Proyecto final

“ Hito 1 :

Diseño y Prototipo”

Integrantes:

Felipe Cruz

Paula Montiel

Jonatan Villarroel

1.- Bocetos vistas de proyectos

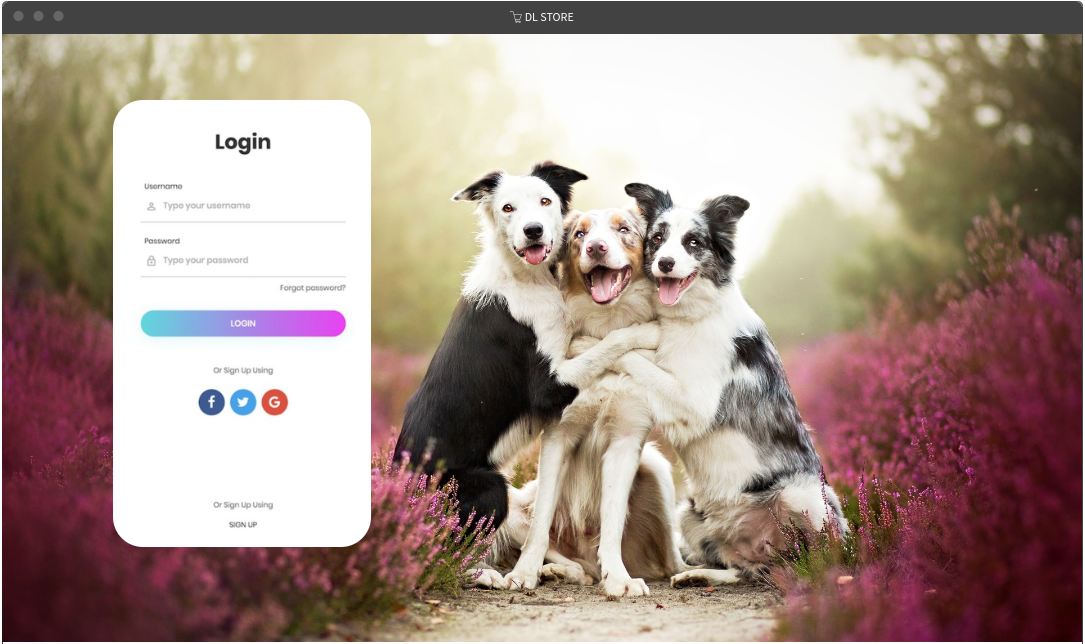
