

**App Ferre Hogar**

**Informe técnico de requerimientos**

José David Villero Hernández

Sergio Andrés Manrique Triana

**Introducción:**

Ferre Hogar es una ferretería emergente de origen familiar, su objetivo es posesionarse como un referente a nivel local de artículos y productos de ferretería, por lo que ha decidido abordar el mercado virtual en busca aumentar su participación, logrando así cumplir su objetivo,

El software desarrollado corresponde a un sitio web (**App Ferre Hogar**) que en si es una aplicación web donde los clientes pueden registrarse y acceder a un catálogo, fácil y rápido, además en donde el propietario puede actualizar y editar los productos, podrá listar los usuarios registrados, así como también los podrá editar o eliminar, gestionar los pedidos generados y generar reportes de ventas, esto atendiendo los requisitos previamente definidos por la empresa **Ferre Hogar**.

El Aplicativo desarrollado cuenta con sus respectivos manuales de usuario – cliente, usuario – administrador y técnico, con el fin de brindar información y lineamientos del funcionamiento de la **App Ferre Hogar**.

**REQUERIMIENTO DE DESARROLLO**

1. **Introducción**

Buscando obtener nuevos clientes, crecer, y ser un referente a nivel local; **FerreHogar** desea implementar un sitio web para que los usuarios puedan realizar la compra de artículos de ferretería. También se desea tener un software de control de inventarios con reporte de los productos vendidos y tener una base de datos solida de clientes.

1. **Resumen de requisitos del negocio**

* Gestión de usuario agregar, editar, eliminar (Usuario -administrador).
* Implementación carrito compras.
* Proceso de pago.
* Generación de factura.
* Cerrar sesión.
* Acceso a diferentes redes sociales de la Organización.
* Creación roles Usuario y Usuario-administrador.
* Registro de usuario.
* Inicio sesión.
* Consulta catálogo de productos.
* Gestión de productos agregar, editar, eliminar (Usuario-administrador).

1. **Resumen del alcance**

El desarrollo de la APP **FERRE HOGAR**, estará limitado al cumplimiento de los requisitos funcionales que se describen en este documento, cualquier desarrollo adicional estará ligado a una nueva solicitud de requerimiento de desarrollo y se debe realizar todo el proceso destinado para ese fin.

1. **Detalle de los requisitos**

| **# Req.** | **Descripción del Requerimiento** | **Requiere Desarrollo** |
| --- | --- | --- |
| RF1 | Creación de roles de usuario (Usuario, Usuario-administrador) | si |
| RF2 | Registro de usuario | si |
| RF3 | Inicio de sesión (login) | si |
| RF4 | Consulta catálogo de productos | si |
| RF5 | Gestión de productos agregar, editar, eliminar (Usuario-administrador). | si |
| RF6 | Gestión de usuario agregar, editar, eliminar (Usuario -administrador). | si |
| RF7 | Implementación carrito compras. | si |
| RF8 | Proceso de pago. | si |
| RF9 | Generación de factura. | si |
| RF10 | Listar pedidos generados | si |
| RF11 | Generar reporte de ventas | si |
| RF12 | Cerrar sesión. | si |
| RF13 | Acceso a diferentes redes sociales de la Organización. | si |

|  |  |  |
| --- | --- | --- |
| **# Req.** | **Descripción del Requerimiento** | **Requiere Desarrollo** |
| RNF1 | Rendimiento | si |
| RNF2 | Disponibilidad | si |
| RNF3 | Accesibilidad | si |
| RNF4 | Integridad de datos | si |
| RNF5 | Flexibilidad | si |
| RNF6 | Seguridad | si |
| RNF7 | Mantenibilidad | si |
| RNF8 | Auditabilidad | si |
| RNF9 | Usabilidad | si |
| RNF10 | Privacidad | si |

1. **Tecnología de desarrollo del producto**

Para el desarrollo de la aplicación del servidor (Backend) serán usadas las tecnologías:

