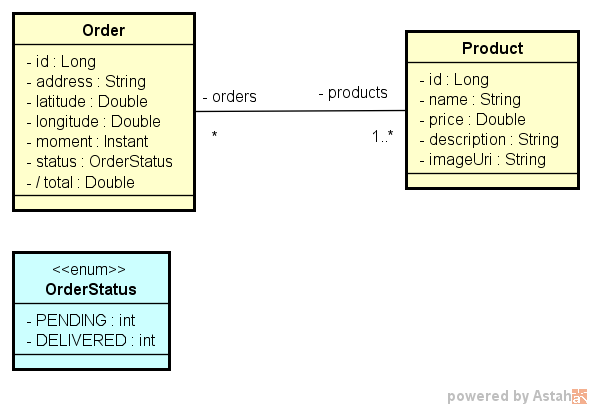
**Aula 1 - Back end**

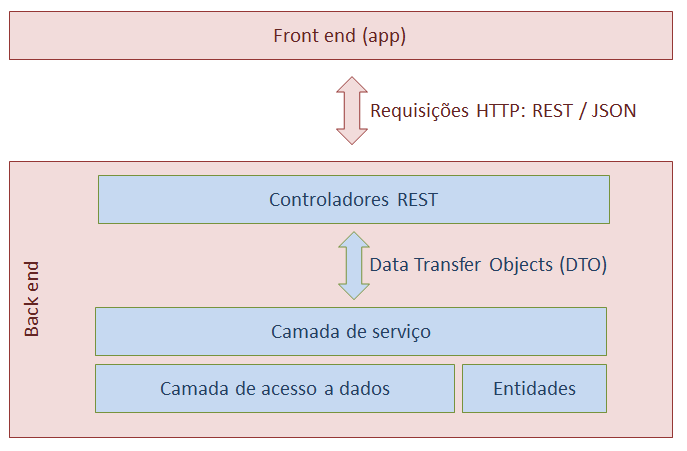
**Nivelamento: back end, front end, padrão camadas, MVC, REST**

[](https://youtu.be/b8uLFfzcVQ8)

**Modelo conceitual**

[](https://raw.githubusercontent.com/devsuperior/sds2/master/assets/modelo-conceitual.png)

**Padrão camadas adotado**

[](https://raw.githubusercontent.com/devsuperior/sds2/master/assets/camadas.png)

**Checklist**

* Setup inicial do projeto
  + Dependências
  + Arquivos .properties
  + Configuração de segurança
* Modelo de domínio
  + Entidades e relacionamentos
  + Mapeamento objeto-relacional
  + Seed
* Criar endpoints
  + [GET] /products
  + [GET] /orders
  + [POST] /orders
  + [PUT] /orders/{id}/delivered
* Validar perfil dev
  + Base de dados Postgres local
  + Testar todos endpoints
* Preparar projeto para implantação
  + Arquivo system.properties
  + Profile prod -> commit
* Implantar projeto no Heroku
  + Criar app e provisionar Postgres
  + Criar base de dados remota
  + Executar comandos no Heroku CLI

heroku login

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

git remote -v

git subtree push --prefix backend heroku main

**Dependências Maven**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

**Classe de 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;

}

}

**Arquivos .properties de cada profile do projeto**

**application.properties**

spring.profiles.active=test

spring.jpa.open-in-view=false

**application-test.properties**

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

spring.datasource.username=sa

spring.datasource.password=

spring.h2.console.enabled=true

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

**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/dsdeliver

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}

**Script SQL de instanciação da base de dados**

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Pizza Bacon', 49.9, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/pizza\_bacon.jpg', 'Pizza de bacon com mussarela, orégano, molho especial e tempero da casa.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Pizza Moda da Casa', 59.9, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/pizza\_moda.jpg', 'Pizza à moda da casa, com molho especial e todos ingredientes básicos, e queijo à sua escolha.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Pizza Portuguesa', 45.0, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/pizza\_portuguesa.jpg', 'Pizza Portuguesa com molho especial, mussarela, presunto, ovos e especiarias.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Risoto de Carne', 52.0, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/risoto\_carne.jpg', 'Risoto de carne com especiarias e um delicioso molho de acompanhamento.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Risoto Funghi', 59.95, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/risoto\_funghi.jpg', 'Risoto Funghi feito com ingredientes finos e o toque especial do chef.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Macarrão Espaguete', 35.9, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/macarrao\_espaguete.jpg', 'Macarrão fresco espaguete com molho especial e tempero da casa.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Macarrão Fusili', 38.0, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/macarrao\_fusili.jpg', 'Macarrão fusili com toque do chef e especiarias.');

INSERT INTO tb\_product (name, price, image\_Uri, description) VALUES ('Macarrão Penne', 37.9, 'https://raw.githubusercontent.com/devsuperior/sds2/master/assets/macarrao\_penne.jpg', 'Macarrão penne fresco ao dente com tempero especial.');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (0, -23.561680, -46.656139, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T10:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (1, -22.946779, -43.217753, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T15:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (0, -25.439787, -49.237759, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T16:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (0, -23.561680, -46.656139, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T12:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (1, -23.561680, -46.656139, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T08:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (0, -23.561680, -46.656139, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T14:00:00Z');

INSERT INTO tb\_order (status, latitude, longitude, address, moment) VALUES (0, -23.561680, -46.656139, 'Avenida Paulista, 1500', TIMESTAMP WITH TIME ZONE '2021-01-01T09:00:00Z');

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (1 , 1);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (1 , 4);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (2 , 2);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (2 , 5);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (2 , 8);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (3 , 3);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (3 , 4);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (4 , 2);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (4 , 6);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (5 , 4);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (5 , 6);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (6 , 5);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (6 , 1);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (7 , 7);

INSERT INTO tb\_order\_product (order\_id, product\_id) VALUES (7 , 5);