

ro# Guia Técnico do Projeto

Documento técnico do backend (Java 8) e frontend (Angular) com detalhes de build, execução, contratos de API, segurança, persistência e migrações.

## Stack e versões

- Java 8 (target 1.8)
- Spring Boot 2.7.18 (Web, Data JPA, Validation, Security)
- H2 Database (memória) + Flyway
- Swagger/OpenAPI (springdoc-openapi-ui 1.7.0)
- Angular 17 (opcional), Bootstrap 5

## Backend – Execução

- Base URL: `http://localhost:8080`
- Swagger UI: `http://localhost:8080/swagger-ui.html`
- H2 Console: `http://localhost:8080/h2-console` (JDBC `jdbc:h2:mem:testdb`, user `sa`, senha vazia)
- Autenticação: HTTP Basic `admin / admin123`

## Build/Run

```
mvn clean package
mvn spring-boot:run
```

## application.properties (chaves relevantes)

- `spring.jpa.hibernate.ddl-auto=validate`
- `spring.jpa.show-sql=true`
- `spring.h2.console.enabled=true`
- `spring.flyway.enabled=true`

## Migrações (Flyway)

- Local: `src/main/resources/db/migration`
- `V1__init.sql` cria `users` e `cards`.

DDL efetiva:

```
CREATE TABLE users (
    id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    nome VARCHAR(255),
    email VARCHAR(255) NOT NULL UNIQUE,
    senha VARCHAR(255)
);

CREATE TABLE cards (
    id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    numero_cartao BIGINT,
    nome VARCHAR(255),
    status BOOLEAN,
    tipo_cartao VARCHAR(50),
    user_id BIGINT,
```

```
CONSTRAINT fk_cards_user FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

## Modelo (JPA)

- User ( users ): id , nome , email (unique, not null), senha ; @OneToMany para Card (cascade ALL, orphanRemoval=true)
- Card ( cards ): id , numeroCartao , nome , status , tipoCartao (ENUM), user ( @ManyToOne )

## Repositórios (inclui nativeQuery)

- UserRepository#findByEmailNative(String email) → SELECT \* FROM users u WHERE u.email = :email
- CardRepository#findAllByUserIdNative(Long userId) → SELECT \* FROM cards c WHERE c.user\_id = ?1

## Segurança

- HTTP Basic (in-memory)
- Público: /v3/api-docs/\*\* , /swagger-ui.html , /swagger-ui/\*\* , /h2-console/\*\*

## Contratos REST

### Usuários

- GET /usuarios → 200 [UserDTO]
- GET /usuarios/{id} → 200 UserDTO | 404
- POST /usuarios → 201/200 UserDTO
- PUT /usuarios/{id} → 200 UserDTO | 404
- DELETE /usuarios/{id} → 204 | 404

UserDTO :

```
{
  "id": number?,
  "nome": string,
  "email": string,
  "senha": string?,
  "cartoes": CardDTO[]?
}
```

### Cartões por usuário

- POST /usuarios/{id}/cartoes → 201/200 CardDTO
- DELETE /usuarios/{id}/cartoes/{cartaoId} → 204

### Cartões (geral)

- GET /cartoes → 200 [CardDTO]
- POST /cartoes → 201/200 CardDTO
- PUT /cartoes/{id}/status?status=true|false → 200 CardDTO
- DELETE /cartoes/{id} → 204

CardDTO :

```
{
  "id": number?,
  "numeroCartao": number,
  "nome": string,
  "status": boolean?,
  "tipoCartao": "COMUM" | "ESTUDANTE" | "TRABALHADOR"
}
```

## Erros

- 400 (Bean Validation) retorna { campo: mensagem }
- 404 para recursos inexistentes

## Frontend (opcional)

```
cd frontend
npm install
npm start
```

## Estrutura

```
src/main/java ...
src/main/resources ...
docs/
frontend/
```