* JAVA 17
* Spring Framework 6:
* Spring Security 6
* Spring MVC
* Spring Boot 3.1.0
* PostgresSQL

En cuanto a la vista, aplicación del lado del cliente (Frontend) serán usadas las tecnologías:

* Thymeleaf
* JavaScript
* HTML5
* CSS3

IDE de desarrollo:

* IDE: IntelliJ IDEA 2023

1. **Requisitos de hardware para la ejecución de la aplicación lado servidor y cliente**

**Servidor:**

* 8 GB de RAM
* Intel core i5 9th generación en adelante
* Ryzen 3 segunda generación en adelante
* Disco duro 500GB
* Sistema Operativo:

Linux: Ubuntu

**Cliente:**

* Navegador web:

Microsoft Edge

Google Crhome

Mozilla Firefox

* Teléfono Android 5.0
* Teléfono IOS 12

1. **Costo de desarrollo.**

|  |  |
| --- | --- |
| **Concepto** | **Costo Mensual Estimado** |
| Salarios de Desarrolladores | $6,000 |
| Alojamiento (Servidores) | $100-$200 |
| Herramientas de Desarrollo | $50 |
| Otros Gastos | $50 |
| **Costo Total** | **$6,200 - $6,400** |

\*Precio en dólares.

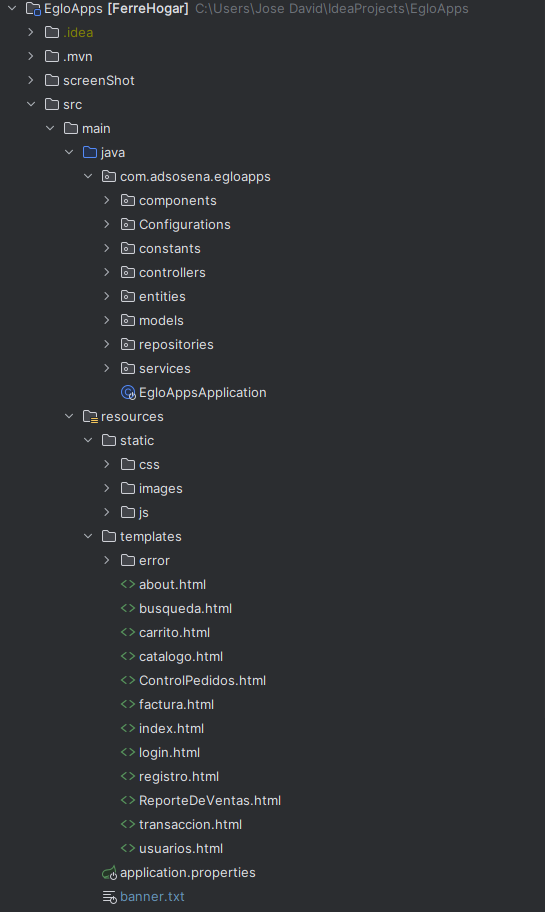
**Estructura del proyecto:**

El proyecto es realizado bajo el patrón de diseño MVC (Modelo-Vista-Controlador), este patrón implementado con Spring permite organizar nuestro proyecto en cuatro paquetes o carpetas principales:

* Controllers
* Services
* Models
* Repositories

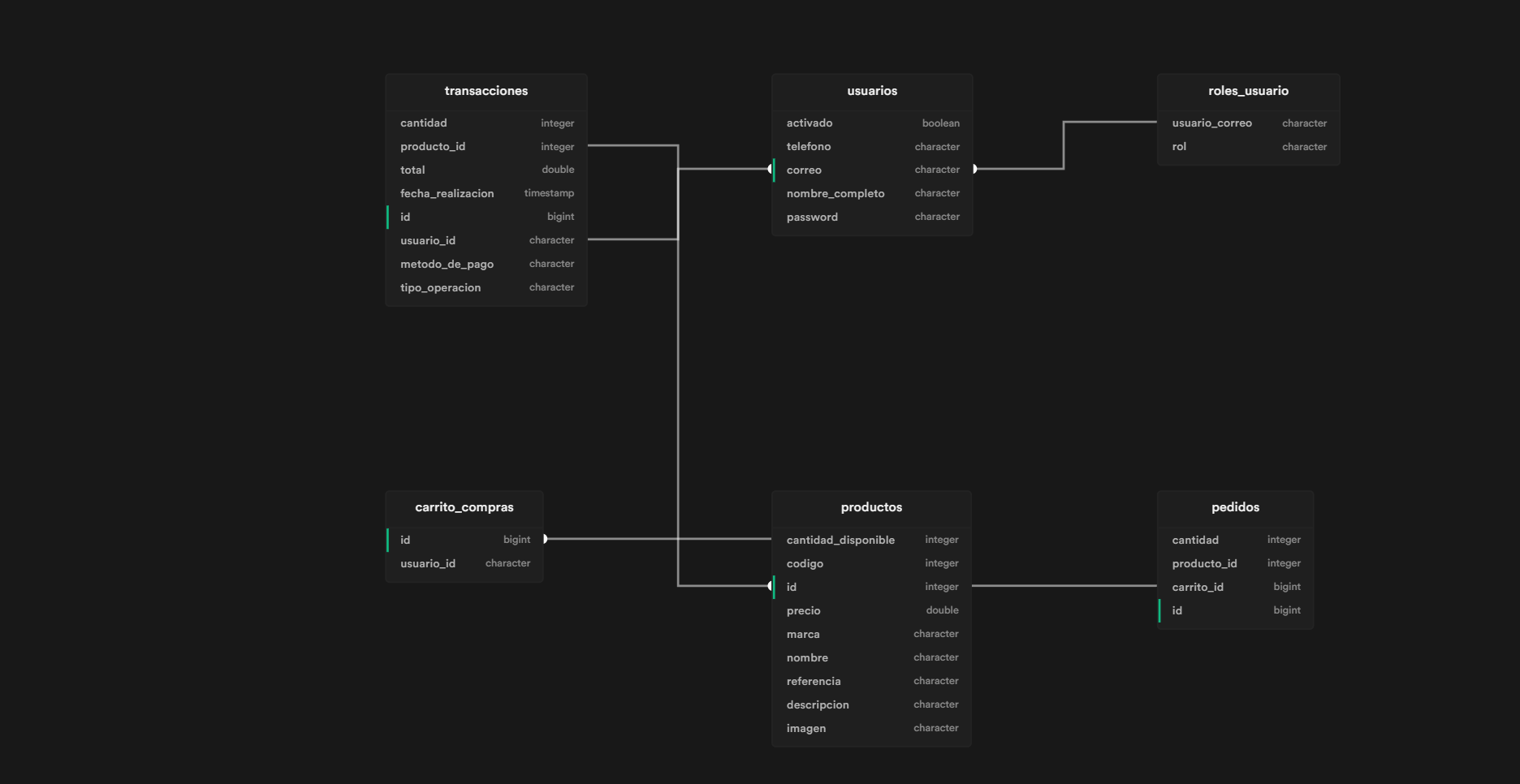
En donde cada clase dentro del paquete se encarga de una función, bien sea controlador, repositorio o modelo de una entidad. A continuación, se muestra la estructura del proyecto en el IDE IntelliJ IDEA.