- Página Principal



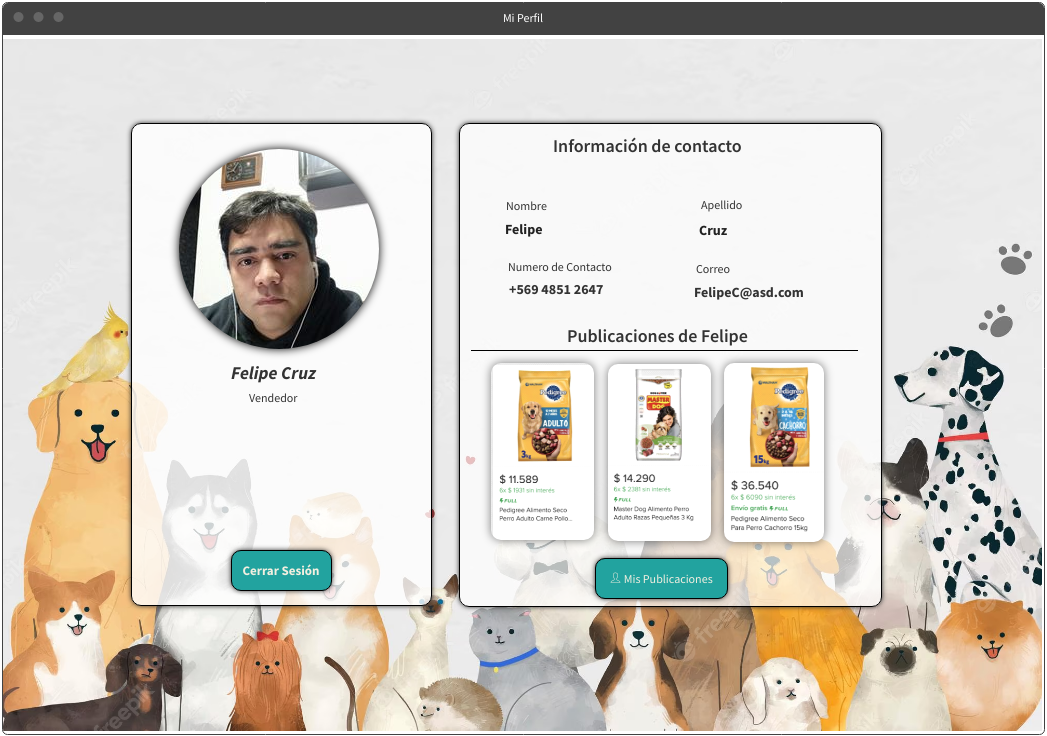
* Registro de usuario



* Inicio de sesión



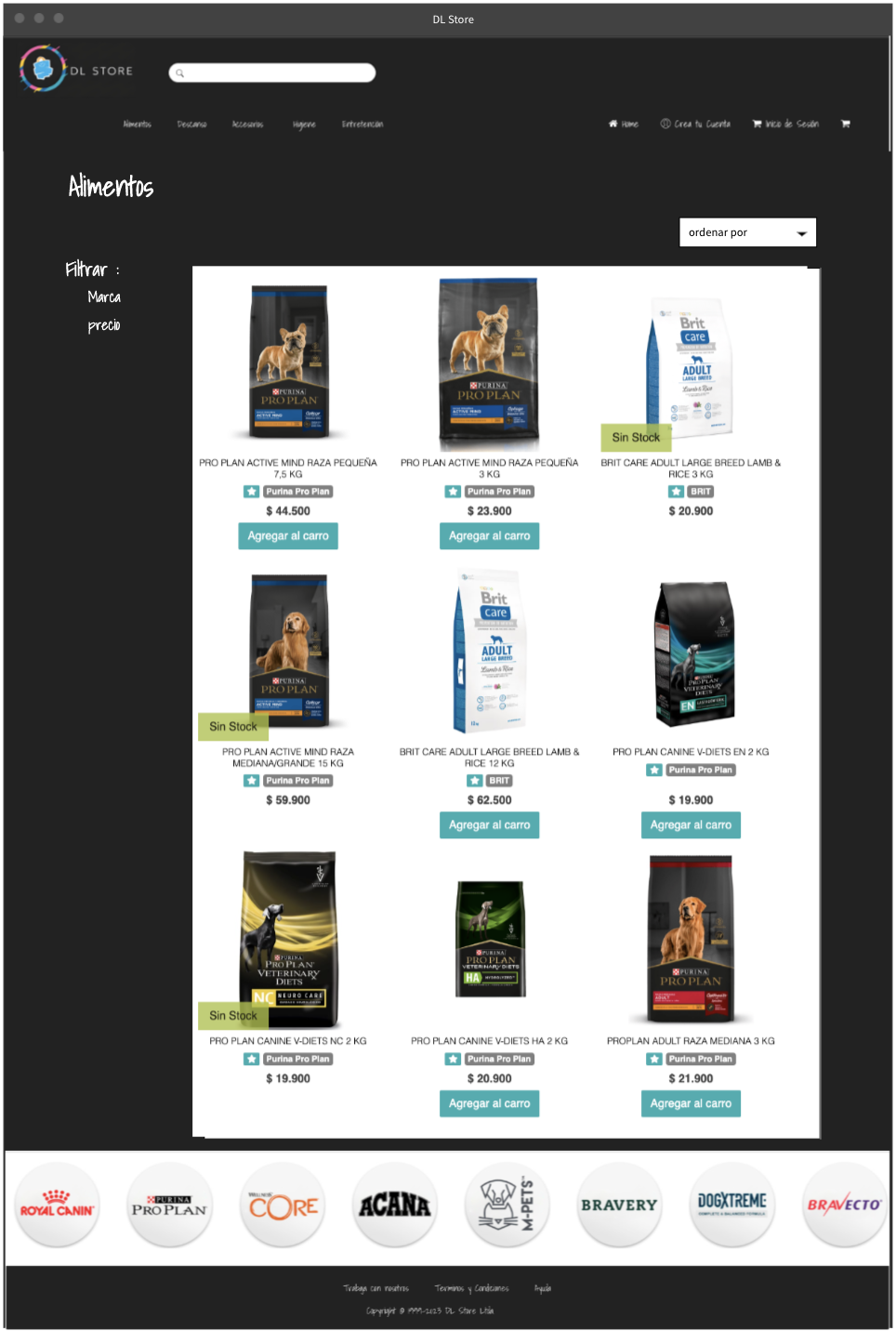
* Perfil Usuario



