**Passo: configuração de segurança**

@Configuration

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Autowired

private Environment env;

@Override

protected void configure(HttpSecurity http) throws Exception {

if (Arrays.asList(env.getActiveProfiles()).contains("test")) {

http.headers().frameOptions().disable();

}

http.cors().and().csrf().disable();

http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);

http.authorizeRequests().anyRequest().permitAll();

}

@Bean

CorsConfigurationSource corsConfigurationSource() {

CorsConfiguration configuration = new CorsConfiguration().applyPermitDefaultValues();

configuration.setAllowedMethods(Arrays.asList("POST", "GET", "PUT", "DELETE", "OPTIONS"));

final UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();

source.registerCorsConfiguration("/\*\*", configuration);

return source;

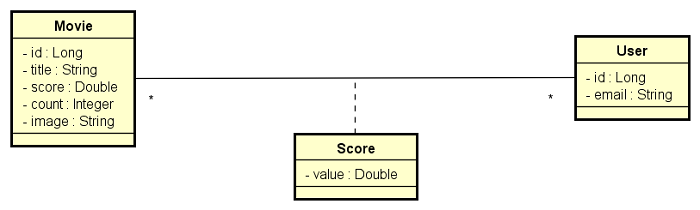
}

}

* **COMMIT: Security config**

**Passo: criar as entidades e o seed do banco**

**Modelo conceitual**

[](https://raw.githubusercontent.com/devsuperior/bds-assets/main/sds/dsmovie-dominio.png)

**application.properties**

spring.profiles.active=test

spring.jpa.open-in-view=false

**application-test.properties**

# Dados de conexão com o banco H2

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.username=sa

spring.datasource.password=

# Configuração do cliente web do banco H2

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

# Configuração para mostrar o SQL no console

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

**import.sql**

INSERT INTO tb\_user(email) VALUES ('maria@gmail.com');

INSERT INTO tb\_user(email) VALUES ('joao@gmail.com');

INSERT INTO tb\_user(email) VALUES ('ana@gmail.com');

INSERT INTO tb\_user(email) VALUES ('lucia@gmail.com');

INSERT INTO tb\_user(email) VALUES ('joaquim@gmail.com');

INSERT INTO tb\_movie(score, count, title, image) VALUES (4.5, 2, 'The Witcher', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/jBJWaqoSCiARWtfV0GlqHrcdidd.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (3.3, 3, 'Venom: Tempo de Carnificina', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/vIgyYkXkg6NC2whRbYjBD7eb3Er.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'O Espetacular Homem-Aranha 2: A Ameaça de Electro', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/u7SeO6Y42P7VCTWLhpnL96cyOqd.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Matrix Resurrections', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/hv7o3VgfsairBoQFAawgaQ4cR1m.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Shang-Chi e a Lenda dos Dez Anéis', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/cinER0ESG0eJ49kXlExM0MEWGxW.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Django Livre', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/2oZklIzUbvZXXzIFzv7Hi68d6xf.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Titanic', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/yDI6D5ZQh67YU4r2ms8qcSbAviZ.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'O Lobo de Wall Street', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/cWUOv3H7YFwvKeaQhoAQTLLpo9Z.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Aves de Rapina: Arlequina e sua Emancipação Fantabulosa', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/jiqD14fg7UTZOT6qgvzTmfRYpWI.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Rogue One: Uma História Star Wars', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/6t8ES1d12OzWyCGxBeDYLHoaDrT.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Star Wars: A Guerra dos Clones', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/uK15I3sGd8AudO9z6J6vi0HH1UU.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Star Wars: Episódio I - A Ameaça Fantasma', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/36LnijfQCOC89rCMOhn2OINXROI.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Vingadores: Ultimato', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/7RyHsO4yDXtBv1zUU3mTpHeQ0d5.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Thor', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/cDJ61O1STtbWNBwefuqVrRe3d7l.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Cisne Negro', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/hqh5O4KssfJWI62HGAgrjHXbxhD.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'O Silêncio dos Inocentes', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/mfwq2nMBzArzQ7Y9RKE8SKeeTkg.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Clube da Luta', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/hZkgoQYus5vegHoetLkCJzb17zJ.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Guerra Mundial Z', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/31VpBgUX5O4Z3dn5ZbX8HLqoXH3.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Harry Potter e as Relíquias da Morte - Parte 1', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/vcrgU0KaNj5mKUCIQSUdiQwTE9y.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Harry Potter e a Pedra Filosofal', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/lvOLivVeX3DVVcwfVkxKf0R22D8.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Alice no País das Maravilhas', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/qNdlZgz9yoSJ8f0YxQWfKGPoVV.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Animais Fantásticos e Onde Habitam', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/8Qsr8pvDL3s1jNZQ4HK1d1Xlvnh.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'A Teoria de Tudo', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/kq2MHrRfH6RTfkvyDEmYLmGHE6U.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'O Livro de Boba Fett', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/sjx6zjQI2dLGtEL0HGWsnq6UyLU.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'O Último Duelo', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/4LrL40XecjGLRpX5I2gzMTUt04l.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Interestelar', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/rAiYTfKGqDCRIIqo664sY9XZIvQ.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Contato', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/yFkUPqBuUnbhYbQL8VFpTrAT9za.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Duna', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/jYEW5xZkZk2WTrdbMGAPFuBqbDc.jpg');