**Organización de los paquetes:**



**Documentación UML:**

1. **Diagrama entidad-relación:**

****

**Código SQL:**

create table usuarios (

activado boolean not null,

telefono character,

correo character not null primary key,

nombre\_completo character not null,

"password" character not null

);

create table carrito\_compras (

id bigint not null primary key,

usuario\_id character

);

create table pedidos (

cantidad integer not null,

producto\_id integer,

carrito\_id bigint references carrito\_compras (id),

id bigint not null primary key

);

create table roles\_usuario (

usuario\_correo character references usuarios (correo),

rol character

);

create table productos (

cantidad\_disponible integer not null,

codigo integer not null,

id serial not null primary key,

precio double not null,

marca character not null,

nombre character not null,

referencia character not null,

descripcion character,

imagen character not null

);

create table transacciones (

cantidad integer not null,

producto\_id integer references productos (id),

total double,

fecha\_realizacion timestamp default now() not null,

id bigint not null primary key,

usuario\_id character references usuarios (correo),

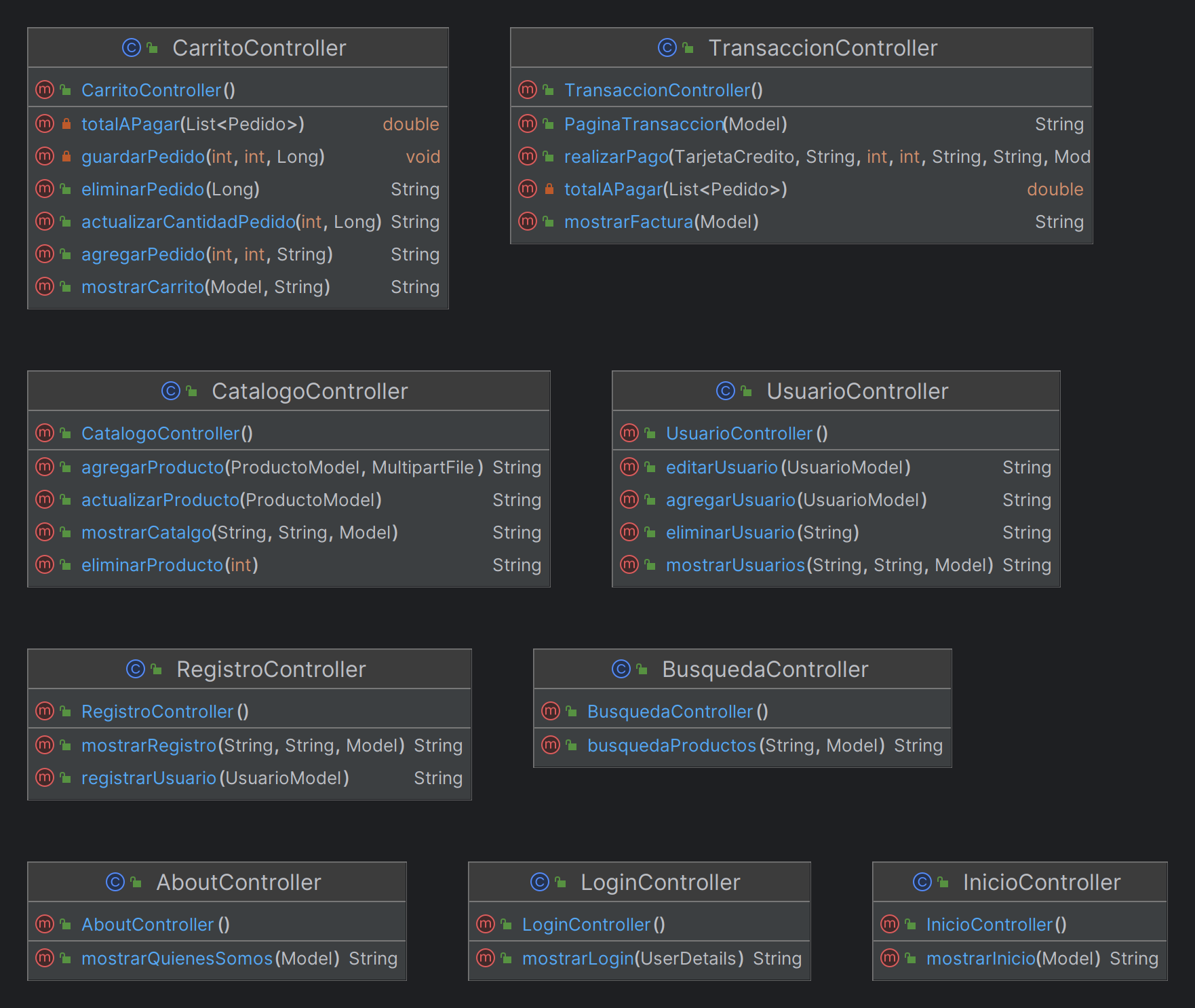
metodo\_de\_pago character,

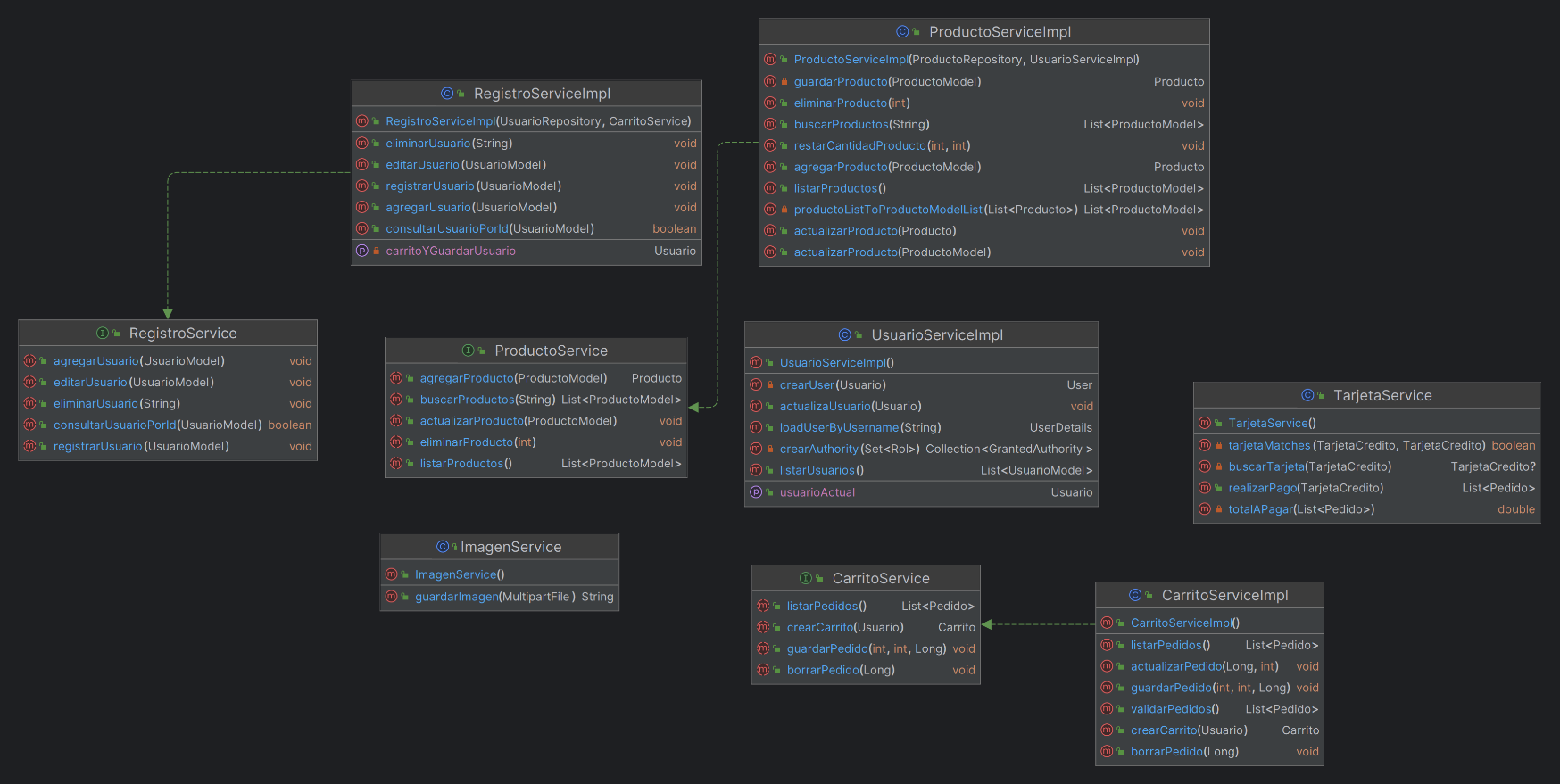
tipo\_operacion character not null

);

