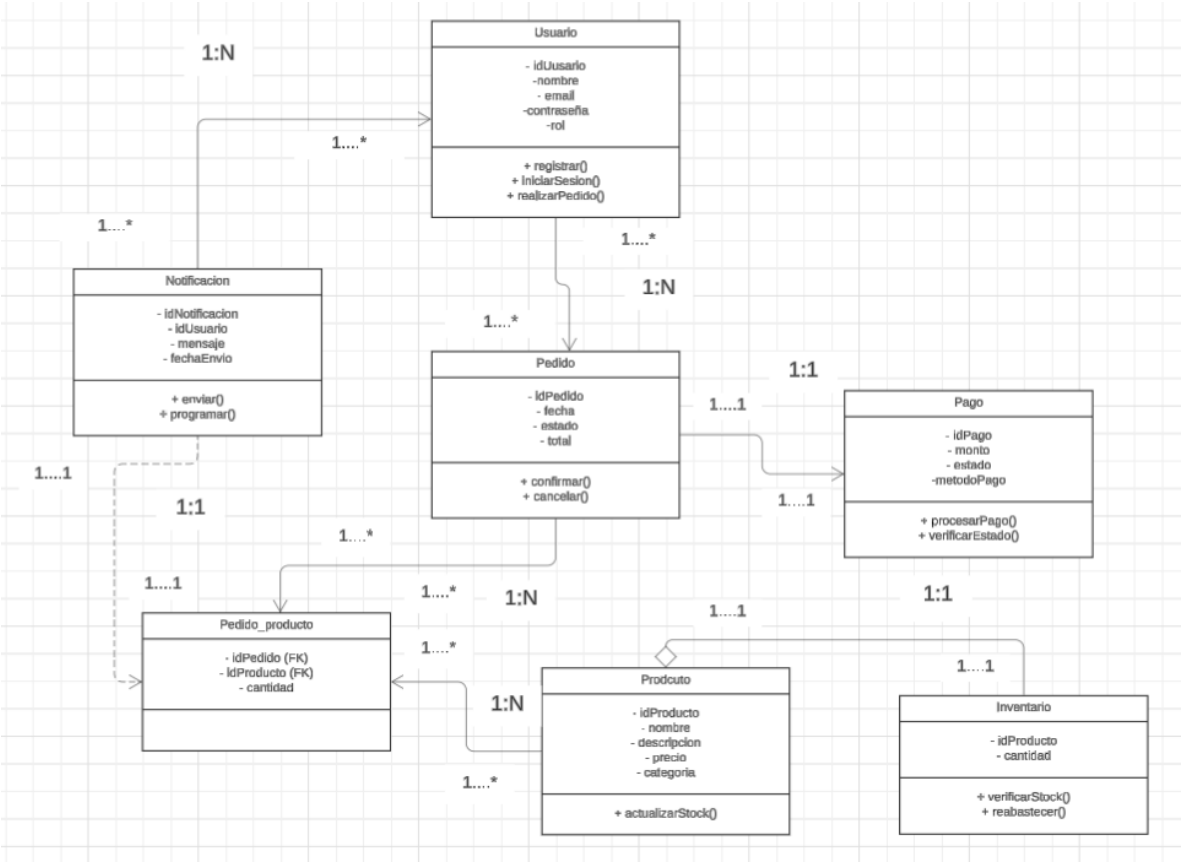
DIAGRAMA UML DE CLASES

Damos a conocer nuestras clases con sus atributos y métodos correspondientes.



FICHAS CRC

Nuestras fichas CRC cumplen el propósito de dar a conocer las responsabilidades y colaboradores de nuestras clases, cada uno con su respectivo dato.

Usuario	
Responsabilidades • Registrar nuevos usuarios • Iniciar sesión en el sistema • Realizar	Colaboradores • Pedido (para realizar los pedidos) • Notificacion (Para recibir notificacione s)

Pedido	
Responsabilidades • Confirmar pedido. • Cancelar pedido. • Calcular el total del	Colaboradores • Usuario (Porque cada pedido pertenece a un usuario). • Pedido_Producto (Para gestionar los productos dentro del pedido).

Producto	
Responsabilidades • Gestionar información del producto. • Actualizar Stock.	Colaboradores • Pedido_producto (Porque un producto puede estar en varios pedidos). • Inventario (Para verificar y actualizar el

Pedido_producto	
Responsabilidades • Registrar la cantidad de productos en un pedido. • Permitir agregar o eliminar	Colaboradores • Pedido (Porque es parte de un pedido). • Producto (porque contiene productos en un pedido).