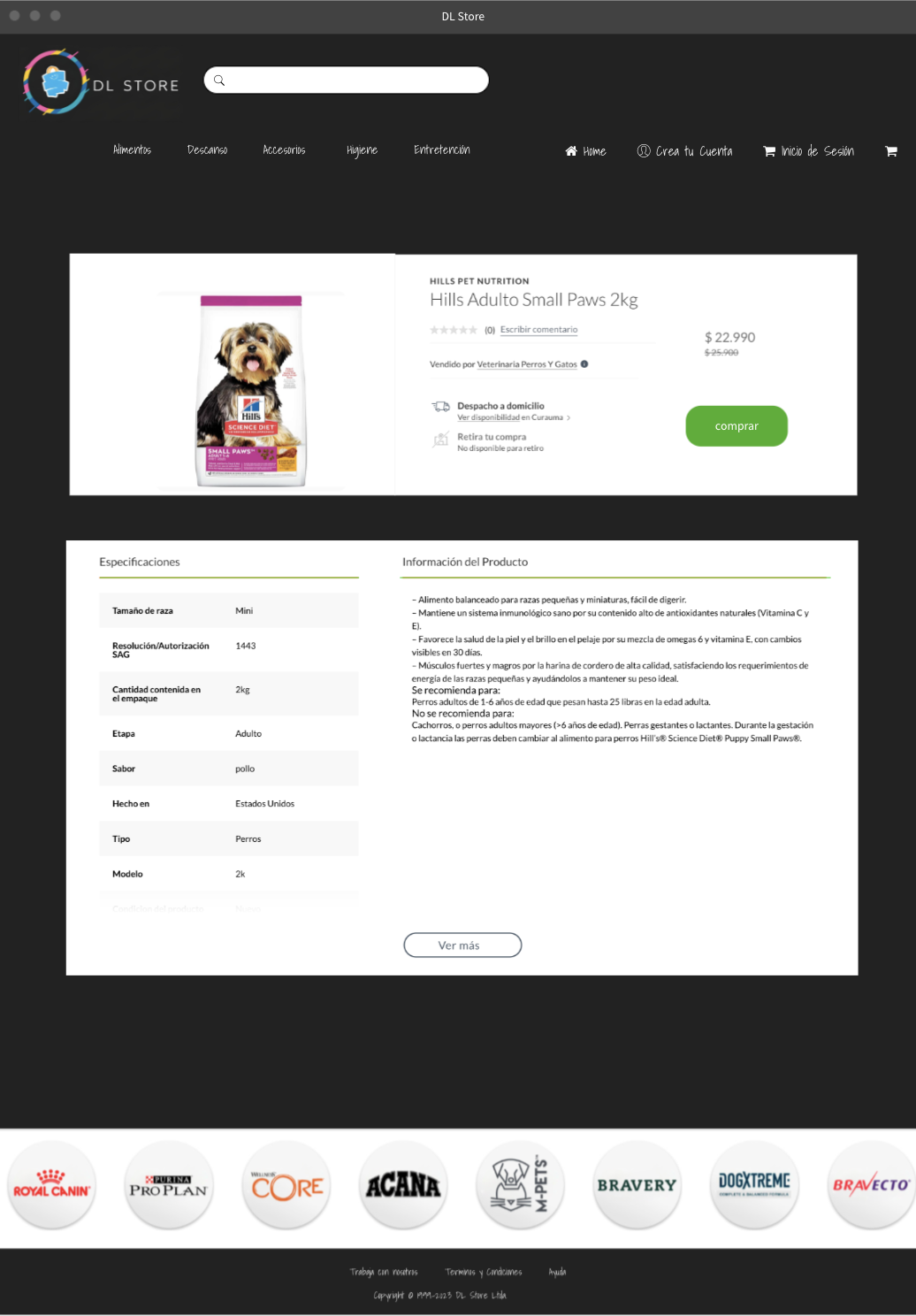
Formulario para crear publicación



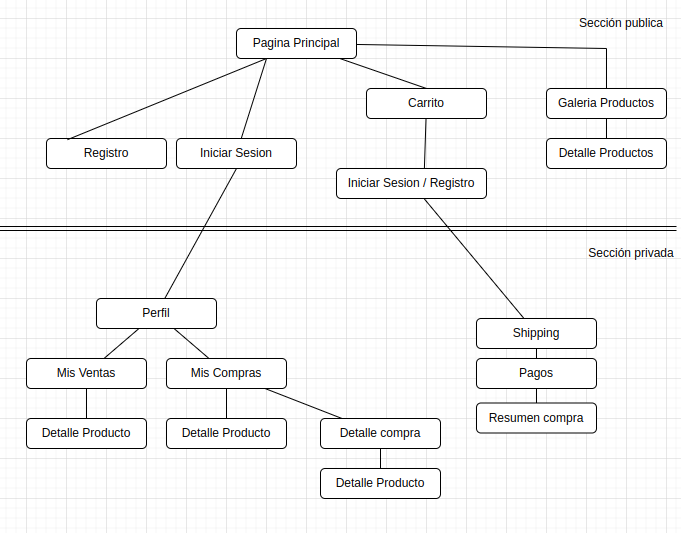
* Galería de Publicaciones



* Vista de detalle de una publicación



2.- Diagrama de Navegación



3.- Listado de dependencias

Dependencias de Frontend

* React-bootstrap
* React-dom
* Gh-pages
* React-router-dom
* React-scripts
* Web-vitals
* axios

