupada utilizando a função \$group que fará o agrupamento de ONGs e suas campanhas. Por fim é utilizado o \$project para projetar o resultado final da consulta.

- **Total de alimento arrecadado:** obtém a quantidade total de doação de um alimento específico.

```
db.doacoes.aggregate([
  {
    "$match": {
      "alimento": {"$ne": null, "$eq": "Feijão"}
    }
  },
  {
    "$group": {
      "_id": "$alimento",
      "totalQuantidade": {"$sum": "$qtde"}
    }
  }
])
```

Esta consulta realiza a somatória da quantidade utilizando a função de agregação **\$sum** e filtra os registros do campo alimento que não são nulos (para garantir apenas doações de alimentos) e sejam iguais a “Feijão”.

- **Alimento mais arrecadado**

```
db.doacoes.aggregate([
  {
    "$match": {
      "alimento": {"$ne": null}
    }
  },
  {
    "$group": {
      "_id": "$alimento",
      "totalQuantity": {"$sum": "$qtde"}
    }
  },
  {
    "$sort": {"totalQuantity": -1}
  },
  {
    "$limit": 1
  }
])
```

Esta consulta realiza a somatória da quantidade de alimento utilizando a função de agregação **\$sum** e ordena a maior quantidade dos registros retornados através da função **\$sort** e limita a visualização de apenas um resultado através da função **\$limit**

Também podemos alterar a consulta para listar o alimento mais doado por campanha. Para isso é necessário apenas alterar o **\$match**:

```
"$match": {
  "alimento": { "$ne": null },
  "id_campanha": 1
}
```

Com isso, será verificado o alimento mais doado apenas da campanha “1”.

- **Alimentos arrecadados por campanha**

```
db.doacoes.aggregate([
  {
    "$match": {
      "id_campanha": 1,
      "alimento": {"$ne": null}
    }
  },
  {
    "$project": {
      "_id": 0,
      "data": 1,
      "alimento": 1,
      "qtde": 1,
      "doador": 1
    }
  }
])
```

A consulta lista todos os alimentos arrecadados por campanha e projeta informações como o nome do alimento, data, quantidade doada e o doador.

- **Dinheiro arrecadados por campanha**

```
db.doacoes.aggregate([
  {
    "$match": {
      "id_campanha": 1,
      "dinheiro": {"$ne": null}
    }
  },
  {
    "$project": {
      "_id": 0,
      "dinheiro": 1
    }
  }
])
```

Semelhante a consulta do **Alimentos arrecadados por campanha** esta consulta irá retornar o total de arrecadação de dinheiro de uma campanha específica e irá projetar apenas o total do dinheiro.

- **Detalhes de uma campanha:** esta agregação obtém dados gerais de uma campanha, como o título, meta, data de início e fim, instituições que participarão e locais de arrecadação.

```
db.doacoes.aggregate([
  {
    "$match": {
      "id": 1
    }
  },
  {
    "$lookup": {
      "from": "ongs",
      "localField": "cnpj_ong",
      "foreignField": "cnpj",
      "as": "ong_details"
    }
  },
  {
    "$unwind": {
      "path": "$ong_details",
      "preserveNullAndEmptyArrays": true
    }
  },
  {
    "$project": {
      "_id": 0,
      "titulo": 1,
      "descricao": 1,
      "data_inicio": 1,
      "data_fim": 1,
      "status": 1,
      "meta": 1,
      "comentario": 1,
      "locais_arrecadacao": 1,
      "cnpj_ong": 1,
      "nome_ong": "$ong_details.nome"
    }
  }
])
```

Nesta consulta também é utilizado as funções de agregação \$lookup e \$unwind, responsáveis para selecionar outra coleção e o campo que fará a junção de ambas as coleções para obter os dados das instituições responsáveis pela campanha.

- **Meta de arrecadação:** apresenta a meta de arrecadação e o total já arrecadado de uma campanha específica.

```
db.campanhas.aggregate([
  {
    "$match": {
      "id": 1
    }
  },
  {
    "$lookup": {
      "from": "doacoes",
      "localField": "id",
      "foreignField": "id_campanha",
      "as": "doacoes_detalhes"
    }
  },
  {
    "$unwind": {
      "path": "$doacoes_detalhes",
      "preserveNullAndEmptyArrays": true
    }
  },
  {
    "$match": {
      "doacoes_detalhes.dinheiro": {"$ne": null}
    }
  },
  {
    "$group": {
      "_id": "$id",
      "titulo": {"$first": "$titulo"},
      "descricao": {"$first": "$descricao"},
      "meta": {"$first": "$meta"},
      "total_arrecadado": {"$sum": "$doacoes_detalhes.dinheiro"}
    }
  },
  {
    "$project": {
      "_id": 0,
      "titulo": 1,
      "descricao": 1,
      "meta": 1,
      "total_arrecadado": 1,
    }
  }
])
```

Nesta agregação, uma nova coleção de arrays é gerada mediante a função \$unwind onde também foi aplicado atributos path para nomear e preserveNullAndEmptyArrays para preservar campos nulos e vazios para que também sejam exibidos. Para a geração dessa consulta também foi necessário utilizar informação de outro documento, para isso é utilizado a função \$lookup na qual define o documento doações e os campos para efetuar o relacionamento. Por fim é utilizado o agrupamento através da função \$group onde a coleção é agrupada pela quantidade total de arrecadações e projetada através da função \$project.

