

**Pró-Reitoria Acadêmica  
Curso de Ciência da Computação  
Trabalho de Disciplina**

**Sistema de Transporte Privado Integrado com Banco de  
Dados Relacional e NoSQL**

**Autores: Ana Beatriz Alves, David Cordeiro, Eduarda Alves,  
Guilherme Andrade, Ícaro de Oliveira  
Orientador: Prof. Jefferson Salomao Rodrigues**

**Ana Beatriz Alves, David Cordeiro, Eduarda Alves,  
Guilherme Andrade, Ícaro de Oliveira**

**Sistema de Transporte Privado Integrado com Banco de Dados Relacional e  
NoSQL**

Documento apresentado ao Curso de  
graduação de Bacharelado em Ciência da  
Computação da Universidade Católica de  
Brasília, como requisito parcial para  
obtenção da aprovação nas disciplinas de  
Laboratório de Banco de Dados

Orientador: Prof. Jefferson Salomao Rodrigues

**Brasília  
2025**

## Resumo

Referência: Ana Beatriz Alves, David Cordeiro, Eduarda Alves, Guilherme Andrade, Ícaro de Oliveira. Sistema de Transporte Privado Integrado com Banco de Dados Relacional e NoSQL, 2025. nr p. Bacharelado em Ciência da Computação – UCB – Universidade Católica de Brasília, Taguatinga – DF, 2025.

Este artigo apresenta o desenvolvimento de um sistema de transporte privado utilizando um banco de dados relacional (MySQL) e um banco de dados NoSQL (MongoDB). O sistema inclui frontend, backend, autenticação, CRUD das entidades principais e mecanismos de auditoria. A solução demonstra integração entre dados estruturados e não estruturados, uso de triggers, views, procedures e fuctions de geração de IDs, além de justificativas para todas as escolhas tecnológicas.

**Palavras-chave:** MySQL, MongoDB, Spring Boot, Sistema de Transporte, Banco de Dados.

# 1. Introdução

Sistemas de transporte precisam lidar com grande volume de informações estruturadas, como motoristas, veículos e viagens. Ao mesmo tempo, devem registrar eventos variáveis, como logs e avaliações. Cada tipo de informação exige um banco adequado.

Este trabalho apresenta uma solução híbrida que combina MySQL para dados críticos e MongoDB para dados flexíveis, oferecendo consistência, desempenho e escalabilidade.

## 2. Objetivos

### 2.1 Objetivo Geral

Desenvolver um sistema funcional que integre MySQL e MongoDB, atendendo aos requisitos da disciplina de Laboratório de Banco de Dados.

### 2.2 Objetivos Específicos

- Implementar banco relacional com índices, triggers, views, procedures e function própria de IDs.
- Utilizar MongoDB para dados não estruturados.
- Criar frontend (HTML/CSS/JAVASCRIPT) e backend (Java/Spring Boot).
- Desenvolver login e controle de acesso por grupos.
- Documentar e justificar cada escolha tecnológica.

### 3. Metodologia

A metodologia consistiu em:

1. Modelar o banco relacional.
2. Implementar o MySQL com todos os itens obrigatórios.
3. Criar um segundo banco NoSQL em MongoDB.
4. Desenvolver backend em Spring Boot conectando ambos.
5. Construir frontend simples e funcional.
6. Realizar testes integrados (CRUD, viagens, avaliações).
7. Elaborar documento técnico seguindo formato de artigo acadêmico.

## 4. Descrição do Sistema

### 4.1 Funcionalidades

- Login e controle de permissões.
- Cadastro e edição de motoristas, veículos, passageiros e rotas.
- Criação e finalização de viagens.
- Associação de passageiros às viagens (N:N).
- Relatórios via views SQL.
- Registro de logs no MySQL e no MongoDB.
- Avaliação de motoristas no MongoDB.