Dependencias de BBF

## react-google-login

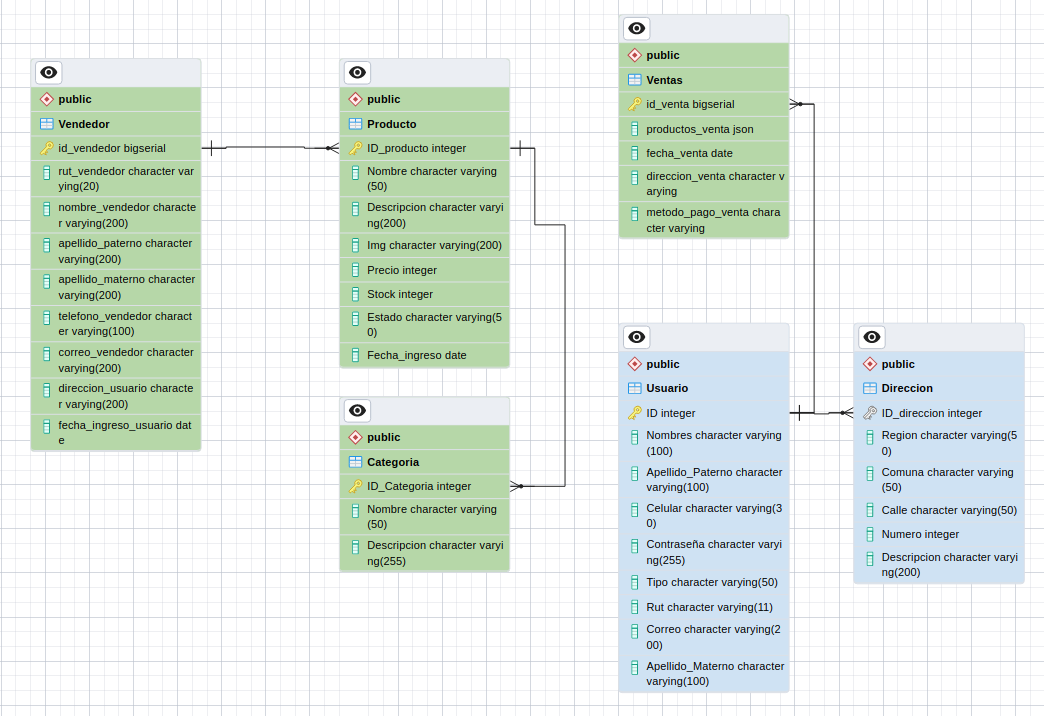
## react-facebook-login

## 

Dependencias de Backend

* Cors
* Express
* Pg
* Nodemon (Desarrollo)
* helmet
* Axios
* Jsonwebtoken
* bcryptjs

4.- Diagrama de Base de datos



5.-Diseño contrato de datos API REST.

openapi: 3.0.0

info:

title: API´s Market-DL

version: "1.0"

description: "API´s para el proyecto final de creación de marketplace Desafio Latam"

servers:

- url: http://localhost:3001

paths:

/login:

post:

summary: Login de Usuarios

tags:

- Login Usuarios

description: Login de usuarios registrados en el sistema

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/LoginRequest'

responses:

'200':

description: OK

content:

application/json:

schema:

$ref: '#/components/schemas/LoginResponse'

'401':

description: Not Authorized

'500':

description: Internal Error

/loginSeller:

post:

summary: Login de Vendedores

tags:

- Login Vendedor

description: Login de vendedores registrados en el sistema

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/LoginSellerRequest'

responses:

'200':

description: OK

content:

application/json:

schema:

$ref: '#/components/schemas/LoginSellerResponse'

'401':

description: Not Authorized

'500':

description: Internal Error

/usuarios:

get:

summary: Obtiene la lista de usuarios

tags:

- Usuario

description: Retorna una lista de todos los usuarios registrados en el sistema

responses:

'200':

description: OK

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/UsuarioResponse'

post:

summary: Crea un nuevo usuario

tags:

- Usuario

description: Crea un nuevo usuario en el sistema

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/UsuarioRequest'

responses:

'200':

description: OK

content:

application/json:

schema:

$ref: '#/components/schemas/UsuarioResponse'

'404':

description: Not Found

/usuarios/{id\_usuario}:

get:

summary: Obtiene un usuario por ID

tags:

- Usuario

description: Retorna un usuario específico basado en su ID

parameters:

- name: id\_usuario

in: path

description: ID del usuario a obtener

required: true

schema:

type: integer

format: int64

responses:

'200':

description: OK

content:

application/json:

schema:

$ref: '#/components/schemas/UsuarioResponse'

'404':

description: Not Found

put:

summary: Actualiza un usuario existente

tags:

- Usuario

description: Actualiza los datos de un usuario existente basado en su ID

parameters:

- name: id\_usuario

in: path

description: ID del usuario a actualizar

required: true

schema:

type: integer

format: int64

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/UsuarioRequest'

responses:

'200':

description: OK

content:

application/json:

schema:

$ref: '#/components/schemas/UsuarioResponse'

'404':

description: Not Found

delete:

summary: Elimina un usuario existente

tags:

- Usuario

description: Elimina un usuario existente basado en su ID

parameters:

- name: id\_usuario

in: path

description: ID del usuario a eliminar

required: true

schema:

type: integer

format: int64

responses:

'204':

description: No Content

'404':

description: Not Found

/vendedores:

get:

summary: Obtiene la lista de todos los vendedores

tags:

- Vendedores

description: Obtiene la lista de todos los vendedores registrados en el sistema

responses:

'200':

description: Lista de vendedores

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/Vendedor'

post:

summary: Crea un nuevo vendedor

tags:

- Vendedores

description: Crea un nuevo vendedor en el sistema

requestBody:

description: Vendedor a crear

content:

application/json:

schema:

$ref: '#/components/schemas/Vendedor'

required: true

responses:

'201':

description: Vendedor creado exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Vendedor'

'400':

description: Error de validación

'500':

description: Error interno del servidor

/vendedores/{id}:

get:

summary: Obtiene un vendedor por su ID

tags:

