Manual de Técnico

Índice

Nombre del sistema: E commerce GT
Autor: Brígido Josías Alvarado Pec RA | 202230251

Ruta Netlify: https://prismatic-frangipane-c2bb28.netlify.app
Ruta Ngrok: https://jade-flinty-dayton.ngrok-free.dev

Introducción

El sistema se desarrolla con un cliente en Angular, un Servidor en Spring Boot y una base de datos en PostgresQL. El sistema permite la compra y venta de productos por parte de usuarios, manejo de paquetes y pagos únicamente con tarjeta, además de la calificación y comentarios a productos del catálogo.

Objetivos

- Administrar la venta y compra de productos.
- Generar ganancias a partir de comisiones en ventas de productos.
- Mantener un entorno seguro y confiable.

Alcance

Incluye administración de usuarios de tipo común, administradores, logística y moderadores, administración de publicación de productos, calificación, comentarios y sanciones a usuarios y compra con tarjeta.

Arquitectura

Cliente: Angular: Angular CLI: 18.2.6

Servidor: Spring Boot 3.5.6, Java 22

Frontend: Bootstrap 5, HTML5, Typescript

Base de datos: Postgresql 18.0

Diagrama de Componentes

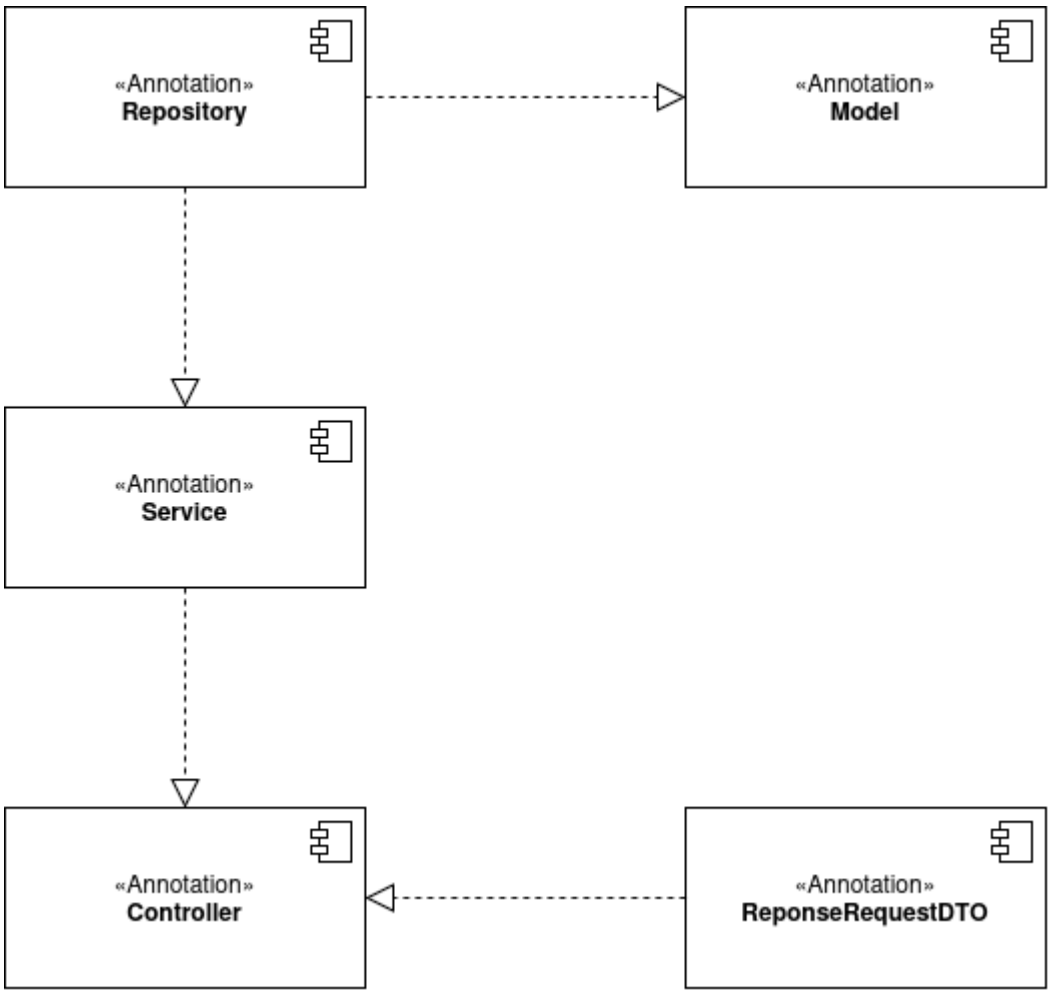


Diagrama de Paquetes

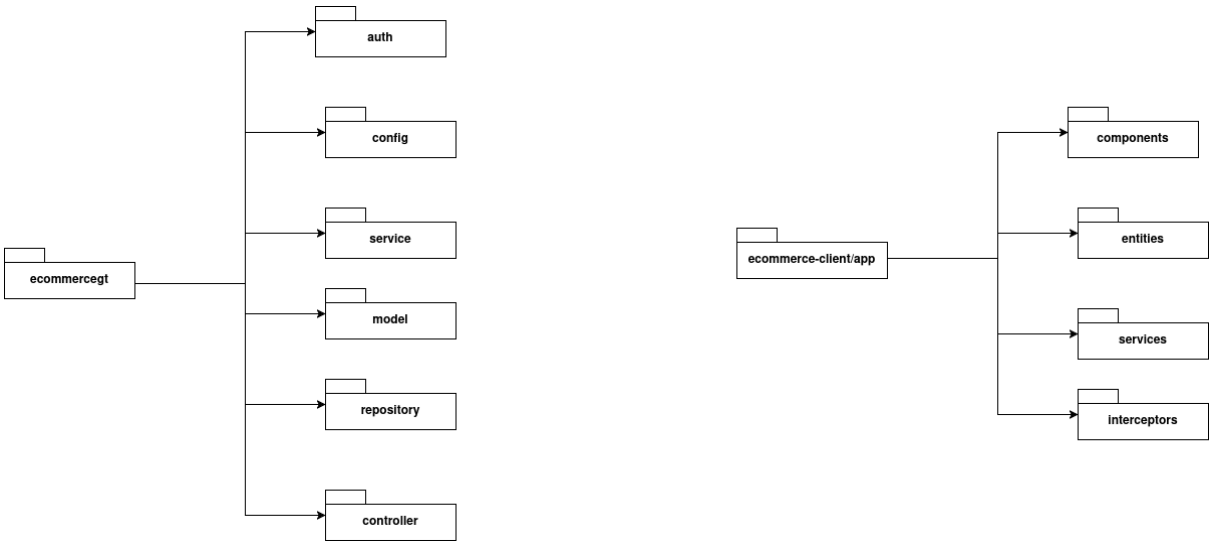


Diagrama de Despliegue

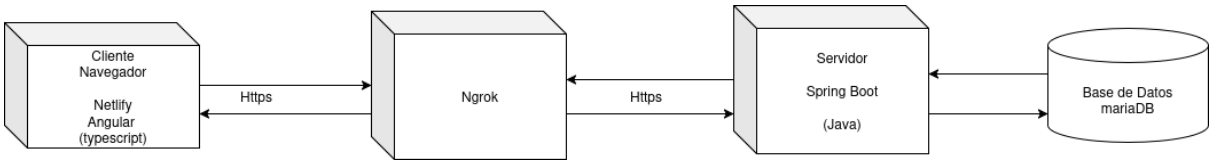


Diagrama de Entidad Relación Notación Peter Chen

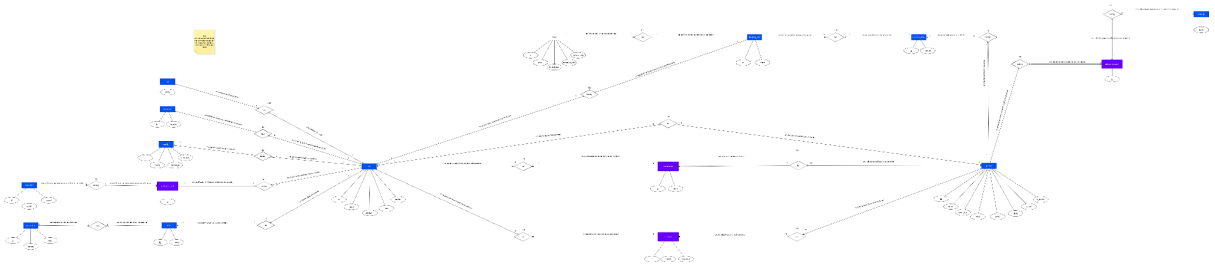
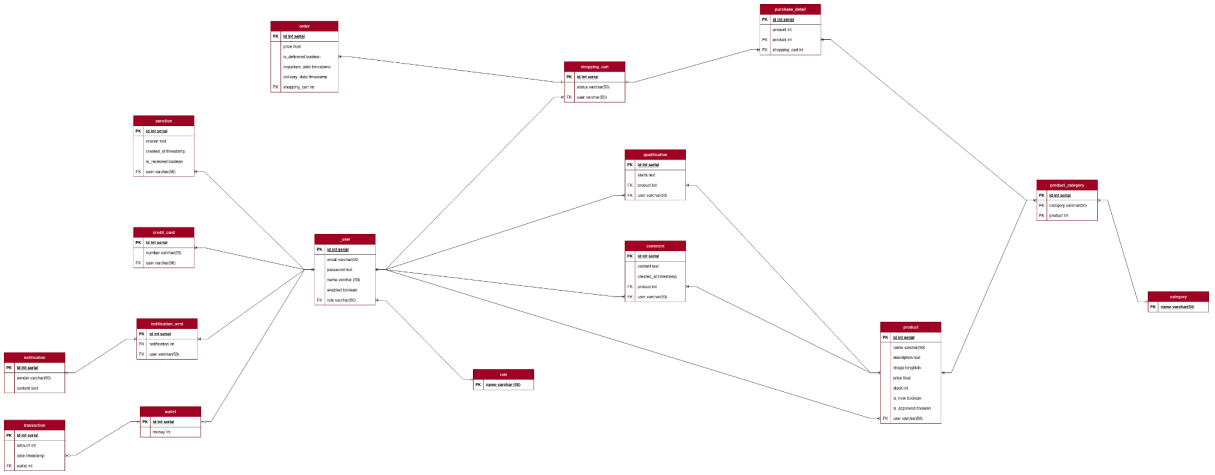