O resultado esperado será conforme o seguinte exemplo:

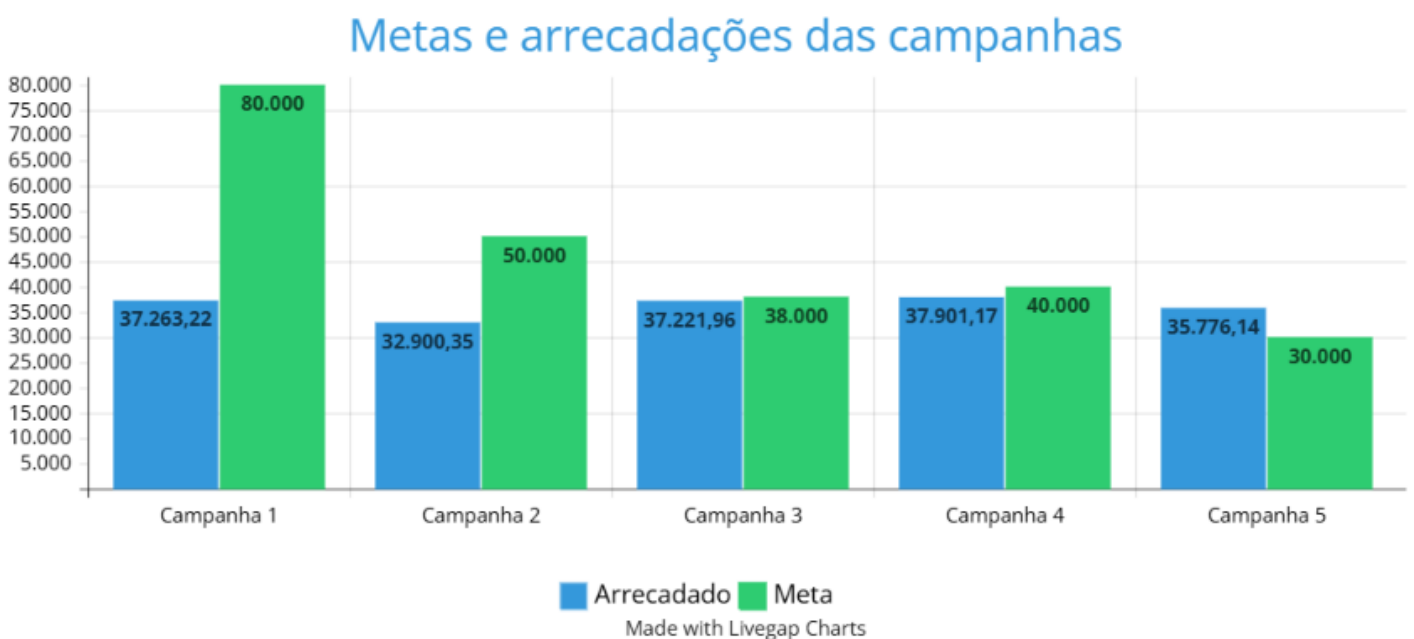
```
{
  "titulo": "Campanha de Arrecadação 1",
  "descricao": "Descrição da campanha 1",
  "meta": 10000,
  "total_arrecadado": 150,
}
```

7. Dados gráficos resultantes

Com a utilização das consultas criadas anteriormente, podemos obter informações valiosas para uso tanto interno do sistema como também para usos estatísticos. Abaixo segue alguns exemplos de gráficos gerados a partir de uma base de dados fictícia

- **Metas e arrecadações das campanhas**

Neste gráfico é apresentando 5 campanhas diferentes durante 1 ano, suas metas e quantidade arrecadada em valor de dinheiro.

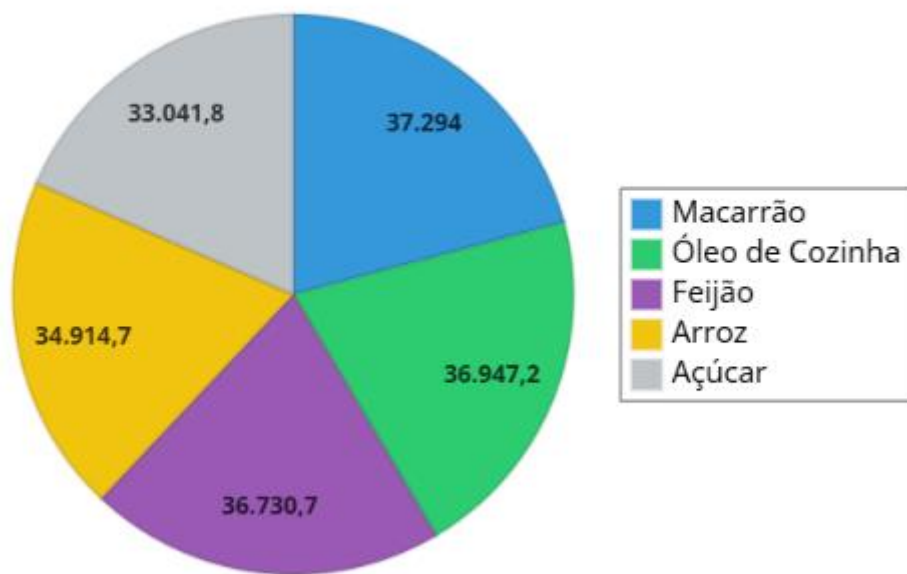


- **Total de alimentos doados durante essa campanha por KG**

No gráfico a seguir é apresentado a arrecadação separado por alimentos durante toda campanha.