### 4.2 Tecnologias Utilizadas

**Frontend:** HTML, CSS, JavaScript.

**Backend:** Java + Spring Boot (REST API).

**Banco Relacional:** MySQL

**Banco NoSQL:** MongoDB

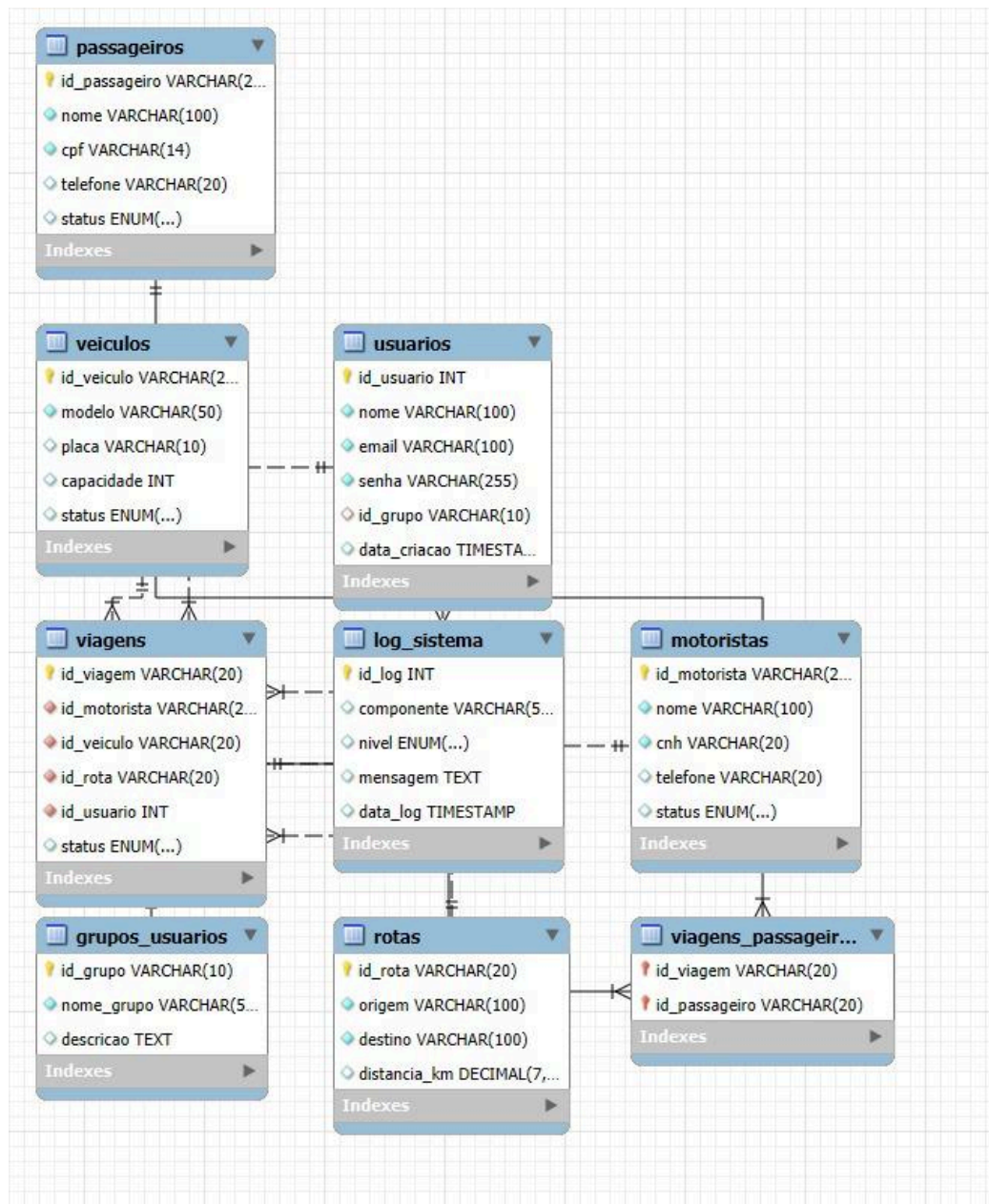
**Ferramenta de Administração do Banco Relacional:** MySQL Workbench

### 4.3 Justificativa das Tecnologias

- MySQL: garante integridade, constraints e relações entre entidades.
- MongoDB: ideal para logs e avaliações sem estrutura fixa.
- Spring Boot: permite integração simples, rápida e segura.
- HTML/JS: atende à necessidade de interface leve para testes.

## 5. Modelagem do Banco de Dados

### 5.1 Diagrama Entidade-Relacionamento





## 5.2 Entidades e Relacionamentos (Resumo)

- **usuarios:** credenciais e vínculo com grupos.
- **grupos\_usuarios:** níveis de acesso.
- **motoristas:** dados profissionais.
- **veiculos:** frota cadastrada.
- **rotas:** origens, destinos e distâncias.
- **passageiros:** pessoas transportadas.
- **viagens:** liga motorista, veículo e rota.
- **viagens\_passageiros:** relação N:N.

## 5.3 Justificativa dos Elementos Técnicos

### Índices

Criados para:

- acelerar buscas por status, e-mail e datas;
- melhorar o desempenho do login e relatórios.

### Triggers

- **trg\_usuario\_criado:** auditoria automática.
- **trg\_finaliza\_viagem:** libera veículo ao finalizar viagem.
- **trg\_passageiro\_criado:** log de cadastro de passageiros.

### Views

- **vw\_viagens\_finalizadas:** relatório administrativo.
- **vw\_usuarios\_grupos:** consulta rápida de permissões.
- **vw\_passageiros\_viagens:** relação viagens ↔ passageiros.

## Procedures / Function

- **Function gerar\_id:** cria IDs únicos e rastreáveis.
- **nova\_viagem:** padroniza criação de viagens.
- **novo\_passageiro:** automatiza cadastro + log.

## 6. Controle de Acesso

### 6.1 Usuários e Grupos

- **ADM:** acesso total.
- **OPERADOR:** pode criar viagens e cadastrar entidades.
- **VISUALIZADOR:** somente leitura.

### 6.2 Regras de Acesso

- O backend usa o grupo do usuário para permitir ou bloquear ações.
- O banco possui usuário administrativo e usuário de aplicação com permissões mínimas.
- Não há uso de root, conforme exigência.

## 7. Uso do Banco NoSQL

### 7.1 Explicação Técnica

MongoDB usa documentos JSON flexíveis, sem necessidade de schema, ideal para informações que mudam com o tempo ou possuem campos variáveis.

### 7.2 Aplicação no Sistema

Foi utilizado para:

- **avaliacoes\_motorista:** notas e comentários livres de passageiros.

O uso do NoSQL evita sobrecarregar o banco relacional com registros volumosos.

## 8. Conclusão

A solução atende integralmente aos requisitos do trabalho, combinando o que há de melhor nos bancos relacionais e não relacionais. O sistema entrega consistência, flexibilidade e escalabilidade, além de interface simples e backend robusto.

Os testes demonstraram funcionamento correto da integração MySQL + MongoDB, além das regras automáticas implementadas via triggers, procedures e function.

## 9. Referências

- Documentação oficial do MySQL.
- Documentação do MongoDB.
- Documentação do Spring Boot.
- Materiais da disciplina de Laboratório de Banco de Dados.