

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

ABRAÃO MANGE GARCIA GAMA

**GERENCIAMENTO DE FARMÁCIAS UTILIZANDO
MONGODB**

**CAMPOS DO JORDÃO
2024**

RESUMO

O gerenciamento de dados desempenha um papel fundamental na administração de farmácias devido à complexidade das informações e à necessidade de controle rigoroso de estoque, vendas e relacionamento com clientes e fornecedores. Este projeto tem como objetivo desenvolver um sistema de gerenciamento de dados para farmácias utilizando o banco de dados NoSQL MongoDB. Por meio da construção de um modelo orientado a documentos, busca-se otimizar os processos operacionais, oferecendo escalabilidade e flexibilidade. A pesquisa baseia-se nas demandas do setor farmacêutico, alinhando soluções tecnológicas às necessidades práticas desse ambiente.

INTRODUÇÃO

O setor farmacêutico apresenta uma série de desafios, como o controle de medicamentos, a gestão de estoque e a conformidade com regulamentações. Para enfrentar essas questões, este trabalho propõe a implementação de um sistema de gerenciamento baseado em MongoDB, um banco de dados NoSQL que possibilita a manipulação eficiente de grandes volumes de dados não estruturados e semiestruturados. O objetivo é otimizar processos operacionais, como vendas, controle de estoque e relacionamento com clientes. A metodologia inclui a análise das necessidades do setor, a modelagem de dados adequada e o desenvolvimento de consultas para suportar operações críticas. A escolha do MongoDB justifica-se pela sua capacidade de lidar com dados dinâmicos e por oferecer uma alternativa moderna aos modelos relacionais tradicionais.

OBJETIVOS

- Desenvolver e implementar um sistema de gerenciamento de dados para farmácias utilizando o MongoDB.
- Proporcionar uma solução escalável e flexível para controle de estoque, registro de vendas e integração com fornecedores.
- Demonstrar a eficiência do modelo NoSQL em contextos práticos e regulamentados.

JUSTIFICATIVA

A complexidade das operações farmacêuticas exige soluções que promovam eficiência e conformidade. O MongoDB, com sua estrutura orientada a documentos, oferece uma maneira eficaz de gerenciar informações diversificadas, como produtos, clientes e fornecedores. Sua aplicação pode reduzir erros, otimizar o uso de recursos e melhorar a experiência dos clientes ao garantir precisão e rapidez nos serviços prestados.

METODOLOGIA

A metodologia baseia-se em três etapas principais:

1. **Pesquisa de campo:** Foram realizadas observações em farmácias para compreender processos como controle de estoque, registro de vendas e interação com fornecedores. Essas informações serviram de base para o desenvolvimento do sistema.
2. **Modelagem de dados:** Foi criado um modelo orientado a documentos no MongoDB, com coleções que representam as principais entidades do sistema, como Produtos, Clientes, Vendas e Fornecedores. As estruturas foram projetadas para facilitar operações comuns no setor farmacêutico.
3. **Implementação do sistema:** O banco de dados foi implementado no MongoDB, destacando a flexibilidade de consultas e a escalabilidade do modelo. O desenvolvimento incluiu a criação de consultas específicas para gerenciar estoques, acompanhar vendas e monitorar fornecedores.

RESULTADOS OBTIDOS

// Estrutura inicial das coleções no MongoDB

// Coleção: Clientes

```
{  
  "_id": ObjectId(),  
  "nome": "string",  
  "cpf": "string",  
  "telefone": "string",  
  "email": "string"  
}
```

// Coleção: Funcionarios

```
{  
  "_id": ObjectId(),  
  "nome": "string",  
  "cpf": "string",  
  "telefone": "string",  
  "dataContratacao": "date",  
  "salario": "number",  
  "email": "string",  
  "cargo": "string"  
}
```

```
// Coleção: CategoriaProdutos
```

```
{  
  "_id": ObjectId(),  
  "nome": "string",  
  "descricao": "string"  
}
```

```
// Coleção: Produtos
```

```
{  
  "_id": ObjectId(),  
  "nome": "string",  
  "descricao": "string",  
  "preco": "number",  
  "categoriaId": ObjectId()  
}
```

```
// Coleção: Fornecedores
```

```
{  
  "_id": ObjectId(),  
  "nome": "string",  
  "telefone": "string",  
  "email": "string",  
  "endereco": "string",  
  "cnpj": "string"  
}
```

// Coleção: Estoques

```
{  
  "_id": ObjectId(),  
  "quantidade": "number",  
  "localizacao": "string",  
  "produtoid": ObjectId()  
}
```

// Coleção: RegistroArmazenamento

```
{  
  "_id": ObjectId(),  
  "dataRegistro": "date",  
  "quantidade": "number",  
  "produtoid": ObjectId(),  
  "fornecedorId": ObjectId(),  
  "estoqueId": ObjectId()  
}
```

