**Nomes: Gabriel de Paula dos Santos RM: 554168  
  
  
1. Decisão de Migrar ou Inovar**

A decisão de usar o MongoDB para a aplicação foi baseada nas necessidades de flexibilidade e escalabilidade que o projeto requer. Como a aplicação envolve informações de sensores de tráfego, cruzamentos, semáforos e alertas, o MongoDB se mostrou uma excelente escolha devido à sua estrutura NoSQL, que é ideal para armazenar dados com diferentes formatos e tipos, e pela capacidade de escalar horizontalmente conforme o número de usuários e dados aumentam.

Além disso, ao criar coleções específicas para cada área, conseguimos segmentar melhor os dados, melhorando o desempenho de consultas e aumentando a eficiência da manipulação de dados.

**2. Escolha das Collections**

1. **Coleção semaforos**: A coleção semaforos é responsável por armazenar as informações sobre os semáforos. Cada semáforo tem informações como sua localização, estado (verde ou vermelho), tempo de duração do sinal verde e presença de veículos. Essa coleção é crucial para o gerenciamento dos semáforos no sistema de tráfego.
2. **Coleção alertas**: A coleção alertas é utilizada para registrar mensagens de alerta no sistema. Isso pode incluir eventos como semáforos com falha, congestionamentos ou outras situações que precisam ser monitoradas.
3. **Coleção sensores**: A coleção sensores armazena informações sobre os sensores instalados no sistema, como sensores de movimento para detecção de veículos. Cada sensor tem um código de série, status e a data da última manutenção.
4. **Coleção gestores**: A coleção gestores armazena informações sobre os gestores de tráfego, que são responsáveis pelo monitoramento e controle dos semáforos e do tráfego. Ela inclui o nome do gestor, ramal e data de cadastro.
5. **Coleção cruzamentos**: A coleção cruzamentos armazena informações sobre os cruzamentos no sistema, como nome, localização, número de vias, estado do semáforo, número de veículos detectados e sensores presentes em cada cruzamento.

**3. Comandos para Criação das Collections e Operações CRUD**

**1. Coleção semaforos**

* **Comando para criar a coleção**:

db.createCollection("semaforos")

* **Operação CRUD**:

**a. Inserir Documentos (Criar)**:

db.semaforos.insertMany([

{ "id": "1", "localizacao": "R. Doze de Outubro, 321 - Centro, São Paulo - SP", "veiculoPresente": true, "tempoVerde": 30 },

{ "id": "2", "localizacao": "Av. Paulista, 1000 - Bela Vista, São Paulo - SP", "veiculoPresente": false, "tempoVerde": 40 },

{ "id": "3", "localizacao": "R. 15 de Novembro, 850 - Centro, Campinas - SP", "veiculoPresente": true, "tempoVerde": 25 },

{ "id": "4", "localizacao": "R. Cardoso Pimentel, 143 - São Miguel Paulista, São Paulo - SP", "veiculoPresente": false, "tempoVerde": 45 },

{ "id": "5", "localizacao": "Av. Brigadeiro Faria Lima, 1600 - Pinheiros, São Paulo - SP", "veiculoPresente": true, "tempoVerde": 35 }

])

**b. Buscar Todos (Read)**:

db.semaforos.find()

**c. Atualizar Documento (Update)**:

db.semaforos.updateOne({ "id": "3" }, { $set: { "tempoVerde": 30 } })

**d. Deletar Documento (Delete)**:

db.semaforos.deleteOne({ "id": "2" })

**2. Coleção alertas**

* **Comando para criar a coleção**:

db.createCollection("alertas")

* **Operação CRUD**:

**a. Inserir Documentos (Criar)**:

db.alertas.insertMany([

{ "id": "1", "message": "Semáforo com falha na R. Doze de Outubro", "dataHora": "2025-03-17T10:00:00" },

{ "id": "2", "message": "Congestionamento na Av. Paulista", "dataHora": "2025-03-17T11:00:00" },

{ "id": "3", "message": "Falha no sensor da R. Cardoso Pimentel", "dataHora": "2025-03-17T12:00:00" },

{ "id": "4", "message": "Semáforo Verde na Av. Brigadeiro Faria Lima", "dataHora": "2025-03-17T13:00:00" },

{ "id": "5", "message": "Semáforo com falha na Av. Paulista", "dataHora": "2025-03-17T14:00:00" }

])

**b. Buscar Todos (Read)**:

db.alertas.find()

**c. Atualizar Documento (Update)**:

db.alertas.updateOne({ "id": "2" }, { $set: { "message": "Congestionamento intenso na Av. Paulista" } })

**d. Deletar Documento (Delete)**:

db.alertas.deleteOne({ "id": "3" })