INSERT INTO tb\_movie(score, count, title, image) VALUES (0, 0, 'Aquaman', 'https://www.themoviedb.org/t/p/w533\_and\_h300\_bestv2/2cUsDz4TzFYHrKktT1bKHHQ7Cgm.jpg');

INSERT INTO tb\_score(movie\_id, user\_id, value) VALUES (1, 1, 5.0);

INSERT INTO tb\_score(movie\_id, user\_id, value) VALUES (1, 2, 4.0);

INSERT INTO tb\_score(movie\_id, user\_id, value) VALUES (2, 1, 3.0);

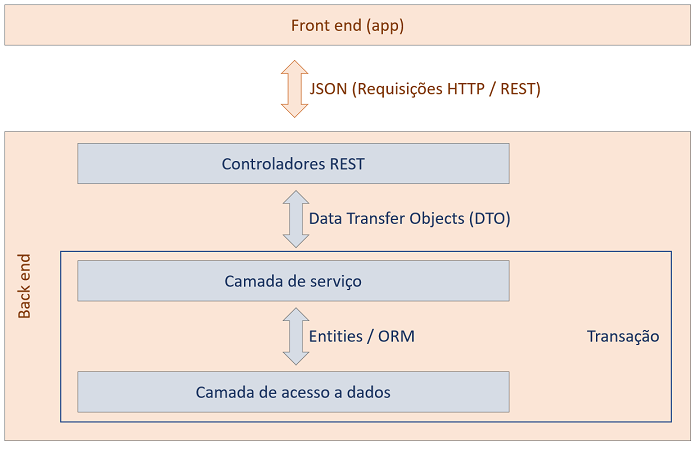
INSERT INTO tb\_score(movie\_id, user\_id, value) VALUES (2, 2, 3.0);

INSERT INTO tb\_score(movie\_id, user\_id, value) VALUES (2, 3, 4.0);

* **COMMIT: Domain model, database seed**

**Passo: Busca de filmes**

**Padrão camadas adotado**

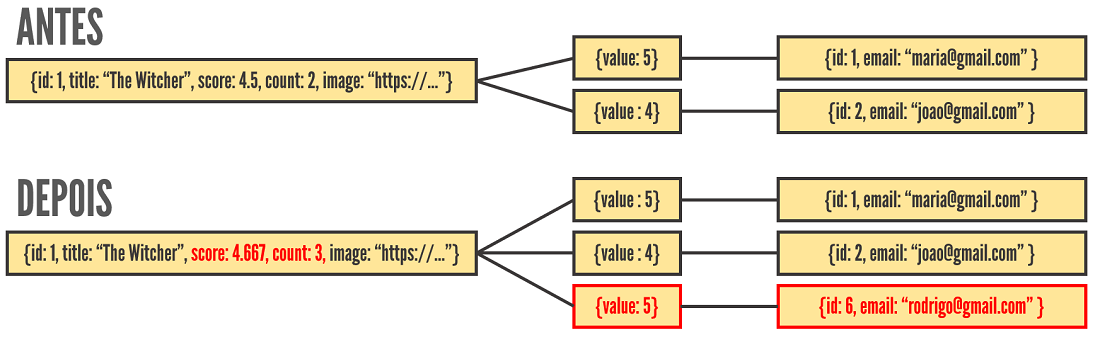
[](https://github.com/devsuperior/bds-assets/raw/main/sds/padrao-camadas.png)

* Criar repository
* Criar DTO
* Criar service
* Criar controller
* **COMMIT: Find movies**

**Passo: Salvar avaliação**

**Lógica:**

1. Informar email, id do filme e valor da avaliação (1 a 5).
2. Recuperar usuário do banco de dados pelo email. Se o usuário não existir, insira no banco.
3. Salvar a avaliação do usuário para o dado filme.
4. Recalcular a avaliação média do filme e salvar no banco de dados.

[](https://raw.githubusercontent.com/devsuperior/bds-assets/main/sds/dsmovie-objs.png)

* **COMMIT: Save score**

**Passo: Validação no Postgres local**

* Criar três perfis de projeto: test, dev, prod
* Gerar script SQL no perfil dev
* Testar projeto no banco Postgres local

**application-dev.properties**

#spring.jpa.properties.javax.persistence.schema-generation.create-source=metadata

#spring.jpa.properties.javax.persistence.schema-generation.scripts.action=create

#spring.jpa.properties.javax.persistence.schema-generation.scripts.create-target=create.sql

#spring.jpa.properties.hibernate.hbm2ddl.delimiter=;

spring.datasource.url=jdbc:postgresql://localhost:5432/dsmovie

spring.datasource.username=postgres

spring.datasource.password=1234567

spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation=true

spring.jpa.hibernate.ddl-auto=none

**application-prod.properties**

spring.datasource.url=${DATABASE\_URL}

**system.properties**

java.runtime.version=17

* **COMMIT: First homolog**

**Passo: Implantação no Heroku**

* Criar app no Heroku
* Provisionar banco Postgres
* Definir variável APP\_PROFILE=prod
* Conectar ao banco via pgAdmin
* Criar seed do banco

heroku -v

heroku login

heroku git:remote -a <nome-do-app>

git remote -v

git subtree push --prefix backend heroku main

**Passo: implantação no Netlify**

* Deploy básico
  + Base directory: frontend
  + Build command: yarn build
  + Publish directory: frontend/build
* Arquivo \_redirects

/\* /index.html 200

* Configurações adicionais
  + Site settings -> Domain Management: (colocar o nome que você quiser)
  + Deploys -> Trigger deploy