// Coleção: Vendas

```
{  
  "_id": ObjectId(),  
  "dataVenda": "date",  
  "valorVenda": "number",  
  "clienteId": ObjectId(),  
  "funcionarioId": ObjectId()  
}
```

```
// Coleção: Pagamentos

{

  "_id": ObjectId(),

  "dataPagamento": "date",

  "valorPagamento": "number",

  "metodoPagamento": "string",

  "vendald": ObjectId()

}
```

```
// Coleção: ItensVenda

{

  "_id": ObjectId(),

  "vendald": ObjectId(),

  "produtold": ObjectId(),

  "quantidade": "number",

  "precoUnitario": "number"

}
```

```
// Coleção: ReceitasMedicas

{

  "_id": ObjectId(),

  "clienteld": ObjectId(),

  "dataReceita": "date"

}
```

// Exemplos de consultas em MongoDB

// 1. Verificar clientes sem e-mail

```
{  
  "email": { "$exists": false }  
}
```

// 2. Listar clientes em ordem alfabética

```
{  
  "find": {},  
  "sort": { "nome": 1 }  
}
```

// 3. Filtrar clientes com telefone iniciando por "11"

```
{  
  "telefone": { "$regex": "^11" }  
}
```

// 4. Produtos com preço acima de R\$100,00

```
{  
  "preco": { "$gt": 100 }  
}
```


// 5. Produtos de uma categoria específica

```
{  
  "categoryId": ObjectId("ID_CATEGORIA")  
}
```

// 6. Ordenar produtos por preço crescente

```
{  
  "find": {},  
  "sort": { "preco": 1 }  
}
```

// 7. Soma de produtos em estoque

```
{  
  "aggregate": [  
    { "$group": { "_id": null, "totalEstoque": { "$sum": "$quantidade" } } }  
  ]  
}
```

// 8. Preço médio dos produtos

```
{  
  "aggregate": [  
    { "$group": { "_id": null, "precoMedio": { "$avg": "$preco" } } }  
  ]  
}
```

// 9. Buscar clientes que criaram a conta nos últimos 30 dias

```
{  
  "dataCriacao": { "$gte": new Date(new Date() - 30 * 24 * 60 * 60 * 1000) }  
}
```

// 10. Listar produtos fora de estoque

```
{  
  "quantidade": { "$lte": 0 }  
}
```

// 11. Contar clientes ativos

```
{  
  "aggregate": [  
    { "$match": { "ativo": true } },  
    { "$count": "clientesAtivos" }  
  ]  
}
```

// 12. Listar clientes que possuem endereço em São Paulo

```
{  
  "endereco.cidade": "São Paulo"  
}
```

// 13. Buscar produtos com desconto

```
{  
  "desconto": { "$exists": true, "$gt": 0 }  
}
```

// 14. Atualizar preço de todos os produtos de uma categoria em 10%

```
{  
  "updateMany": {  
    "filter": { "categoricalId": ObjectId("ID_CATEGORIA") },  
    "update": { "$mul": { "preco": 1.1 } }  
  }  
}
```

// 15. Filtrar clientes pelo nome (case-insensitive)

```
{  
  "nome": { "$regex": ".*nome_cliente.*", "$options": "i" }  
}
```

// 16. Buscar pedidos realizados entre duas datas

```
{  
  "dataPedido": { "$gte": new Date("2023-01-01"), "$lte": new Date("2023-12-31") }  
}
```

// 17. Agrupar clientes por estado e contar

```
{  
  "aggregate": [  
    { "$group": { "_id": "$endereco.estado", "totalClientes": { "$sum": 1 } } }  
  ]  
}
```

// 18. Ordenar pedidos por data decrescente

```
{  
  "find": {},  
  "sort": { "dataPedido": -1 }  
}
```

// 19. Buscar produtos com nome ou descrição contendo "promoção"

```
{  
  "$or": [  
    { "nome": { "$regex": "promoção", "$options": "i" } },  
    { "descricao": { "$regex": "promoção", "$options": "i" } }  
  ]  
}
```

// 20. Filtrar pedidos com valor total maior que R\$500

```
{  
  "valorTotal": { "$gt": 500 }  
}
```

// 21. Contar produtos por categoria

```
{  
  "aggregate": [  
    { "$group": { "_id": "$categoriaId", "totalProdutos": { "$sum": 1 } } }  
  ]  
}
```

// 22. Adicionar um campo calculado de preço final (com desconto aplicado)

```
{  
  "aggregate": [  
    {  
      "$addFields": {  
        "precoFinal": { "$subtract": [ "$preco", { "$multiply": [ "$preco", "$desconto" ] } ] }  
      }  
    }  
  ]  
}
```