- Vendedores

description: Obtiene un vendedor en el sistema existente basado en su ID

parameters:

- name: id

in: path

description: ID del vendedor a obtener

required: true

schema:

type: integer

format: int64

responses:

'200':

description: Vendedor obtenido exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Vendedor'

'404':

description: El vendedor solicitado no existe

'500':

description: Error interno del servidor

put:

summary: Actualiza los datos de un vendedor existente

tags:

- Vendedores

description: Actualiza un vendedor en el sistema existente basado en su ID

parameters:

- name: id

in: path

description: ID del vendedor a actualizar

required: true

schema:

type: integer

format: int64

requestBody:

description: Datos actualizados del vendedor

content:

application/json:

schema:

$ref: '#/components/schemas/Vendedor'

required: true

responses:

'200':

description: Vendedor actualizado exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Vendedor'

'400':

description: Error de validación

'404':

description: El vendedor solicitado no existe

'500':

description: Error interno del servidor

delete:

summary: Elimina un vendedor existente

tags:

- Vendedores

description: Elimina un vendedor en el sistema existente basado en su ID

parameters:

- name: id

in: path

description: ID del vendedor a eliminar

required: true

schema:

type: integer

format: int64

responses:

'204':

description: Vendedor eliminado exitosamente

'404':

description: El vendedor solicitado no existe

'500':

description: Error interno del servidor

/compras:

post:

summary: Crea una nueva compra

tags:

- Compras

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Compras'

responses:

'201':

description: Compra creada exitosamente

'400':

description: Solicitud inválida

get:

summary: Obtiene todas las compras realizadas

tags:

- Compras

responses:

'200':

description: Lista de todas las compras realizadas

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/Compras'

'404':

description: No se encontraron compras registradas

/compras/{id}:

get:

summary: Obtiene una compra específica por su ID

tags:

- Compras

parameters:

- name: id

in: path

description: ID de la compra a buscar

required: true

schema:

type: integer

format: int64

responses:

'200':

description: Compra encontrada

content:

application/json:

schema:

$ref: '#/components/schemas/Compras'

'404':

description: No se encontró la compra solicitada

put:

summary: Actualiza una compra existente por su ID

tags:

- Compras

parameters:

- name: id

in: path

description: ID de la compra a actualizar

required: true

schema:

type: integer

format: int64

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Compras'

responses:

'200':

description: Compra actualizada exitosamente

'400':

description: Solicitud inválida

'404':

description: No se encontró la compra solicitada

delete:

summary: Elimina una compra existente por su ID

tags:

- Compras

parameters:

- name: id

in: path

description: ID de la compra a eliminar

required: true

schema:

type: integer

format: int64

responses:

'200':

description: Compra eliminada exitosamente

'404':

description: No se encontró la compra solicitada

/categorias:

get:

summary: Obtener todas las categorías

tags:

- Castegorias

responses:

'200':

description: OK

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/categorias'

post:

summary: Crear una nueva categoría

tags:

- Castegorias

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/categorias'

responses:

'201':

description: Created

/categorias/{id\_categoria}:

get:

summary: Obtener una categoría por su ID

tags:

- Castegorias

parameters:

- in: path

name: id\_categoria

schema:

type: integer

format: int64

required: true

description: ID de la categoría

responses:

'200':

description: OK

'404':

description: Categoría no encontrada

put:

summary: Actualizar una categoría existente

tags:

- Castegorias

parameters:

- in: path

name: id\_categoria

schema:

type: integer

format: int64

required: true

description: ID de la categoría

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/categorias'

responses:

'200':

description: OK

'404':

description: Categoría no encontrada

delete:

summary: Eliminar una categoría existente

tags:

- Castegorias

parameters:

- in: path

name: id\_categoria

schema:

type: integer

format: int64

required: true

description: ID de la categoría

responses:

'204':

description: No content

'404':

description: Categoría no encontrada

/productos:

get:

summary: Obtener todos los productos

tags:

- Productos

description: Obtiene la lista de todos los productos.

responses:

'200':

description: Lista de productos

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/Productos'

post:

summary: Añadir un nuevo producto

tags:

- Productos

description: Añade un nuevo producto al catálogo.

requestBody:

description: Producto a añadir.