Inventario	
Responsabilidades • Verificar la cantidad de stock de un producto. • Reabastecer el stock cuando sea necesario	Colaboradores • Producto (Porque cada producto tiene una cantidad en

Pago	
Responsabilidades • Procesar el pago de un pedido. • Verificar el estado del pago.	Colaboradores • Pedido (Porque cada pago esta asociado a un pedido).

Notificacion	
Responsabilidades • Enviar notificaciones al usuario. • Programar notificaciones .	Colaboradores • Usuario (Porque cada notificación esta dirigida a un

DIAGRAMA DE SECUENCIA

CASO DE USO: REALIZACION DE UN PEDIDO

El diagrama de secuencia tiene como propósito plasmar gráficamente la función de la realización de uno de nuestros productos, desde que el usuario inicia sesión, selecciona sus productos, hasta que se le envía la notificación de su compra, pasando por los procesos que existen antes de recibir el producto.

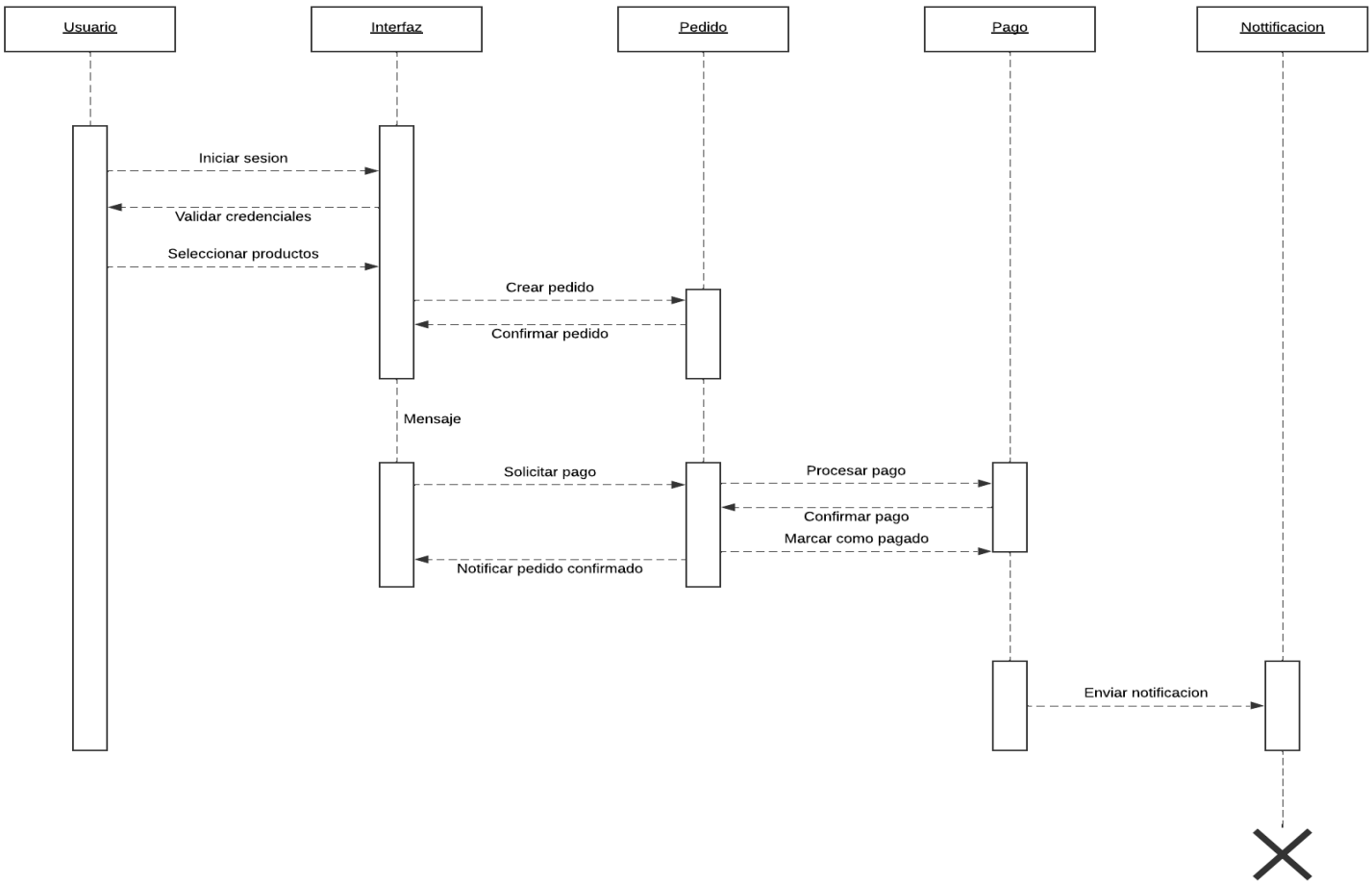


DIAGRAMA DE INTERACCION

CASO DE USO: PAGO DE UN PEDIDO

El siguiente diagrama tiene como propósito dar a entender mediante un UML de interacción la función del pago de un pedido mediante la forma en las que el usuario puede ingresar desde la interfaz, y en la misma realizar el pedido y el pago.