// 23. Buscar clientes que fizeram mais de 5 pedidos

```
{  
  "quantidadePedidos": { "$gt": 5 }  
}
```

// 24. Remover clientes inativos há mais de um ano

```
{  
  "deleteMany": {  
    "filter": {  
      "ativo": false,  
      "ultimaAtividade": { "$lt": new Date(new Date() - 365 * 24 * 60 * 60 * 1000) }  
    }  
  }  
}
```

// 25. Buscar produtos com múltiplas tags

```
{  
  "tags": { "$all": ["promoção", "novo"] }  
}
```

// 26. Listar os 10 produtos mais vendidos

```
{  
  "aggregate": [  
    { "$sort": { "quantidadeVendida": -1 } },  
    { "$limit": 10 }  
  ]  
}
```

// 27. Contar pedidos por mês em um ano

```
{  
  "aggregate": [  
    { "$match": { "ano": 2023 } },  
    {  
      "$group": {  
        "_id": { "mes": { "$month": "$dataPedido" } },  
        "totalPedidos": { "$sum": 1 }  
      }  
    },  
    { "$sort": { "_id.mes": 1 } }  
  ]  
}
```

// 28. Atualizar o status de pedidos atrasados para "Atrasado"

```
{  
  "updateMany": {  
    "filter": { "dataEntrega": { "$lt": new Date(), "$exists": true }, "status": "Pendente" },  
    "update": { "$set": { "status": "Atrasado" } }  
  }  
}
```

// 29. Produtos com preço entre R\$50 e R\$150

```
{  
  "preco": { "$gte": 50, "$lte": 150 }  
}
```

// 30. Buscar os 5 últimos clientes cadastrados

```
{  
  "find": {},  
  "sort": { "dataCriacao": -1 },  
  "limit": 5  
}
```


// Inserção de dados

// Inserção na coleção Clientes

```
db.clientes.insertMany([  
  { "nome": "João Silva", "cpf": "12345678901", "telefone": "11987654321", "email":  
    "joao.silva@gmail.com" },  
  { "nome": "Maria Oliveira", "cpf": "98765432100", "telefone": "11965432100" },  
  { "nome": "Pedro Santos", "cpf": "11222333445", "telefone": "11954321098", "email":  
    "pedro.santos@gmail.com" },  
  { "nome": "Ana Maria", "cpf": "22334455667", "telefone": "11987653210" }  
]);
```

// Inserção na coleção Funcionários

```
db.funcionarios.insertMany([  
  { "nome": "Ana Souza", "cpf": "11223344556", "telefone": "11987654322",  
    "dataContratacao": ISODate("2019-01-10"), "salario": 2500.50, "email":  
    "ana.souza@gmail.com", "cargo": "Farmacêutica" },  
  { "nome": "Carlos Lima", "cpf": "22334455667", "telefone": "11987654323",  
    "dataContratacao": ISODate("2020-05-15"), "salario": 1800.00, "email":  
    "carlos.lima@gmail.com", "cargo": "Atendente" },  
  { "nome": "Beatriz Alves", "cpf": "33445566778", "telefone": "11987654324",  
    "dataContratacao": ISODate("2021-03-20"), "salario": 2000.00, "email":  
    "beatriz.alves@gmail.com", "cargo": "Gerente" }  
]);
```

// Inserção na coleção CategoriaProdutos

```
db.categoriaProdutos.insertMany([  
  { "nome": "Medicamentos", "descricao": "Produtos farmacêuticos." },  
  { "nome": "Higiene", "descricao": "Produtos de higiene pessoal." },  
  { "nome": "Cosméticos", "descricao": "Produtos de beleza e cuidados pessoais." }  
]);
```

// Inserção na coleção Produtos

```
db.produtos.insertMany([  
  { "nome": "Paracetamol", "descricao": "Medicamento analgésico.", "preco": 10.00,  
    "categoricalId": ObjectId("ID_CATEGORIA_MEDICAMENTOS") },  
  { "nome": "Shampoo", "descricao": "Produto para higiene capilar.", "preco": 25.00,  
    "categoricalId": ObjectId("ID_CATEGORIA_HIGIENE") },  
  { "nome": "Creme Hidratante", "descricao": "Hidratação para a pele.", "preco": 15.00,  
    "categoricalId": ObjectId("ID_CATEGORIA_COSMETICOS") }  
]);
```

// Inserção na coleção Fornecedores

```
db.fornecedores.insertMany([  
  { "nome": "Distribuidora Farma", "telefone": "1133334444", "email": "contato@farma.com",  
    "endereco": "Rua das Flores, 123", "cnpj": "11222333444455" },  
  { "nome": "Higiene Total", "telefone": "1144445555", "email": "vendas@higiene.com",  
    "endereco": "Av. Brasil, 456", "cnpj": "223334445555666" },  
  { "nome": "Beleza Distribuidora", "telefone": "1155556666", "email": "vendas@beleza.com",  
    "endereco": "Rua Bela Vista, 789", "cnpj": "334445556666777" }  
]);
```