Diagrama de Tablas UML



DDL

```
drop database if exists e_commerce_gt;

create database e_commerce_gt;

create table role
(
  name varchar(50) primary key
);

create table _user
(
  id       serial primary key,
  email    varchar(50) unique not null,
  password text            not null,
  name     varchar(50)       not null,
  enabled  boolean           not null default true,
  role     varchar(50)       not null,

  constraint fk_role_in_user
     foreign key (role)
        references role (name)
        on delete cascade
        on update cascade

);

create table wallet
(
```

```
  id    serial primary key,
  money float           not null,
  _user varchar(50) unique not null,

  constraint fk_user_in_wallet
     foreign key (_user)
        references _user (email)
        on delete cascade
        on update cascade
);

create table transaction
(
  id     serial primary key,
  amount float not null,
  date timestamp not null  default current_timestamp,
  wallet int   not null,

  constraint fk_wallet_in_transaction
     foreign key (wallet)
        references wallet (id)
        on delete cascade
        on update cascade
);

create table notification
(
  id     serial primary key,
  sender  varchar(50) not null,
  content text       not null,
  subject text       not null,
  _user   varchar(50) not null,

  constraint fk_user_int_notification
     foreign key (_user)
     references _user(email)
     on delete cascade
     on update cascade
);


create table credit_card
(
  id     serial primary key,
  number varchar(20) not null,
  _user  varchar(50) not null,
```

```
  constraint fk_user_in_credit_card
    foreign key (_user)
      references _user (email)
      on delete cascade
      on update cascade,

  constraint unique_number_and_user
    unique (number, _user)
);

create table sanction
(
  id         serial primary key ,
  reason     text       not null ,
  created_at  timestamp   not null default current_timestamp,
  end_at     timestamp   not null ,
  _user      varchar(50) not null ,

  constraint fk_user_in_sanction
    foreign key (_user)
      references _user (email)
      on delete cascade
      on update cascade
);

create table product
(
  id         serial primary key,
  name       varchar(50) not null,
  description text       not null,
  image_url   text       not null,
  price      float      not null,
  stock      int        not null,
  is_new     boolean    not null,
  is_approved boolean    not null default false,
  is_revised boolean     not null default false,
  _user      varchar(50) not null,

  constraint fk_user_in_product
    foreign key (_user)
      references _user (email)
      on delete cascade
      on update cascade
);

create table comment
(
```

```
  id        serial primary key,
  content   text       not null,
  created_at timestamp default current_timestamp,
  product   int        not null,
  _user     varchar(50) not null,

  constraint fk_product_int_comment
     foreign key (product)
        references product (id)
        on delete cascade
        on update cascade,

  constraint fk_user_in_comment
     foreign key (_user)
        references _user (email)
        on delete restrict
        on update cascade
);

create table qualification
(
  id     serial primary key,
  starts  int        not null,
  product int        not null,
  _user   varchar(50) not null unique ,

  constraint fk_product_in_qualification
     foreign key (product)
        references product (id)
        on delete cascade
        on update cascade,

  constraint fk_user_in_qualification
     foreign key (_user)
        references _user (email)
        on delete restrict
        on update cascade,

  constraint unique_product_user
     unique ( product, _user )
);

create table category
(
  name varchar(50) primary key
);
```

```
create table product_category
(
  id      serial primary key,
  category varchar(50) not null,
  product  int        not null,

  constraint fk_category_in_product_category
    foreign key (category)
      references category (name)
      on delete cascade
      on update cascade,

  constraint fk_product_in_product_category
    foreign key (product)
      references product (id)
      on delete cascade
      on update cascade
);

create table shopping_cart
(
  id     serial primary key,
  status boolean     not null default true,
  _user  varchar(50) not null,

  constraint fk_user_in_shopping_cart
    foreign key (_user)
      references _user (email)
      on delete cascade
      on update cascade
);

create table purchase_detail
(
  id          serial primary key,
  amount      int not null,
  product     int not null,
  shopping_cart int not null,

  constraint fk_product_in_purchase_detail
    foreign key (product)
      references product (id)
      on delete cascade
      on update cascade,

  constraint fk_shopping_cart_in_purchase_detail
    foreign key (shopping_cart)
```

```
        references shopping_cart (id)
        on delete cascade
        on update cascade,

  constraint unique_product_shopping_cart
     unique (shopping_cart, product)
);

create table package
(
  id           serial primary key,
  total        float                                  not null,
  is_delivered   boolean   default false                    not null,
  is_revised     boolean   default false                  not null,
  departure_date timestamp default current_timestamp                not null,
  delivery_date  timestamp default (current_timestamp + interval '5 days') not
null,
  deliver_at     timestamp null ,
  shopping_cart  int                                    not null,

  constraint fk_shopping_cart_in_package
     foreign key (shopping_cart)
        references shopping_cart (id)
        on delete cascade
        on update cascade
);
```

Seguridad

- Validación de formularios.
- Autenticación con sesiones mediante JWT.
- Hash de contraseñas.