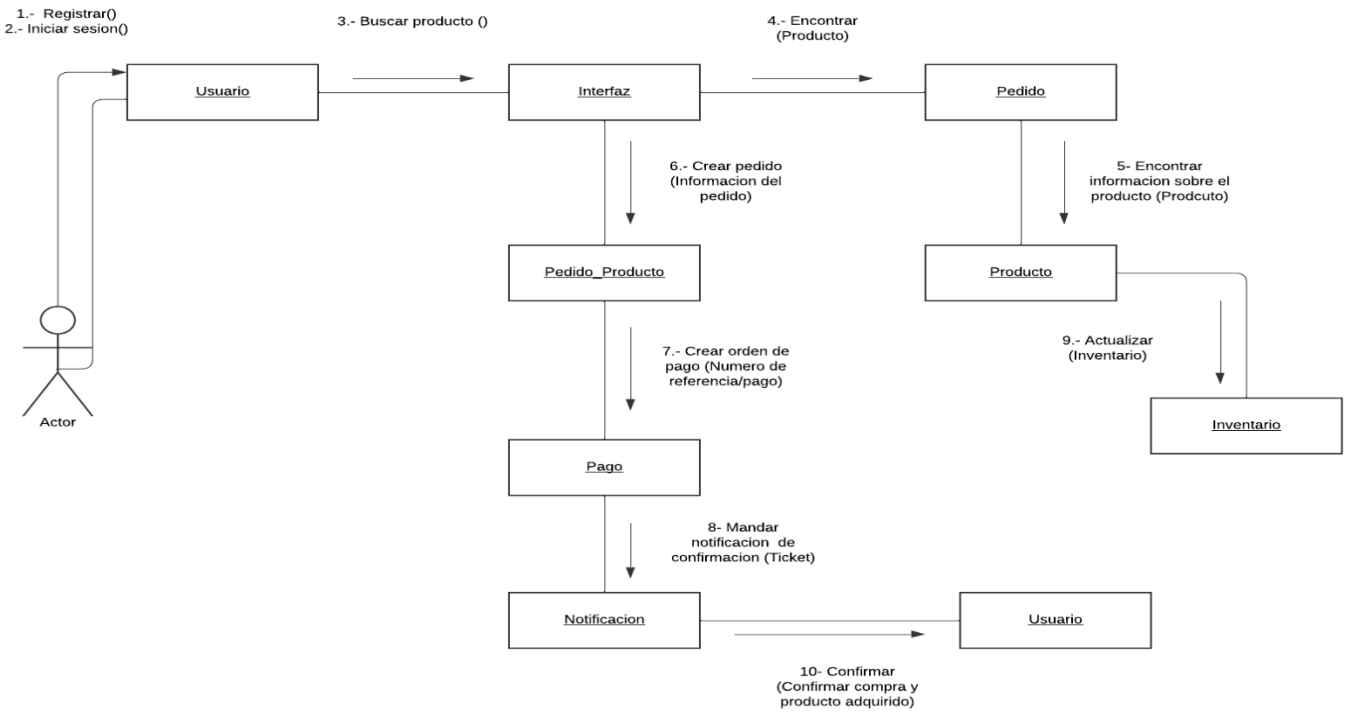
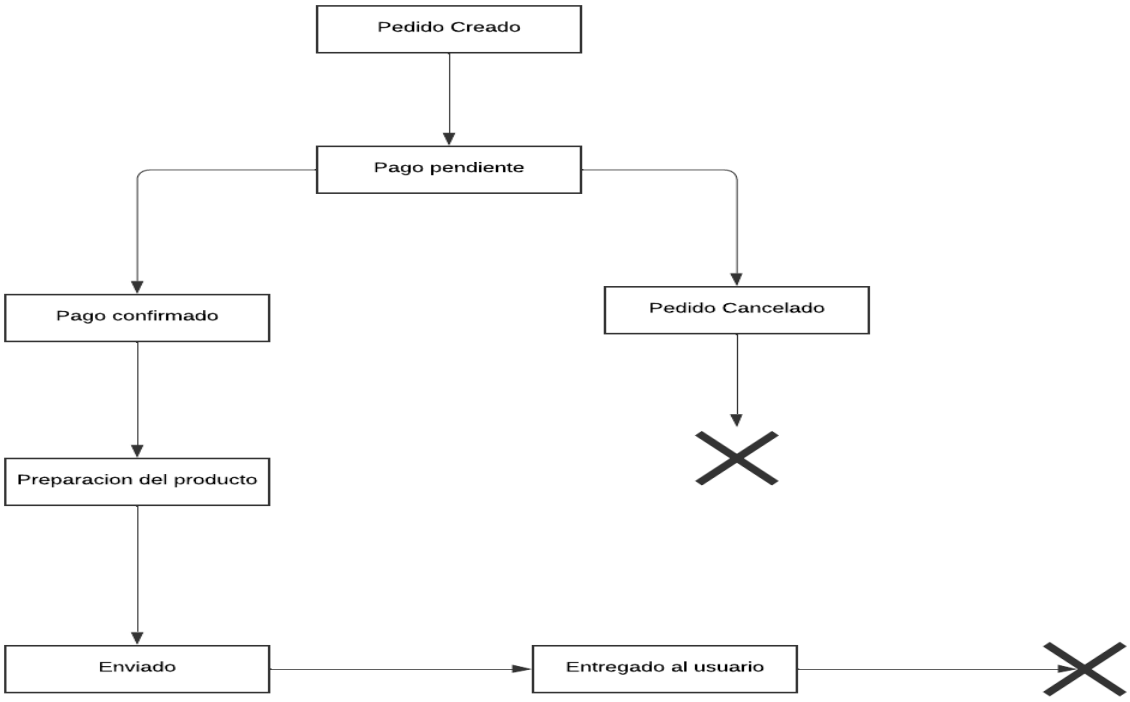


