## **1) Resumen del Dominio Principal**

**Dominio Principal:** Gestión de Inventario y Órdenes.

Motivación: permitir registrar productos, gestionar stock, crear y modificar órdenes, generar reportes y manejar carritos/notificaficaciones.

# **2) Event Storming**

## **Eventos de dominio (Domain Events)**

* ProductoRegistrado
* StockActualizado
* StockBajoDetectado
* OrdenCreada
* OrdenModificada
* EstadoPedidoCambiado (pendiente → confirmado → en\_despacho → entregado)
* CarritoAbandonado
* ReporteSemanalGenerado
* NotificacionEnviada

# **3) Modelo de Dominio (entidades y relaciones conceptuales)**

Entidades principales y atributos:

* **Producto**
* **Abastecimiento**
* **Cliente**
* **Pedido**
* **PeidoItem**
* **Carrito**
* **CarritoItem**
* **Reporte**

# **4) Bounded Contexts y Subdominios**

## **Bounded Contexts**

1. **Inventario**  — gestión de productos, stock y historial.
2. **Ventas / Pedidos**  — creación/modificación de órdenes, carritos, estados de pedido.
3. **Clientes**  — información y autenticación básica.
4. **Reportes** — generación y exportación de reportes.
5. **Notificaciones** — correo, notificaciones en plataforma.
6. **Gateway / API**  — orkestración y contratos HTTP.
7. **Autenticación / Seguridad**  — auth, roles.

## **Subdominios**

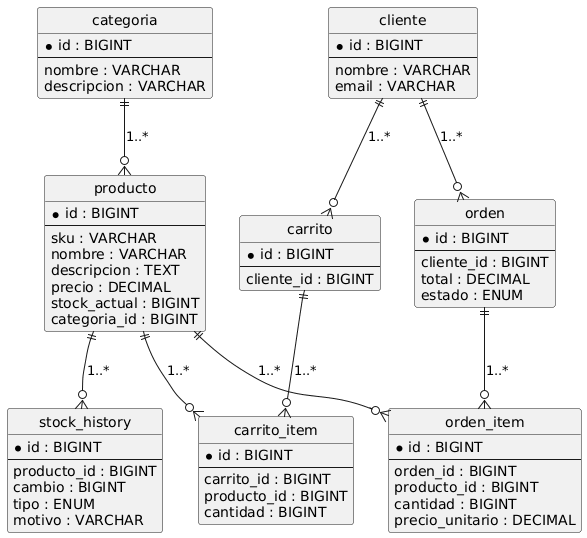
* **Prinicipales:** Inventario, Pedidos,Clientes.
* **Soporte:** Reportes, Notificaciones.
* **Genericos:** Autenticacion.

# **5) Lenguaje ubicuo (Ubiquitous Language) — términos y definiciones rápidas**

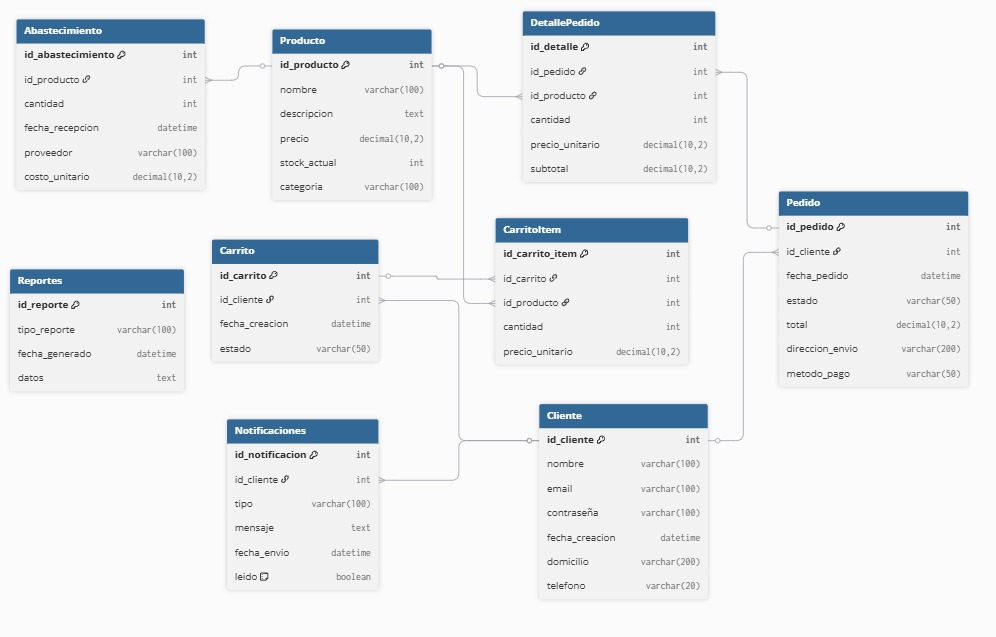
* **Producto:** artículo que se vende; tiene precio, stock y categoría.
* **Stock Disponible:** cantidad sin reservas.
* **Orden:** petición de compra en un estado (pendiente, confirmado, en\_despacho, entregado).
* **Carrito:** selección temporal de items por un cliente.
* **Reporte Semanal:** archivo que agrupa métricas de ventas por semana.
* **Historial de stock:** registro de todos los movimientos de inventario.