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Productos'

responses:

'201':

description: Producto añadido exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Productos'

'400':

description: Solicitud incorrecta

'401':

description: No autorizado

'500':

description: Error del servidor

/productos/{id}:

get:

summary: Obtener un producto por ID

tags:

- Productos

description: Obtiene un producto según su ID.

parameters:

- in: path

name: id

description: ID del producto a obtener.

required: true

schema:

type: integer

format: int64

responses:

'200':

description: Producto obtenido exitosamente.

content:

application/json:

schema:

$ref: '#/components/schemas/Productos'

'404':

description: Producto no encontrado.

put:

summary: Actualizar un producto existente

tags:

- Productos

description: Actualiza un producto según su ID.

parameters:

- in: path

name: id

description: ID del producto a actualizar.

required: true

schema:

type: integer

format: int64

requestBody:

description: Producto actualizado.

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Productos'

responses:

'200':

description: Producto actualizado exitosamente.

content:

application/json:

schema:

$ref: '#/components/schemas/Productos'

'404':

description: Producto no encontrado.

delete:

summary: Eliminar un producto existente

tags:

- Productos

description: Elimina un producto según su ID.

parameters:

- in: path

name: id

description: ID del producto a eliminar.

required: true

schema:

type: integer

format: int64

responses:

'204':

description: Producto eliminado exitosamente.

'404':

description: Producto no encontrado.

/direccion:

get:

summary: Obtiene todas las direcciones

tags:

- Direccion

responses:

'200':

description: Direcciones obtenidas

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/Direccion'

'404':

description: No se encontraron direcciones

post:

summary: Crea una nueva dirección

tags:

- Direccion

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Direccion'

responses:

'201':

description: Dirección creada exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Direccion'

'400':

description: Datos inválidos para crear una dirección

/direccion/{id}:

parameters:

- name: id

in: path

description: ID de la dirección

required: true

schema:

type: integer

format: int64

get:

summary: Obtiene una dirección por su ID

tags:

- Direccion

responses:

'200':

description: Dirección obtenida

content:

application/json:

schema:

$ref: '#/components/schemas/Direccion'

'404':

description: No se encontró la dirección

put:

summary: Actualiza una dirección existente por su ID

tags:

- Direccion

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/Direccion'

responses:

'200':

description: Dirección actualizada exitosamente

content:

application/json:

schema:

$ref: '#/components/schemas/Direccion'

'400':

description: Datos inválidos para actualizar una dirección

'404':

description: No se encontró la dirección

delete:

summary: Elimina una dirección por su ID

tags:

- Direccion

responses:

'200':

description: Dirección eliminada exitosamente

'404':

description: No se encontró la dirección

components:

schemas:

UsuarioRequest:

type: object

properties:

nombres:

type: string

description: Nombres del usuario

example: 'Felipe'

apellido\_paterno:

type: string

description: Apellido paterno del usuario

example: 'Cruz'

apellido\_materno:

type: string

description: Apellido materno del usuario

example: 'Bonilla'

celular:

type: string

description: Número de celular del usuario

example: '+56987654321'

contrasena:

type: string

description: Contraseña del usuario

example: 'PasswordExample123'

rut:

type: string

description: RUT del usuario

example: '98765432-1'

correo:

type: string

format: email

description: Correo electrónico del usuario

example: 'felipe@dl-store.cl'

sexo:

type: string

enum: [Masculino, Femenino, Otro]

description: Género del usuario

fecha\_nacimiento:

type: string

format: date

description: Fecha de nacimiento del usuario

example: '1995-03-31'

region:

type: string

description: Ciudad del usuario

example: 'Valparaíso'

ciudad:

type: string

description: Región del usuario

example: 'Valparaíso'

comuna:

type: string

description: Comuna del usuario

example: 'Valparaíso'

numero:

type: integer

format: int64

description: Número de dirección del usuario

example: 50

required:

- nombres

- apellido\_paterno

- apellido\_materno

- celular

- contrasena

- rut

- correo

- sexo

- fecha\_nacimiento

- region

- ciudad

- comuna

- numero

UsuarioResponse:

type: object

properties:

id\_usuario:

type: integer

format: int64

description: Identificador único del usuario

example: 1

nombres:

type: string

description: Nombres del usuario

example: 'Felipe'

apellido\_paterno:

type: string

description: Apellido paterno del usuario

example: 'Cruz'

apellido\_materno:

type: string

description: Apellido materno del usuario

example: 'Bonilla'

celular:

type: string

description: Número de celular del usuario

example: '+56987654321'

contrasena:

type: string

description: Contraseña del usuario

example: 'PasswordExample123'

rut:

type: string

description: RUT del usuario

example: '98765432-1'

correo:

type: string

format: email

description: Correo electrónico del usuario

example: 'felipe@dl-store.cl'

sexo:

type: string

enum: [Masculino, Femenino, Otro]

description: Género del usuario

fecha\_nacimiento:

type: string

format: date

description: Fecha de nacimiento del usuario

example: '1995-03-31'

required:

- nombres

- apellido\_paterno

- apellido\_materno

- celular

- contrasena

- rut

- correo

- sexo

- fecha\_nacimiento

Vendedor:

type: object

required:

- id\_vendedor

- rut

- nombre

- apellido\_paterno

- apellido\_materno

- telefono

- correo

- region

- ciudad

- comuna

- calle

- numero

- fecha\_ingreso

- contrasena

properties:

id\_vendedor:

type: integer

format: int64

rut:

type: string

nombre:

type: string

apellido\_paterno:

type: string

apellido\_materno:

type: string

telefono:

type: string

correo:

type: string

region:

type: string

ciudad:

type: string

comuna:

type: string

calle:

type: string

numero:

type: integer

format: int32

fecha\_ingreso:

type: string

format: date

contrasena:

type: string

Compras:

type: object

properties:

id\_compra:

type: integer

format: int64

productos:

type: string

fecha:

type: string

format: date

direccion:

type: string

id\_usuario:

type: integer

format: int64

required:

- productos

- fecha

- direccion

- id\_usuario

categorias:

type: object

properties:

id\_categoria:

type: integer

format: int64

nombre:

type: string

descripcion:

type: string

required:

- nombre

- descripcion

Productos:

type: object

properties:

id\_producto:

type: integer

format: int64

nombre:

type: string

descripcion:

type: string

img:

type: string

precio:

type: integer

format: int32

stock:

type: integer

format: int32

estado:

type: string

fecha\_ingreso:

type: string

format: date

marca:

type: string

id\_vendedor:

type: integer

format: int64

id\_categoria:

type: integer

format: int64

required:

- nombre

- descripcion

- img

- precio

- stock

- estado

- fecha\_ingreso

- marca

- id\_vendedor

- id\_categoria

Direccion:

type: object

properties:

id\_direccion:

type: integer

format: int64

description: ID de la dirección

region:

type: string

description: Región de la dirección

ciudad:

type: string

description: Ciudad de la dirección

comuna:

type: string

description: Comuna de la dirección

calle:

type: string

description: Calle de la dirección

numero:

type: string

description: Número de la dirección

descripcion:

type: string

description: Descripción de la dirección

id\_usuario:

type: integer

format: int64

description: ID del usuario asociado a la dirección

LoginRequest:

type : object

properties:

correo:

type: string

format: email

description: Correo electrónico del usuario

example: 'felipe@dl-store.cl'

contrasena:

type: string

description: Contraseña del usuario

example: 'PasswordExample123'

LoginResponse:

type: object

properties:

statusCode:

type: integer

format: int64

description: Status Code de respuesta

message:

type: string

description: Mensaje de respuesta

jwt\_token:

type: string

description: Token JWT

LoginSellerRequest:

type : object

properties:

correo:

type: string

format: email

description: Correo electrónico del usuario

example: 'felipe@dl-store.cl'

contrasena:

type: string

description: Contraseña del usuario

example: 'PasswordExample123'

LoginSellerResponse:

type: object

properties:

statusCode:

type: integer

format: int64

description: Status Code de respuesta

message:

type: string

description: Mensaje de respuesta

jwt\_token:

type: string

description: Token JWT