

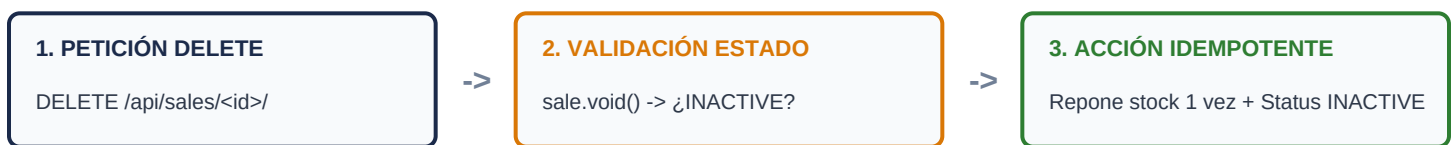
ISS-12: CONSULTA Y ANULACIÓN DE VENTAS

Arquitectura de Lectura, Preservación de Precio Histórico y Anulación Idempotente

1. RESUMEN DE ARQUITECTURA DE VISTAS

En el ISS-12 se completa el ciclo de vida de la venta habilitando la lectura (listado y detalle) y la anulación blanda. Se implementa **SaleReadViewSet** mezclando de forma precisa ListModelMixin, RetrieveModelMixin, DestroyModelMixin y GenericViewSet. Se omiten intencionalmente las capacidades de actualización (PUT/PATCH) para preservar la inmutabilidad contable del registro transaccional.

2. FLUJO DE ANULACIÓN IDEMPOTENTE (sale.void())



3. MATRIZ DE ENDPOINTS Y MAPPING MUESTRAL

Método HTTP	Ruta API	Componente / Handler	Efecto en Negocio / Base de Datos
POST	/api/sales/	SaleCollectionAPIView (ISS-11)	Registro atómico, descuenta stock, unit_price congelado
GET	/api/sales/	SaleCollectionAPIView -> SaleReadViewSet.list	Lista todas las ventas con client y resumen de totales
GET	/api/sales/<int:pk>/	SaleReadViewSet.as_view({'get':'retrieve'})	Detalle con líneas. Mantiene unit_price histórico
DELETE	/api/sales/<int:pk>/	SaleReadViewSet.as_view({'delete':'destroy'})	Invocación de sale.void(): status=INACTIVE + reabastece stock

4. CRITERIOS DE ACEPTACIÓN Y MATRIZ GATE

Criterio	Requisito de Aceptación	Verificación / Evidencia PASS	Estado
AC-12-01	GET detalle incluye líneas y unit_price congelado	test_detail_keeps_historical_price PASS	PASS
AC-12-02	DELETE responde HTTP 204 No Content	test_void_restores_stock_once PASS	PASS
AC-12-03	La fila en sales permanece conservada en status INACTIVE	EVI-12-01 / Inspección ORM de estado	PASS
AC-12-04	Anular dos veces mantiene el stock en valor original (idempotente)	test_void_restores_stock_once stock==5	PASS