DIAGRAMA DE COMPORTAMIENTO

CASO DE USO: PASOS POR LOS QUE PASA UN PRODUCTO

El diagrama UML de comportamiento muestra la forma en la que se lleva a cabo los pagos de nuestros productos desde que se crea el pedido, hasta que el usuario recibe su producto.



BASE DE DATOS

La creación de nuestra base de datos para el proyecto de una pizzería cuenta con una estructura relación mostrada a continuación:

- Usuarios (id_usuario, nombre, correo, contraseña, direccion, telefono)
- Productos (id_producto, nombre, descripcion, precio, stock)
- Pedidos (id_pedido, id_usuario, fecha, estado, total)

- Pedido_Producto (id_pedido_producto, id_pedido, id_producto, cantidad, subtotal)

- Pagos (id_pago, id_pedido, metodo_pago, estado_pago, fecha_pago, total_pagado)

- Notificaciones (id_notificacion, id_usuario, mensaje, fecha)

Por lo que la estructura de nuestra base de datos es la siguiente, esta fue creada en PostgreSQL.

```
CREATE TABLE Usuarios (
    id_usuario SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100) UNIQUE NOT NULL,
    contraseña TEXT NOT NULL,
    direccion TEXT,
    telefono VARCHAR(20)
);



CREATE TABLE Productos (
    id_producto SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    precio NUMERIC(10,2) NOT NULL,
    stock INT NOT NULL
);



CREATE TABLE Pedidos (
```

```
    id_pedido SERIAL PRIMARY KEY,
    id_usuario INT NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado VARCHAR(20) CHECK (estado IN ('Creado', 'Pendiente de Pago', 'Pagado', 'En Preparación', 'Enviado', 'Entregado', 'Cancelado')),
    total NUMERIC(10,2) NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
);


CREATE TABLE Pedido_Producto (
    id_pedido_producto SERIAL PRIMARY KEY,
    id_pedido INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    subtotal NUMERIC(10,2) NOT NULL,
    FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido),
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)
);


CREATE TABLE Pagos (
    id_pago SERIAL PRIMARY KEY,
    id_pedido INT NOT NULL,
    metodo_pago VARCHAR(20) CHECK (metodo_pago IN ('Tarjeta', 'PayPal', 'Efectivo')),
    estado_pago VARCHAR(20) CHECK (estado_pago IN ('Pendiente', 'Aprobado', 'Rechazado')),
    fecha_pago TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total_pagado NUMERIC(10,2) NOT NULL,
    FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido)
```

```
);


CREATE TABLE Notificaciones (
    id_notificacion SERIAL PRIMARY KEY,
    id_usuario INT NOT NULL,
    mensaje TEXT NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)
);
```