

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

LUCAS APARECIDO DOS SANTOS

**PROJETO E IMPLEMENTAÇÃO DE BANCO DE DADOS
NOSQL PARA O SISTEMA DE GESTÃO DE IMPRESSÕES
3D**

**CAMPOS DO JORDÃO
2025**

RESUMO

Este trabalho apresenta a modelagem e a implementação de um banco de dados Não Relacional (NoSQL) para o sistema "PrintFlow", voltado ao gerenciamento de serviços de impressão 3D. Diferentemente dos modelos relacionais tradicionais, que exigem esquemas rígidos, este projeto adota uma abordagem orientada a documentos utilizando o SGBD MongoDB. O objetivo é demonstrar como a flexibilidade do esquema NoSQL beneficia o armazenamento de dados polimórficos, como configurações de fatiamento (slicing) e especificações técnicas de materiais, que variam drasticamente entre pedidos. O relatório descreve os conceitos fundamentais de NoSQL, a arquitetura do MongoDB e apresenta a estrutura das coleções JSON/BSON desenvolvidas, evidenciando o ganho de performance e agilidade no desenvolvimento através da desnormalização de dados.

Palavras-chave: NoSQL, MongoDB, Banco de Dados Orientado a Documentos, Impressão 3D, Big Data.

ABSTRACT

This paper presents the modeling and implementation of a non-relational database (NoSQL) for the “PrintFlow” system, designed for managing 3D printing services. Unlike traditional relational models, which require rigid schemas, this project adopts a document-oriented approach using the MongoDB DBMS. The goal is to demonstrate how the flexibility of the NoSQL schema benefits the storage of polymorphic data, such as slicing configurations and technical specifications of materials, which vary dramatically between orders. The report describes the fundamental concepts of NoSQL, the architecture of MongoDB, and presents the structure of the JSON/BSON collections developed, highlighting the gains in performance and agility in development through data denormalization.

Keywords: NoSQL, MongoDB, Document-Oriented Database, 3D Printing, Big Data.

SUMÁRIO

1.INTRODUÇÃO.....	5
1.1 Objetivos.....	5
1.2 Justificativa.....	5
2. APORTE TEÓRICO.....	5
2.1 Bancos de Dados NoSQL.....	5
3. METODOLOGIA.....	6
3.1 Considerações Iniciais.....	6
4. RESULTADOS OBTIDOS.....	7
4.1 Coleção produtos.....	7
4.2 Coleção estoque_filamentos.....	8
4.3 Coleção pedidos.....	9
4.4 Consultas (Resultados de Execução).....	9
5. CONCLUSÃO.....	10
REFERÊNCIAS	12

1 INTRODUÇÃO

A indústria da manufatura aditiva (impressão 3D) lida com um volume de dados que possui alta variabilidade. Um único pedido pode conter arquivos complexos de modelagem, dezenas de parâmetros de configuração de impressão (temperatura, velocidade, preenchimento) e metadados que não se encaixam bem em tabelas rígidas de linhas e colunas.

1.1 Objetivos

O objetivo deste trabalho é projetar a camada de persistência de dados do sistema "PrintFlow" utilizando o paradigma NoSQL. Busca-se explorar como a modelagem orientada a documentos pode simplificar a arquitetura da aplicação, eliminando a necessidade de *joins* complexos e permitindo que a estrutura dos dados evolua sem a necessidade de migrações de esquema custosas (ALTER TABLE).

1.2 Justificativa

A escolha por uma abordagem NoSQL justifica-se pela natureza semi-estruturada dos dados de impressão 3D. Tentar normalizar parâmetros de impressão em um banco relacional resultaria em tabelas excessivamente fragmentadas ou colunas com muitos valores nulos. O NoSQL oferece a escalabilidade horizontal e a flexibilidade de schema necessárias para um sistema moderno e ágil.

2 APORTE TEÓRICO

O projeto foi executado em fases sequenciais, partindo da abstração do problema até a implementação técnica.

2.1 Bancos de Dados NoSQL

O termo NoSQL (*Not Only SQL*) refere-se a uma classe de sistemas de gerenciamento de banco de dados que não seguem o modelo relacional tradicional

(RDBMS). Eles surgiram para atender demandas de alta escalabilidade, disponibilidade e armazenamento de dados semi-estruturados.