# **6) ER (lógico), Diagrama BD y Diagrama de Clases**

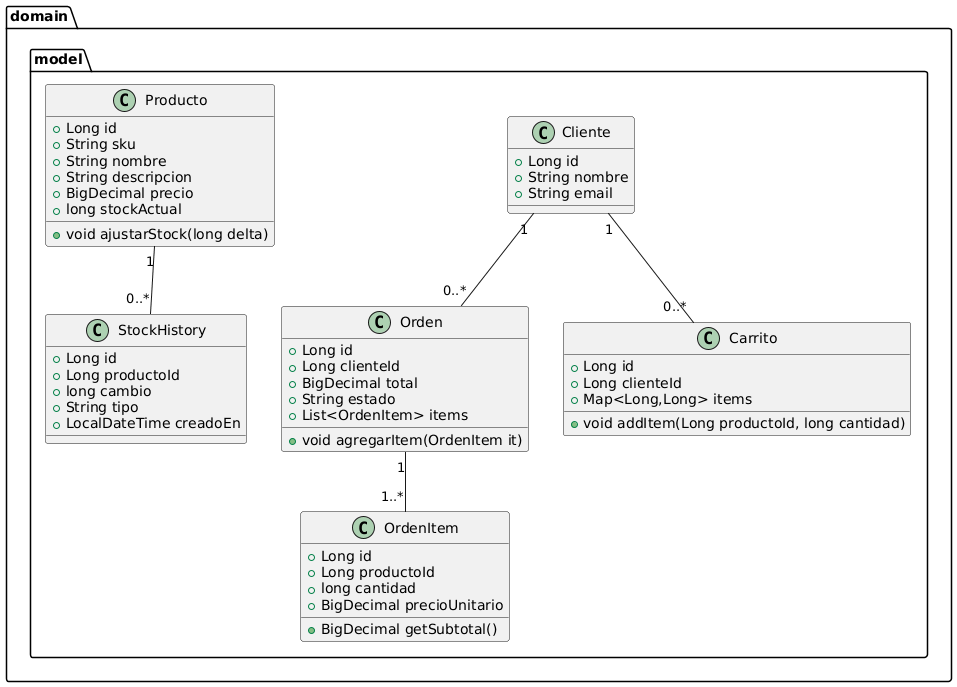
## **Diagrama Entidad-Relacion**



## **Diagrama de Base de Datos.**



## **Diagrama de Clases - dominio**



# **7) Endpoints principales**

### **Inventario (Inventory API)**

* POST /api/v1/productos — crear producto  
   RequestBody: ProductoCreateDto → Response Mono<ProductoDto>
* GET /api/v1/productos — listar productos → Flux<ProductoDto>
* GET /api/v1/productos/{id} — obtener producto → Mono<ProductoDto>
* PATCH /api/v1/productos/{id}/stock — actualizar stock → Mono<Void> (body { "delta": 10, "motivo":"compra\_proveedor" })
* GET /api/v1/reportes/abastecer — lista productos por debajo de umbral → Flux<ProductoDto>

### **Pedidos (Ordering API)**

* POST /api/v1/ordenes — crear orden → Mono<OrdenDto>
* PUT /api/v1/ordenes/{id} — modificar orden (solo si pendiente) → Mono<OrdenDto>
* GET /api/v1/ordenes/{id} — obtener orden → Mono<OrdenDto>
* GET /api/v1/ordenes?cliente={id} — listar por cliente → Flux<OrdenDto>
* POST /api/v1/ordenes/{id}/estado — cambiar estado → Mono<OrdenDto>

### **Carrito**

* POST /api/v1/carritos/{clienteId}/items — agregar item
* GET /api/v1/carritos/{clienteId} — ver carrito

### **Reportes**

* GET /api/v1/reportes/ventas/semanal — generar/descargar reporte semanal (Mono<Resource>)

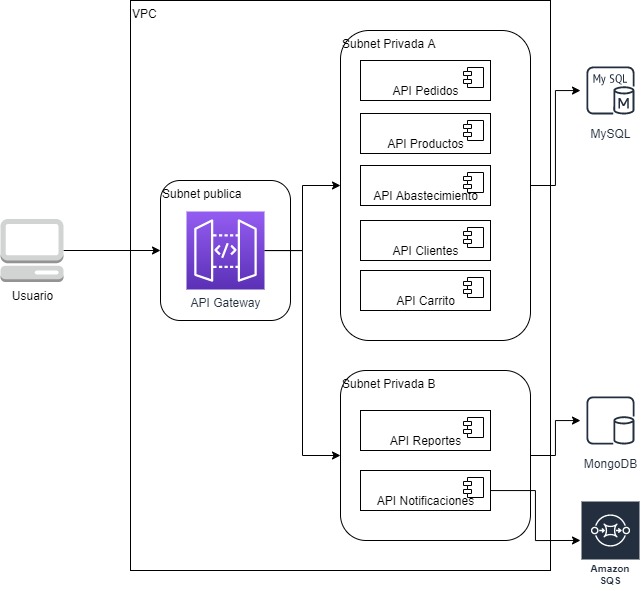
### **Notificaciones**

* POST /api/v1/notificaciones/email — enviar notificación (interno)

# **8) Integridad y transacciones (estrategia)**

* Operaciones de stock y creación de orden: usar transacción.
* Validaciones: precio >= 0, cantidad > 0 en dominio; no permitir stock negativo.
* Reservas en creación de orden: decrementar stock\_actual y guardar stock\_en la misma transacción.

