# Etapa 5 - Projeto de um banco de dados NoSQL para a aplicação

*Guilherme Sagawa - 11225676*

*Pedro Rezende Mendonça - 8961491*

## 1. Escolha do tipo de modelo NoSQL

Nesta etapa do projeto, optamos por utilizar um banco de dados NoSQL baseado em documentos. Essa escolha foi feita com base na análise dos requisitos funcionais e operacionais definidos nas etapas anteriores e nos conceitos abordados em aula, voltada à representação de dados semiestruturados (JSON/árvore).

O modelo de documentos é particularmente apropriado para aplicações com dados estruturados de forma hierárquica, como é o caso da nossa rede de transporte público. Muitas entidades do sistema — como usuários com seus bilhetes, veículos com seu histórico de manutenções, viagens com ocorrências e escalas de operadores — possuem uma estrutura naturalmente aninhada e autocontida, o que torna a modelagem por documentos mais direta e eficiente.

Além disso, a maior parte das consultas e operações frequentes do sistema são focadas em uma única entidade e seus dados relacionados, como por exemplo:

- Consultar os bilhetes de um usuário;
- Registrar uma ocorrência em uma viagem;
- Atualizar o status de um veículo;
- Verificar o histórico de manutenções de um ônibus.

Todas essas ações podem ser executadas com eficiência sobre documentos individuais, sem necessidade de operações complexas de junção, o que melhora o desempenho e simplifica o código da aplicação.

Outros modelos NoSQL, como bancos de grafos ou famílias de colunas, não se mostraram tão adequados. O modelo em grafos é mais indicado quando há relações altamente conectadas e navegação relacional complexa, o que não é o caso predominante nesta aplicação. Poderia se considerar que, por se tratar de uma rede, o modelo de grafos poderia otimizar operações relacionadas a escolha de rotas. Mas o propósito da aplicação não é implementar um serviço de roteamento. Além disso, operações recorrentes, como atualizações na localização dos veículos ou dos saldos dos bilhetes, seriam prejudicadas pelo desempenho pior dos modelos de grafos para operações de adição, remoção e atualização. Já o modelo colunar é mais eficiente para grandes volumes de dados tabulares com acesso massivo por chave, mas tem baixa

flexibilidade para representar estruturas aninhadas ou registros com múltiplos subcomponentes, como bilhetes, escalas ou manutenções.

Dessa forma, o uso de um banco de documentos proporciona:

- Flexibilidade estrutural;
- Facilidade de leitura e escrita;
- Eficiência no armazenamento e acesso a dados aninhados;
- Aderência à modelagem orientada ao domínio do problema.

Por esses motivos, o modelo de documentos foi escolhido como a solução NoSQL mais adequada para esta aplicação.

# 2. Critérios e escolhas de modelagem

Para construir o diagrama do esquema NoSQL, utilizamos o método de conversão de esquemas EER e a notação de modelo de agregados usada por Frozza et al. (2022). Identificamos a escolha de relações de referência com setas rosa e identificamos aninhamentos com balões azuis. Essas escolhas foram feitas usando como referência a lista de operações mais comuns apresentada na etapa 2, reproduzida abaixo:

**Consulta**

- 1: Buscar linhas que conectam uma origem a um destino
- 2: Listar todos os horários de uma linha específica
- 3: Consultar veículos atualmente em operação
- 4: Verificar estatísticas de pontualidade por linha
- 5: Consultar histórico de ocorrências por tipo
- 6: Analisar demanda por horário e linha
- 7: Consultar rotas alternativas entre dois pontos
- 8: Visualizar histórico de manutenções por veículo
- 9: Consultar operadores disponíveis para uma determinada data
- 10: Analisar tendências de demanda por linha

**Modificação**

- 1: Registrar nova viagem
- 2: Atualizar status de um veículo
- 3: Adicionar nova linha ao sistema
- 4: Registrar manutenção realizada
- 5: Atualizar horários de uma linha
- 6: Registrar ocorrência operacional
- 7: Adicionar novo veículo à frota
- 8: Modificar rota de uma linha existente

- 9: Adicionar ou remover ponto
- 10: Atualizar escala de operadores
- 11: Adicionar novo operador ao sistema

As escolhas de modelagem utilizaram esses critérios, somados à relação hierárquica das entidades no projeto conceitual EER. Destacamos as seguintes escolhas em algumas coleções:

**Funcionário** - Como espera-se que os atributos relacionados à estas tabelas não sejam modificados com muita frequência e as principais operações realizadas são de inserção ou remoção de funcionários (contratação e demissão, respectivamente) as tabelas `FUNCIONARIO`, `TECNICO`, `ADMINISTRADOR`, `COBRADOR`, `MOTORISTA`, `EMPRESA` e `ESCALA` foram denormalizadas em um único documento.