Conforme descrito pela AWS Amazon (AMAZON, s.d.), os bancos de dados NoSQL utilizam diversos modelos de dados para acessar e gerenciar informações, sendo as quatro categorias principais:

1. **Chave-Valor (Key-Value):** O modelo mais simples, altamente particionável, onde cada item contém chaves e valores.
2. **Documentos (Document-Oriented):** Armazena dados em documentos (JSON, BSON), permitindo estruturas aninhadas e indexação flexível (Foco deste trabalho).
3. **Grafos (Graph):** Focado em criar e navegar relacionamentos entre entidades conectadas.
4. **Em Memória e Colunares:** Otimizados para velocidade de resposta ou armazenamento de grandes volumes de dados analíticos.

3 METODOLOGIA

O desenvolvimento deste trabalho seguiu uma abordagem prática de modelagem de dados voltada para performance de leitura e facilidade de manutenção.

3.1 Considerações Iniciais

Ao contrário da modelagem relacional, onde a prioridade é a normalização para evitar redundância, na modelagem NoSQL para o "PrintFlow", priorizou-se a forma como os dados são acessados pela aplicação. Foi identificado que, ao visualizar um pedido, o sistema precisa de todas as informações do cliente e dos itens simultaneamente, sugerindo uma estratégia de embedding (aninhamento) de dados.

Para a implementação, foi utilizado o **MongoDB**. Trata-se de um banco de dados de código aberto, orientado a documentos, que oferece alta performance, alta disponibilidade e escalabilidade automática.

- **Modelo de Dados:** O MongoDB armazena dados em **BSON** (Binary JSON). O BSON estende o modelo JSON (JavaScript Object Notation) para incluir tipos de dados adicionais e ser eficiente para codificação e decodificação em várias linguagens (MONGODB, s.d.).
- **Estrutura:**
 - **Collections (Coleções):** Equivalentes às tabelas em SQL, mas não impõem um esquema fixo.
 - **Documents (Documentos):** Equivalentes às linhas/registros, mas são estruturas hierárquicas (objetos dentro de objetos) e polimórficas (documentos na mesma coleção podem ter campos diferentes).

A metodologia de modelagem seguiu o padrão de **Desnormalização Inteligente**. Em vez de separar Pedido, Item_Pedido e Endereco em tabelas distintas, optou-se por consolidar informações que são lidas juntas dentro do mesmo documento.

4 RESULTADOS OBTIDOS

Abaixo apresenta-se a estrutura das principais *Collections* projetadas para o sistema "PrintFlow", demonstrando a sintaxe JSON/BSON utilizada pelo MongoDB.

4.1 Coleção produtos

Diferente do modelo relacional, onde teríamos que criar tabelas extras para "Componentes", aqui podemos armazenar a receita do produto (lista de partes) dentro do próprio documento do produto.

```

1  /* Exemplo de Documento na coleção 'produtos' */
2  {
3      "_id": ObjectId("654a3b2c9d8e7f1a2b3c4d5e"),
4      "nome": "Boneco Astronauta Articulado",
5      "descricao": "Boneco com 15 pontos de articulação",
6      "preco_base": 150.00,
7      "categoria": "Colecionáveis",
8      "tags": ["espaço", "brinquedo", "articulado"],
9      "especificacoes_tecnicas": {
10         "altura_cm": 25,
11         "peso_total_estimado_g": 130
12     },
13     "componentes": [
14         {
15             "nome": "Capacete",
16             "gramas_necessarias": 10,
17             "material_recomendado": "PETG"
18         },
19         {
20             "nome": "Corpo",
21             "gramas_necessarias": 120,
22             "material_recomendado": "PLA"
23         }
24     ],
25     "data_criacao": ISODate("2025-11-10T08:00:00Z")
26 }

```

4.2 Coleção estoque_filamentos

Aqui armazenamos os insumos.

```

1  /* Exemplo de Documento na coleção 'estoque_filamentos' */
2  {
3      "_id": ObjectId("507f1f77bcf86cd799439011"),
4      "material": "PLA",
5      "cor": "Azul Marinho",
6      "fabricante": "3D Fila",
7      "preco_por_grama": 0.15,
8      "estoque_atual_gramas": 4500.50,
9      "historico_movimentacao": [
10         { "data": ISODate("2025-11-01"), "tipo": "entrada", "qtd": 5000 },
11         { "data": ISODate("2025-11-05"), "tipo": "saida_pedido", "qtd": -499.50 }
12     ]
13 }