**3. Coleção sensores**

* **Comando para criar a coleção**:

db.createCollection("sensores")

* **Operação CRUD**:

**a. Inserir Documentos (Criar)**:

db.sensores.insertMany([

{ "id": "1", "codigoSerie": "SEN001", "ultimaManutencao": "2025-02-15" },

{ "id": "2", "codigoSerie": "SEN002", "ultimaManutencao": "2025-03-10" },

{ "id": "3", "codigoSerie": "SEN003", "ultimaManutencao": "2025-03-12" },

{ "id": "4", "codigoSerie": "SEN004", "ultimaManutencao": "2025-01-20" },

{ "id": "5", "codigoSerie": "SEN005", "ultimaManutencao": "2025-02-01" }

])

**b. Buscar Todos (Read)**:

db.sensores.find()

**c. Atualizar Documento (Update)**:

db.sensores.updateOne({ "id": "2" }, { $set: { "ultimaManutencao": "2025-03-15" } })

**d. Deletar Documento (Delete)**:

db.sensores.deleteOne({ "id": "4" })

**4. Coleção gestores**

* **Comando para criar a coleção**:

db.createCollection("gestores")

* **Operação CRUD**:

**a. Inserir Documentos (Criar)**:

db.gestores.insertMany([

{ "id": "1", "nome": "Carlos Silva", "ramal": "102", "dataCadastro": "2025-03-01" },

{ "id": "2", "nome": "Fernanda Souza", "ramal": "103", "dataCadastro": "2025-03-02" },

{ "id": "3", "nome": "José Almeida", "ramal": "104", "dataCadastro": "2025-03-03" },

{ "id": "4", "nome": "Ana Lima", "ramal": "105", "dataCadastro": "2025-03-04" },

{ "id": "5", "nome": "Ricardo Oliveira", "ramal": "106", "dataCadastro": "2025-03-05" }

])

**b. Buscar Todos (Read)**:

db.gestores.find()

**c. Atualizar Documento (Update)**:

db.gestores.updateOne({ "id": "3" }, { $set: { "ramal": "110" } })

**d. Deletar Documento (Delete)**:

db.gestores.deleteOne({ "id": "2" })

**5. Coleção cruzamentos**

* **Comando para criar a coleção**:

db.createCollection("cruzamentos")

* **Operação CRUD**:

**a. Inserir Documentos (Criar)**:

db.cruzamentos.insertMany([

{ "id": "1", "nome": "Cruzamento A", "localizacao": "Av. Faria Lima x R. Cardoso Pimentel", "vias": 4, "estadoSemaforo": "verde", "nroVeiculosDetectados": 20, "dataCadastro": "2025-03-01", "horaUltimaAtualizacao": "2025-03-17T10:00:00", "sensores": "SEN001, SEN003" },

{ "id": "2", "nome": "Cruzamento B", "localizacao": "Av. Paulista x R. Augusta", "vias": 3, "estadoSemaforo": "vermelho", "nroVeiculosDetectados": 15, "dataCadastro": "2025-03-02", "horaUltimaAtualizacao": "2025-03-17T11:00:00", "sensores": "SEN002, SEN004" },

{ "id": "3", "nome": "Cruzamento C", "localizacao": "R. 15 de Novembro x Av. Brigadeiro Faria Lima", "vias": 5, "estadoSemaforo": "verde", "nroVeiculosDetectados": 30, "dataCadastro": "2025-03-03", "horaUltimaAtualizacao": "2025-03-17T12:00:00", "sensores": "SEN005, SEN003" }

])

**b. Buscar Todos (Read)**:

db.cruzamentos.find()

**c. Atualizar Documento (Update)**:

db.cruzamentos.updateOne({ "id": "1" }, { $set: { "nroVeiculosDetectados": 25 } })

**d. Deletar Documento (Delete)**:

db.cruzamentos.deleteOne({ "id": "2" })

**4. Evidências das Collections e Operações CRUD**

**Evidências**: As evidências podem ser capturadas através de prints de tela no MongoDB Compass ou qualquer outro cliente de MongoDB ao rodar os comandos de inserção, leitura, atualização e exclusão, mostrando os resultados e o estado das coleções após as operações.

**Conclusão**

O MongoDB foi escolhido devido à sua flexibilidade para lidar com dados de tráfego e semáforos, permitindo uma estrutura NoSQL escalável e adaptável às necessidades do sistema. As coleções criadas têm um papel essencial para organizar e acessar dados sobre semáforos, alertas, sensores, gestores e cruzamentos, todos relacionados ao controle e monitoramento do tráfego inteligente. As operações de CRUD garantem que os dados sejam manipulados de forma eficiente e persistente no banco de dados.