// Inserção na coleção Estoques

```
db.estoques.insertMany([  
  { "quantidade": 100, "localizacao": "Setor A", "produtoId":  
    ObjectId("ID_PRODUTO_PARACETAMOL") },  
  { "quantidade": 50, "localizacao": "Setor B", "produtoId":  
    ObjectId("ID_PRODUTO_SHAMPOO") },  
  { "quantidade": 30, "localizacao": "Setor C", "produtoId":  
    ObjectId("ID_PRODUTO_CREME_HIDRATANTE") }  
]);
```

// Inserção na coleção RegistroArmazenamento

```
db.registroArmazenamento.insertMany([  
  
  { "dataRegistro": ISODate("2023-12-01"), "quantidade": 20, "produtoId":  
    ObjectId("ID_PRODUTO_PARACETAMOL"), "fornecedorId":  
    ObjectId("ID_FORNECEDOR_FARMA"), "estoqueId": ObjectId("ID_ESTOQUE_A") },  
  
  { "dataRegistro": ISODate("2023-12-05"), "quantidade": 10, "produtoId":  
    ObjectId("ID_PRODUTO_SHAMPOO"), "fornecedorId":  
    ObjectId("ID_FORNECEDOR_HIGIENE"), "estoqueId": ObjectId("ID_ESTOQUE_B") }  
  
]);
```

// Inserção na coleção Vendas

```
db.vendas.insertMany([  
  
  { "dataVenda": ISODate("2023-12-07"), "valorVenda": 50.00, "clienteId":  
    ObjectId("ID_CLIENTE_JOAO"), "funcionarioId": ObjectId("ID_FUNCIONARIO_ANA") },  
  
  { "dataVenda": ISODate("2023-12-08"), "valorVenda": 30.00, "clienteId":  
    ObjectId("ID_CLIENTE_MARIA"), "funcionarioId": ObjectId("ID_FUNCIONARIO_CARLOS")  
  }  
  
]);
```

// Inserção na coleção Pagamentos

```
db.pagamentos.insertMany([  
  
  { "dataPagamento": ISODate("2023-12-07"), "valorPagamento": 50.00,  
    "metodoPagamento": "Cartão", "vendaId": ObjectId("ID_VENDA_1") },  
  
  { "dataPagamento": ISODate("2023-12-08"), "valorPagamento": 30.00,  
    "metodoPagamento": "Dinheiro", "vendaId": ObjectId("ID_VENDA_2") }  
  
]);
```

```
// Inserção na coleção ItensVenda
```

```
db.itensVenda.insertMany([
```

```
  { "vendaid": ObjectId("ID_VENDA_1"), "produtoid":  
    ObjectId("ID_PRODUTO_PARACETAMOL"), "quantidade": 2, "precoUnitario": 10.00 },
```

```
  { "vendaid": ObjectId("ID_VENDA_2"), "produtoid":  
    ObjectId("ID_PRODUTO_CREME_HIDRATANTE"), "quantidade": 1, "precoUnitario": 15.00 }
```

```
]);
```

```
// Inserção na coleção ReceitasMedicas
```

```
db.receitasMedicas.insertMany([
```

```
  { "clienteid": ObjectId("ID_CLIENTE_MARIA"), "dataReceita": ISODate("2023-11-30") },
```

```
  { "clienteid": ObjectId("ID_CLIENTE_JOAO"), "dataReceita": ISODate("2023-11-25") }
```

```
]);
```

CONCLUSÃO

O sistema de gerenciamento de farmácias utilizando MongoDB demonstrou como os bancos de dados NoSQL podem otimizar a gestão de informações dinâmicas, como clientes, produtos, fornecedores e vendas. A flexibilidade do MongoDB permitiu a criação de um modelo eficiente, adaptável e escalável, atendendo às necessidades operacionais do setor farmacêutico. O sistema reduziu erros e otimizou processos, mas futuras melhorias podem incluir análises preditivas, integração com automação de vendas e dashboards interativos. O projeto mostrou o potencial dos bancos NoSQL, oferecendo uma base sólida para avanços tecnológicos no setor.

REFERÊNCIAS BIBLIOGRÁFICAS

- Chodorow, Kristina. MongoDB: The Definitive Guide. O'Reilly Media, 2013.
- Strauch, Claus. NoSQL Databases. Lecture Notes, 2011.
- Date, C. J. An Introduction to Database Systems. Pearson, 2019.
- Documentação oficial do MongoDB (<https://www.mongodb.com/docs/>).