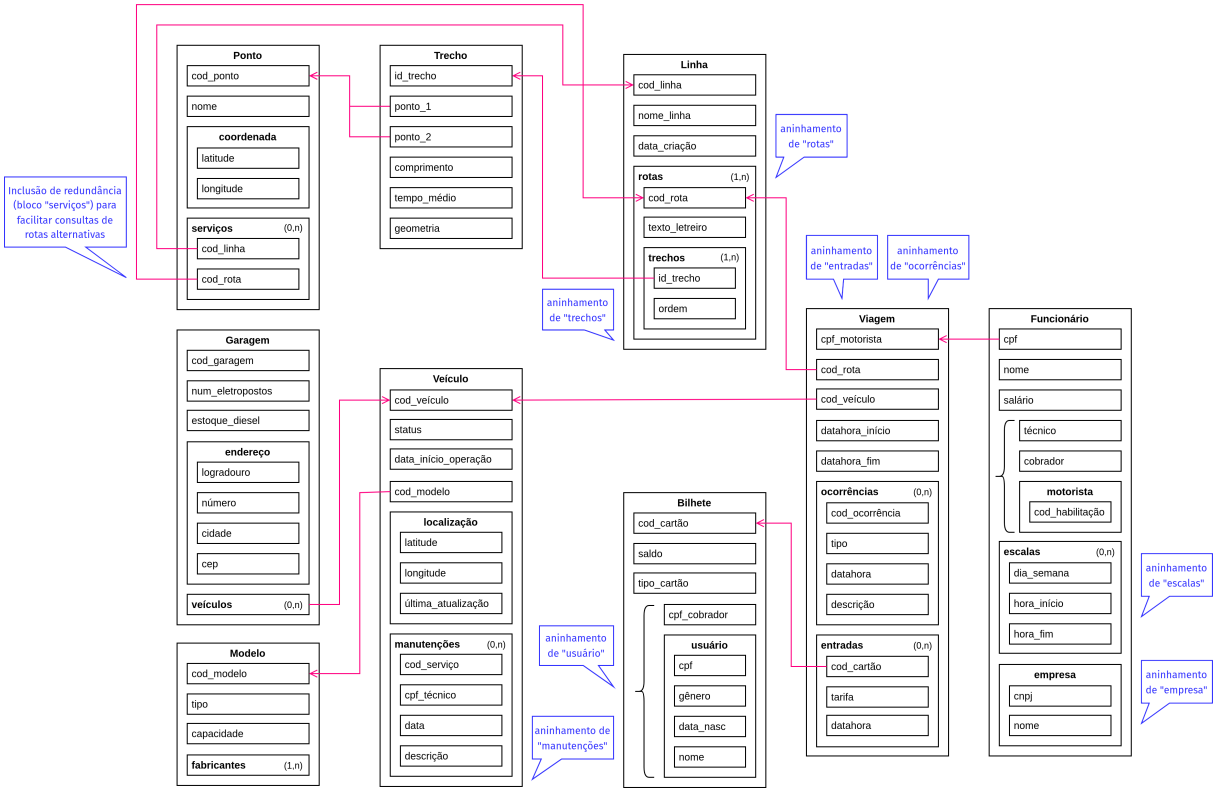
**Ponto** - Neste agregado, optamos por inserir uma redundância adicional no banco (bloco `serviços`) para facilitar as operações de consulta de alternativas de serviços na rede que atendam um ponto de origem e outro de destino — uma das operações cotidianas mais comuns.

**Linha** - A coleção `linhas` foi criada a partir da denormalização das tabela `LINHA`, `ROTA` e `PERCORRE`. Esta denormalização decorre da relação estrita entre essas tabelas e seus atributos, os quais não se espera modificações frequentes. Além disso, a criação desta coleção permite a realização de consultas de linhas pelos usuários e adição/remoção de linhas pelos operadores de forma mais eficiente.

**Bilhete** - A coleção `bilhete` é a denormalização das tabelas `USUARIO`, `BILHETE_COBRADOR`, `BILHETE_USUARIO` e `BILHETE`. Estas três tabelas foram agrupadas nesta coleção devido à relação intrínseca que elas possuem. Como a relação entre `BILHETE` e `USUARIO` é de N:1, englobar `USUARIO` como um atributo da coleção permite uma eficiente forma de inserção e consulta. Optamos por aninhar a entidade "usuário" dentro da coleção de agregados "Bilhete", e não o oposto, porque esperamos um baixo número de bilhetes por usuário e sabemos que as atualizações de saldo dos bilhetes são mais frequentes, o que torna esse agregado mais importante na hierarquia.