```


4.3 Coleção pedidos

Este é o exemplo mais claro da vantagem do NoSQL. O documento de pedido contém **tudo**: os dados do cliente (snapshot), os itens comprados e, crucialmente, as configurações de impressão específicas para aquele pedido, sem necessidade de *joins*.

```
1  /* Exemplo de Documento na coleção 'pedidos' */
2  {
3    "_id": ObjectId("654a4f8e9d8e7f1a2b3c9999"),
4    "data_pedido": ISODate("2025-11-25T14:30:00Z"),
5    "status": "Em Impressão",
6    "valor_total": 300.00,
7    "cliente": {
8      "id_usuario": ObjectId("554a3b2c9d8e7f1a2b3c1111"),
9      "nome": "João Silva",
10     "email": "joao@email.com"
11   },
12   "itens": [
13     {
14       "id_produto": ObjectId("654a3b2c9d8e7f1a2b3c4d5e"),
15       "nome_produto": "Boneco Astronauta Articulado",
16       "quantidade": 2,
17       "preco_unitario": 150.00,
18       "configuracao_personalizada": {
19         "resolucao_camada": "0.12mm",
20         "infill": "20%",
21         "suportes": true
22       },
23       "filamentos_escolhidos": [
24         { "componente": "Capacete", "cor": "Dourado", "material": "PLA Silk" },
25         { "componente": "Corpo", "cor": "Branco", "material": "PLA" }
26       ]
27     }
28   ]
29 }
```

4.4 Consultas (Resultados de Execução)

Para validar o modelo, abaixo estão exemplos de operações de consulta utilizando a sintaxe do MongoDB (MQL).

Consulta 1: Buscar pedidos com status "Em Impressão"

```
1 db.pedidos.find(  
2   { "status": "Em Impressão" },  
3   { "cliente.nome": 1, "data_pedido": 1 }  
4 )
```

Resultado: Retorna instantaneamente os pedidos sem precisar cruzar tabelas de status.

Consulta 2: Calcular o total de vendas de produtos da categoria

"Colecionáveis" Utiliza-se o *Aggregation Framework* do MongoDB (equivalente ao GROUP BY).

```
1 db.pedidos.aggregate([  
2   { $unwind: "$itens" }, // Desmembra o array de itens  
3   { $lookup: {           // Busca dados na coleção produtos (semelhante ao JOIN)  
4     from: "produtos",  
5     localField: "itens.id_produto",  
6     foreignField: "_id",  
7     as: "dados_produto"  
8   }},  
9   { $match: { "dados_produto.categoria": "Colecionáveis" } },  
10  { $group: {  
11    _id: null,  
12    total_vendas: { $sum: { $multiply: ["$itens.quantidade", "$itens.preco_unitario"] } }  
13  } }  
14 ])
```

5 CONCLUSÃO

A implementação do projeto "PrintFlow" utilizando o MongoDB demonstrou que bancos de dados NoSQL orientados a documentos são altamente adequados para sistemas de e-commerce e manufatura que lidam com produtos customizáveis.

A principal vantagem observada foi a **flexibilidade do esquema**. Foi possível adicionar campos como `configuracao_personalizada` dentro dos itens do

pedido sem precisar alterar a estrutura global do banco, algo que exigiria alterações complexas em um modelo relacional. Além disso, a estratégia de embutir os itens dentro do documento do pedido reduziu a latência de leitura, pois uma única busca no banco já retorna o pedido completo pronto para ser exibido na interface ou enviado para a impressora.

Como sugestão de melhoria futura, recomenda-se a implementação de **índices de texto (Text Indexes)** do MongoDB para permitir buscas rápidas por palavras-chave nas descrições e tags dos produtos, além de estudar o particionamento (**Sharding**) das coleções caso o volume de pedidos cresça a nível de Big Data.

REFERÊNCIAS

AWS AMAZON. *O que é NoSQL?*. Disponível em:

<<https://aws.amazon.com/pt/nosql/>>. Acesso em: 25 nov. 2025.

DEV MEDIA. *Introdução ao MongoDB*. Disponível em:

<<https://www.devmedia.com.br/introducao-ao-mongodb/30792>>. Acesso em: 25 nov. 2025.

MONGODB. *MongoDB Documentation: The Document Model*. Disponível em:

<<https://www.mongodb.com/docs/manual/core/document/>>. Acesso em: 25 nov. 2025.