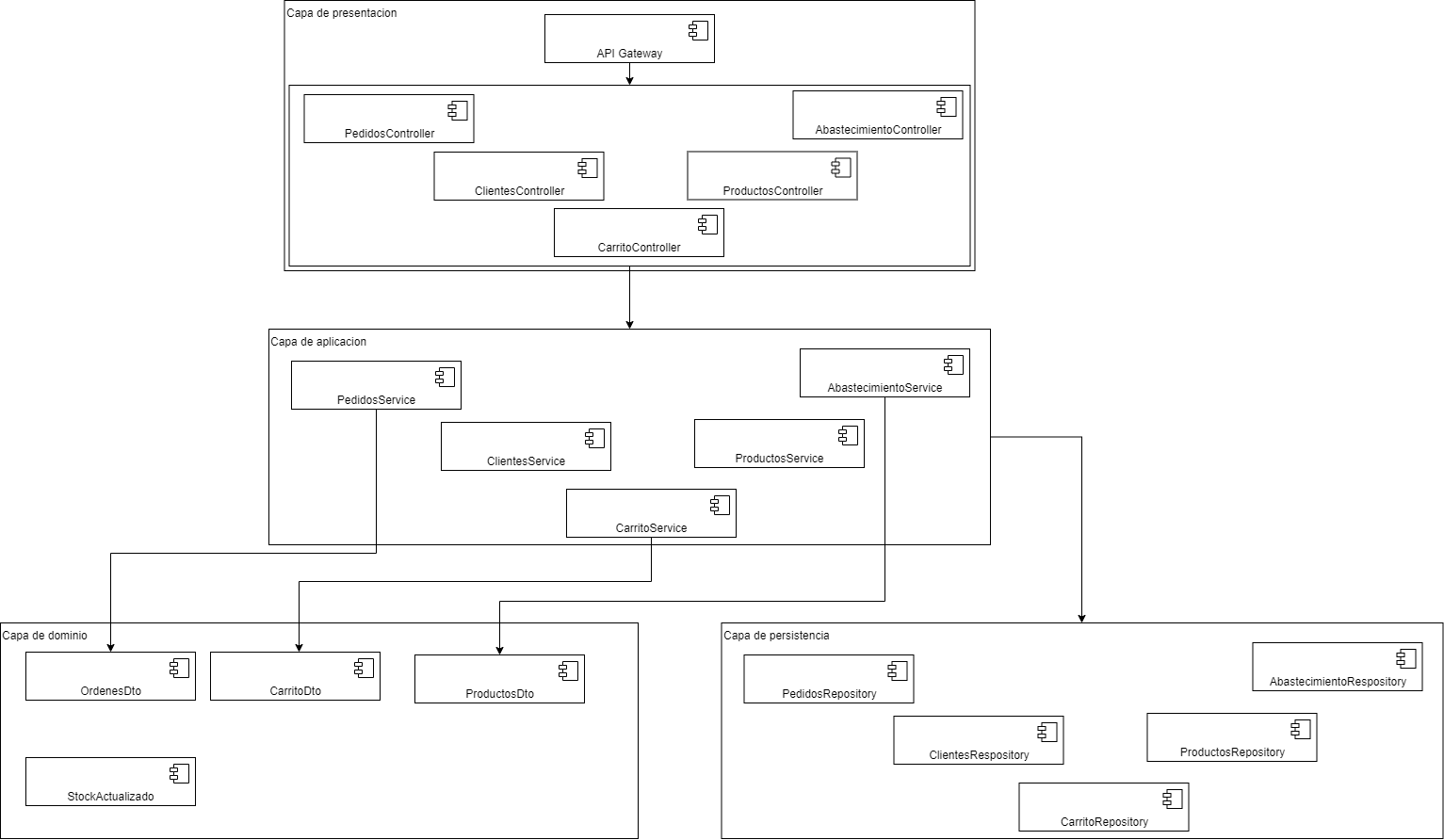
# **9) Diagrama de la infraestructura**



**Descripción rápida**:

* Gateway para expone apis mediante ECS.
* MySQL: Abastecimiento, Pedidos, Clientes, Carrito, Productos.
* MongoDB/DocumentDB: Reportes, Notificaciones.
* SQS/SNS: mensajería asincrónica.

# **10) Diagrama de arquitectura**



* Presentación: controladores WebFlux exponen endpoints REST reactivos.
* Aplicación: contiene orquestación de casos de uso (servicios).
* Dominio: entidades y eventos de dominio.
* Persistencia (MySQL y MongoDB).