**Viagem** - A coleção `viagens` é resutlado da denormalização das tabelas `ENTRADA`, `OCORRENCIA` e `VIAGEM` do modelo relacional. Esta coleção foi criada com visando a escalabilidade do modelo de Documentos NoSQL devido ao alto fluxo de registros esperado e sem a necessidade de edição de agregados antigos já inseridos no DB -- pois toda viagem finalizada não deve receber novas operações de atualização. Além disso, há uma correlação lógica entre as três tabelas: Toda entrada ou ocorrência deve ser feita dentro de um viagem.

Apresentamos abaixo o diagrama de agregados resultante.




# 3. Esquema de dados

Apresentamos abaixo a implementação do diagrama de agregados em esquema XML.

1. Coleção funcionários

```
<funcionário>
    <cpf></cpf>
    <nome></nome>
    <salario></salario>
    <tipo></tipo>
        <operador_tipo>
            <cod_habilitação></cod_habilitação>
        </operador_tipo>
    <escalas>
        <escala>
            <dia_semana></dia_semana>
            <hora_inicio></hora_inicio>
            <hora_fim></hora_fim>
        </escala>
    </escalas>
    <empresa>
        <cnpj></cnpj>
        <nome></nome>
    </empresa>
</funcionário>
```

2. Coleção pontos

```
<ponto>
    <cod_ponto></cod_ponto>
    <nome></nome>
    <coordenada>
        <lat></lat>
        <lon></lon>
    </coordenada>
    <serviços>
        <serviço>
            <cod_linha></cod_linha>
            <cod_rota></cod_rota>
        </serviço>
    </serviços>
</ponto>
```

3. Coleção trechos

```
<trecho>
    <id_trecho></id_trecho>
    <ponto_1></ponto_1>
    <ponto_2></ponto_2>
    <comprimento></comprimento>
    <tempo_medio></tempo_medio>
    <geometria></geometria>
</trecho>
```

4. Coleção Linhas

```
<linha>
    <cod_linha></cod_linha>
    <nome_linha></nome_linha>
    <data_criação></data_criação>
    <rotas>
        <rota>
            <cod_rota></cod_rota>
            <texto_letreiro></texto_letreiro>
            <trechos>
                <trecho>
                    <id_trecho></id_trecho>
```

```
                    <ordem></ordem>
                </trecho>
            </trechos>
        </rota>
    </rotas>
</linha>
```

5. Coleção Bilhete

```
<bilhete>
    <cod_cartao></cod_cartao>
    <saldo></saldo>  <!-- Modificações frequentes -->
    <tipo_cartao></tipo_cartao>
    <cpf_cobrador></cpf_cobrador>
    <usuário>
        <cpf></cpf>
        <genero></genero>
        <data_nascimento></data_nascimento>
        <nome></nome>
    </usuário>
</bilhete>
```

6. Coleção Garagem

```
<garagem>
    <cod_garagem></cod_garagem>
    <num_eletropostos></num_eletropostos>
    <estoque_diesel></estoque_diesel>
    <capacidade></capacidade>
    <endereço>
        <logradouro></logradouro>
        <número></número>
        <cidade></cidade>
        <cep></cep>
    </endereço>
    <veículos>
        <veículo></veículo>
</veículos>
</garagem>
```

7. Coleção veículos

```
<veículo>
    <cod_veículo></cod_veículo>
    <localização>
        <lat></lat>
        <lon></lon>
        <última_atualização></última_atualização>
    </localização>
    <status></status>
    <data_início_operação></data_início_operação>
    <cod_modelo></cod_modelo>
    <manutenções>
        <manutenção>
            <cod_serviço></cod_serviço>
            <cpf_técnico></cpf_técnico>
            <data></data>
            <descrição></descrição>
        </manutenção>
    </manutenções>
</veículo>
```

8. Coleção modelos

```
<modelo>
<cod_modelo></cod_modelo>
    <tipo></tipo>
    <capacidade></capacidade>
    <fabricantes>
        <fabricante></fabricante>
    </fabricantes>
</modelo>
```

9. Coleção viagens

```
<viagem>
    <cod_veiculo></cod_veiculo>
    <cpf_motorista></cpf_motorista>
    <datahora_início></datahora_início>
    <datahora_fim></datahora_fim>
    <cod_rota></cod_rota>
    <ocorrências>
        <ocorrência>
            <cod_ocorrência></cod_ocorrência>
            <tipo></tipo>
            <datahora></datahora>
```

```
            <descrição></descrição>
        </ocorrência>
    </ocorrências>
    <entradas>
        <entrada>
        <cod_cartão></cod_cartão>
        <tarifa></tarifa>
        <datahora></datahora>
    </entrada>
</entradas>
</viagem>
```

## 4. Referências

Frozza, A. , Schreiner, A., & Mello, R. Projeto de Bancos de Dados NoSQL - material de um minicurso do Simpósio Brasileiro de Bancos de Dados (SBBD 2022)