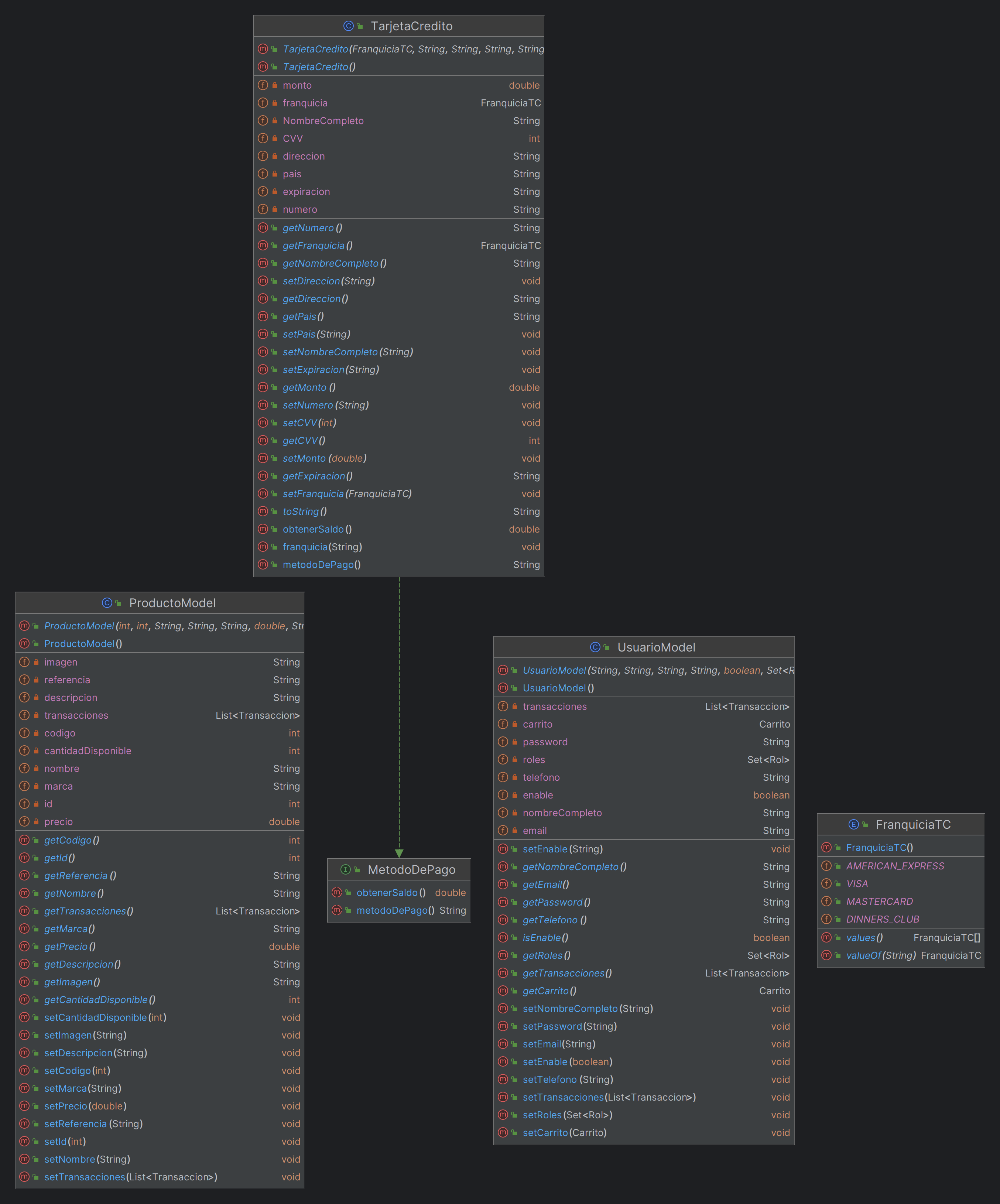
1. **Diagrama de clases:**

**Controladores:**

