

## Rúbrica de Evaluación: FashionStore (10 Puntos)

### 1. Arquitectura y Stack Tecnológico (2.0 Puntos)

Evalúa el uso correcto de Astro 5.0, la estructura del proyecto y la gestión del estado.

Criterio	Puntuación	Descripción del Requisito
<b>Astro Híbrido (SSG/SSR)</b>	<b>0.75</b>	Implementación correcta de output: 'hybrid'. El catálogo público y fichas de producto deben ser estáticos (SSG) para SEO. El Admin, Carrito y Checkout deben ser dinámicos (SSR).
<b>Islas y Estado (Nano Stores)</b>	<b>0.75</b>	Uso correcto de "Islands Architecture" (hidratación parcial solo donde es necesario). El carrito de compras debe persistir usando <b>Nano Stores</b> , funcionando fluidamente entre navegaciones.
<b>Calidad de Código y TS</b>	<b>0.50</b>	Estructura de carpetas limpia y modular (según lo solicitado). Uso estricto de <b>TypeScript</b> (interfaces para Productos, Pedidos, Usuarios). Sin errores de linter en consola.

### 2. Base de Datos y Lógica Crítica (Supabase) (2.5 Puntos)

Evalúa el diseño del esquema, la seguridad y la integridad transaccional (la parte más difícil).

Criterio	Puntuación	Descripción del Requisito
<b>Esquema y Relaciones</b>	<b>0.50</b>	Tablas normalizadas (products, categories, orders, invoices, order_items). Uso correcto de claves foráneas y tipos de datos (ej: enteros para precios).

<b>Atomicidad y Stock</b>	<b>1.00</b>	<b>CRÍTICO:</b> Implementación de Transacciones o <i>Stored Procedures</i> para el stock. Al cancelar un pedido ("No enviado"), el stock debe volver automáticamente. No se permite "race condition" (vender sin stock).
<b>Seguridad (RLS y Auth)</b>	<b>0.50</b>	Políticas <i>Row Level Security</i> (RLS) activas: El público solo lee productos; solo el Admin edita. El Admin está protegido por Middleware de autenticación.
<b>Storage y Multimedia</b>	<b>0.50</b>	Buckets configurados correctamente. Subida de múltiples imágenes (Drag & Drop) en el admin y optimización de carga en el front.

### 3. Funcionalidad Tienda Pública (UX/UI) (2.0 Puntos)

Evalúa la experiencia de compra, diseño y las "features" extra solicitadas.

Criterio	Puntuación	Descripción del Requisito
<b>Diseño y UX "Premium"</b>	<b>0.50</b>	Estética minimalista con Tailwind CSS. Tipografías y paleta de colores acorde al <i>briefing</i> . Responsividad perfecta (Móvil/Desktop). Slide-over del carrito funcional.
<b>Buscador y Filtros</b>	<b>0.50</b>	Buscador "Live Search" con <i>Debounce</i> (no sobrecarga la API) y filtrado por categorías funcional.

<b>Marketing (Newsletter/Ofertas)</b>	<b>0.50</b>	Pop-up de captación de leads funcional. Sección "Ofertas Flash" controlada por interruptor desde el admin. Aplicación correcta de códigos de descuento en el carrito.
<b>Recomendador de Talla</b>	<b>0.50</b>	Lógica algorítmica implementada en un modal (Isla) que sugiere talla basándose en inputs de altura/peso.

#### 4. Backoffice y Gestión Administrativa (2.5 Puntos)

*Evalúa la capacidad del sistema para gestionar el negocio real (Pedidos, Devoluciones y Facturas).*

Criterio	Puntuación	Descripción del Requisito
<b>Dashboard y KPIs</b>	<b>0.50</b>	Gráficos visuales (Chart.js/Recharts) reales conectados a la BD. Tarjetas de métricas (Ventas totales, Pedidos pendientes) calculadas con SQL (Count/Sum).
<b>Gestión de Pedidos</b>	<b>0.75</b>	Flujo completo de estados: <i>Pendiente -&gt; Pagado -&gt; Enviado -&gt; Entregado</i> . Posibilidad de cancelar antes del envío (disparando la restauración de stock).
<b>Facturación y Abonos</b>	<b>1.25</b>	<b>CRÍTICO:</b> Generación de factura al comprar. Al procesar una <b>devolución</b> , el sistema debe generar una <b>Factura de Abono</b> (rectificativa) con importe negativo para cuadrar caja.

#### 5. Despliegue y Entrega (1.0 Punto)

*Evalúa la puesta en producción y la profesionalidad de la entrega.*

Criterio	Puntuación	Descripción del Requisito
<b>Despliegue VPS (Coolify)</b>	<b>0.50</b>	La aplicación es accesible vía URL pública. Funciona en entorno de producción (Docker/Node). Las variables de entorno están seguras.
<b>Documentación Técnica</b>	<b>0.50</b>	Informe PDF entregado con diagrama ER (Entidad-Relación), justificación de tecnologías y explicación del flujo de facturación.

---

### Penalizaciones y Bonus

- **Penalización Grave (-2.0 Puntos):** Claves de API secretas (Service Role) expuestas en el repositorio público o en el cliente (navegador).
- **Penalización Media (-1.0 Punto):** La aplicación rompe (crash) al intentar comprar sin stock o al subir una imagen muy pesada.
- **Bonus (+1.0 Punto):** Implementación de envío de correos reales (con Resend o similar) al confirmar pedido, enviar factura o confirmar devolución.

---

### Resumen de Calificación

Nivel	Puntuación	Descripción
Junior	<b>0 - 4.9</b>	No compila, faltan funcionalidades críticas (Facturación/Stock) o usa tecnologías prohibidas.
Mid-Level	<b>5.0 - 6.9</b>	Funciona lo básico. El diseño es pobre o falta la lógica compleja (Abonos/Atomicidad).

<b>Avanzado</b>	<b>7.0 - 8.9</b>	Cumple todos los requisitos funcionales. Buena arquitectura. Falla en detalles menores de UX o documentación.
<b>Senior</b>	<b>9.0 - 10</b>	Arquitectura impecable, código limpio, gestión de errores robusta. El sistema de facturación y stock es a prueba de balas. UX excelente.