Arredcadação de alimentos por KG

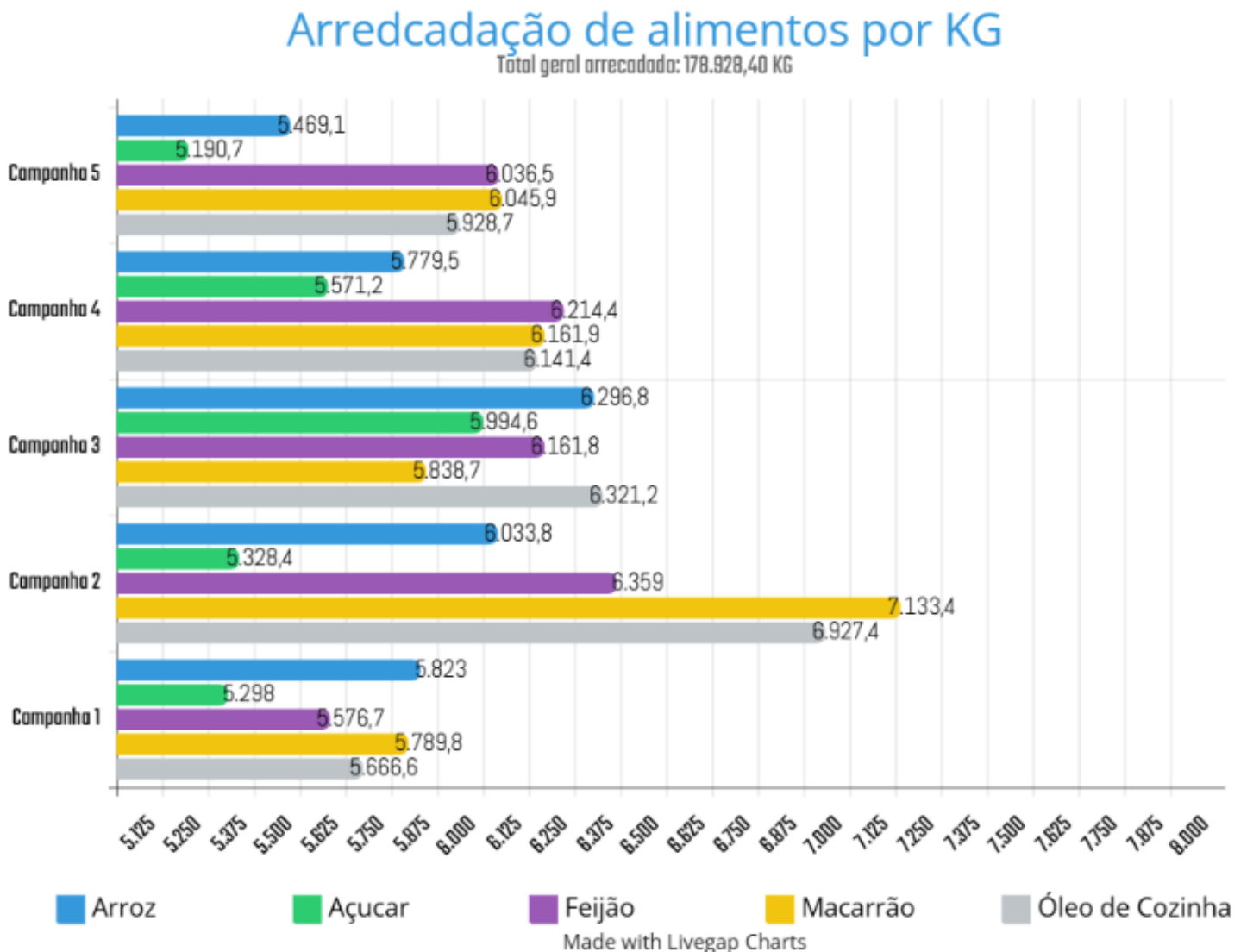
Total geral arrecadado: 178.928,40 KG



Made with Livegap Charts

- **Total de cada alimento doado por campanha em KG**

No seguinte gráfico as arrecadações de cada alimento são separado por campanha para uma melhor visualização dos alimentos arrecadados.



Toda as informações acima são fictícias utilizadas para simular as consultas realizadas através do banco não relacional Mongo de acordo com a estrutura das coleções citadas nessa documentação. Para a visualização e aquisição dos dados utilizados nesta documentação acesse https://github.com/VitorEduardoOliveira/PI_1SEM_2024