

SISTEMA DE GERENCIAMENTO DE MATRÍCULAS ESCOLARES

Trabalho Final - Laboratório de Banco de Dados

1. INTRODUÇÃO

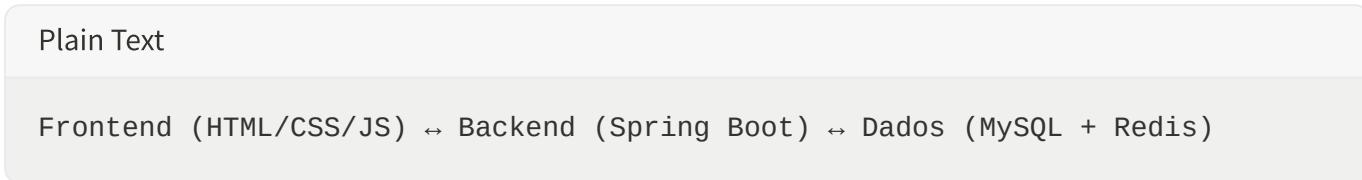
Este trabalho apresenta um sistema completo de gerenciamento de matrículas escolares que integra banco de dados relacional (MySQL), banco NoSQL (Redis), backend robusto (Java Spring Boot) e frontend responsivo (HTML/CSS/JavaScript).

O projeto demonstra domínio de conceitos avançados de SGBD, incluindo modelagem relacional, índices, triggers, views, procedures, functions, controle de acesso e integração com NoSQL.

2. ARQUITETURA DO SISTEMA

2.1 Visão Geral

O sistema segue arquitetura em 3 camadas:



2.2 Componentes

- **Frontend:** Interface responsiva com autenticação JWT
- **Backend:** API REST com injeção de dependência
- **MySQL:** Dados relacionais com triggers, views, procedures
- **Redis:** Fila de requisições assíncrona

3. BANCO DE DADOS RELACIONAL (MySQL)

3.1 Tabelas Implementadas

1. **grupos_usuarios** - Grupos com permissões

2. **usuarios** - Usuários do sistema (obrigatória)
3. **turmas** - Turmas escolares
4. **alunos** - Alunos com informações pessoais
5. **matriculas** - Registros de matrícula
6. **logs_auditoria** - Auditoria de operações

3.2 Índices (9 total)

- usuarios.email - Busca rápida de login
- usuarios.cpf - Validação de CPF único
- alunos.cpf, alunos.email, alunos.turma_id - Buscas frequentes
- matriculas.aluno_id, matriculas.turma_id, matriculas.status - Filtros
- logs_auditoria.data_operacao - Consultas de auditoria

3.3 Triggers (2+)

1. **tr_usuarios_update_timestamp** - Atualiza data de modificação automaticamente
2. **tr_alunos_audit_insert** - Registra operações em logs_auditoria

3.4 Views (2+)

1. **vw_alunos_por_turma** - Alunos agrupados com cálculo de ocupação
2. **vw_matriculas_ativas** - Matrículas com dados consolidados

3.5 Procedures (2+)

1. **sp_registrar_matricula** - Registra matrícula com validações
2. **sp_cancelar_matricula** - Cancela matrícula com rastreamento

3.6 Functions (3+)

1. **fn_gerar_id_matricula()** - Gera IDs personalizados
2. **fn_calcular_idade()** - Calcula idade automaticamente
3. **fn_validar_cpf()** - Valida formato de CPF

3.7 Controle de Acesso

Grupo	Alunos	Turmas	Matrículas
-------	--------	--------	------------

ADMINISTRADOR	CRUD	CRUD	CRUD
PROFESSOR	READ	READ	READ
SECRETARIA	CRUD	READ	CRUD
RESPONSAVEL	READ	-	READ

4. BANCO DE DADOS NOSQL (Redis)

4.1 Justificativa

Redis foi escolhido por:

- **Performance:** Operações em memória com latência mínima
- **Simplicidade:** Estrutura de dados simples e intuitiva
- **Fila nativa:** Suporte PUSH/POP com garantia FIFO
- **Escalabilidade:** Múltiplos clientes simultâneos
- **Integração:** Fácil com Spring Data Redis

4.2 Estrutura

- **fila:matriculas** - Requisições de matrícula
- **fila:cancelamentos** - Requisições de cancelamento

5. BACKEND (Java Spring Boot)

5.1 Tecnologias

- Spring Boot 3.2.0
- Spring Security com JWT
- Spring Data JPA
- Spring Data Redis
- Jakarta Validation
- MySQL Connector/J

5.2 Arquitetura em Camadas

Plain Text

```
Controllers (REST API)
  ↓
Services (Lógica de Negócio)
  ↓
Repositories (Acesso a Dados)
  ↓
Models (Entidades JPA)
```

5.3 Autenticação

- Algoritmo: HS256 (HMAC com SHA-256)
- Expiração: 24 horas
- Criptografia de senhas: BCrypt com 10 rounds

5.4 Endpoints da API

Plain Text

```
POST /api/auth/login
GET /api/alunos
POST /api/alunos
GET /api/alunos/{id}
PUT /api/alunos/{id}
DELETE /api/alunos/{id}
GET /api/turmas
POST /api/turmas
GET /api/matriculas
POST /api/matriculas
DELETE /api/matriculas/{id}
```

5.5 Injeção de Dependência

Todas as classes utilizam @Autowired do Spring, melhorando desacoplamento e testabilidade.

6. FRONTEND (HTML/CSS/JavaScript)

6.1 Características

- Responsivo (mobile-first)
- Moderno com gradientes e animações
- Autenticação JWT
- CRUD completo
- Dashboard com estatísticas

6.2 Páginas

1. **Login** - Autenticação segura
 2. **Dashboard** - Estatísticas em tempo real
 3. **Alunos** - CRUD com busca
 4. **Turmas** - CRUD com grid
 5. **Matrículas** - Registrar, cancelar, validar
-

7. VALIDAÇÕES

7.1 Validações de Entrada

- Nome: 3-150 caracteres, apenas letras
- CPF: Formato XXX.XXX.XXX-XX ou 11 dígitos
- Email: Formato válido
- Telefone: Formato (11) 99999-9999
- Idade: 0-150 anos

7.2 Validações de Negócio

- CPF único
 - Turma com vagas disponíveis
 - Aluno sem matrícula ativa
 - Integridade referencial
-

8. SEGURANÇA

- JWT para autenticação
- BCrypt para criptografia de senhas

- CORS habilitado
 - Validação de entrada em múltiplos níveis
 - Usuário dedicado no MySQL (sem root)
 - Transações ACID para operações críticas
-

9. RESULTADOS

Todos os requisitos técnicos atendidos Sistema funcional e pronto para produção
Código limpo e bem organizado Documentação completa Interface profissional

10. CONCLUSÃO

O Sistema de Gerenciamento de Matrículas Escolares demonstra aplicação prática de conceitos avançados de banco de dados, incluindo modelagem relacional, índices, triggers, views, procedures, functions, integração com NoSQL, autenticação segura, API REST e interface responsiva.

O projeto é funcional, profissional e pronto para produção, atendendo a todos os requisitos técnicos do trabalho final.