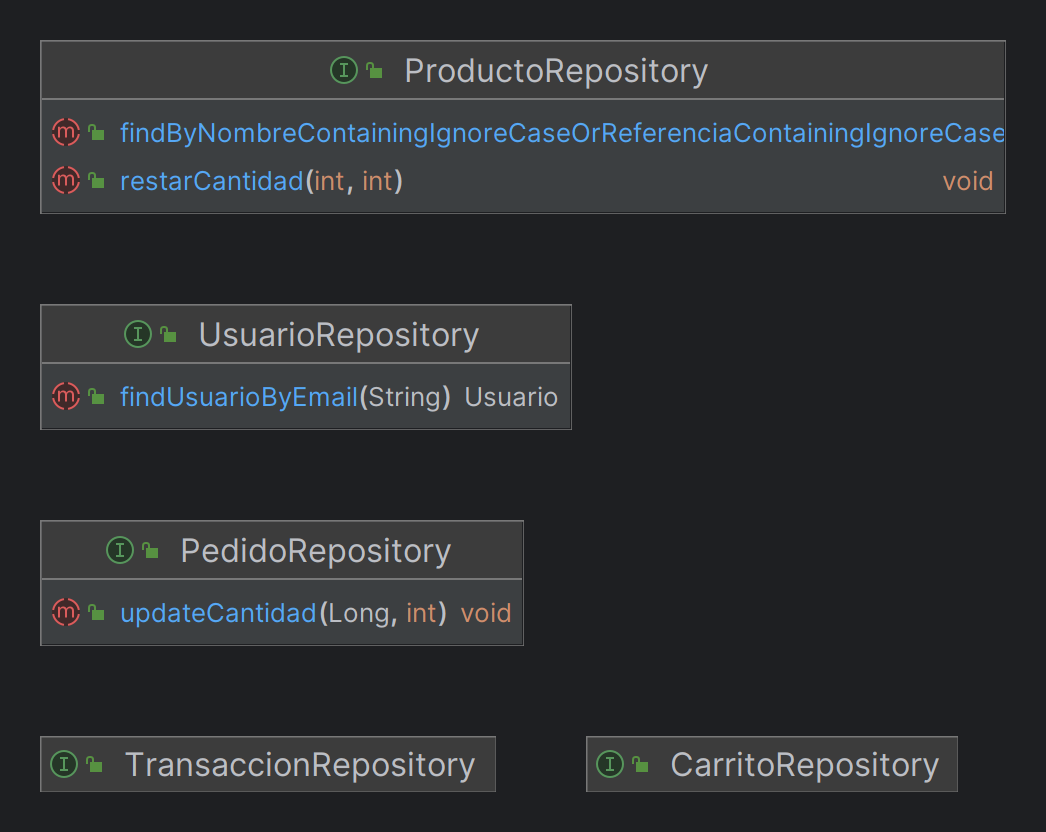
****

**Servicios:**

**Modelos:**

****

**Repositorios:**

****