Business Model Canvas – Plataforma E-commerce Licenciada

| **Bloque** | **Detalle** |
| --- | --- |
| **Propuesta de valor** | Plataforma móvil de e-commerce lista para usar, personalizada con marca y catálogo de cada tienda. Se activa mediante una licencia de uso por tiempo determinado. Permite ventas con pago contra entrega y pasarelas locales, soporta multi-moneda (COP/USD). |
| **Segmentos de clientes** | - Dueños de tiendas físicas que quieren vender online sin invertir en desarrollo.- Pequeñas y medianas empresas (PYMEs) que necesitan presencia móvil rápida y robusta.- Emprendedores que quieren lanzar tienda online en pocos días. |
| **Canales** | - Página web oficial.- Redes sociales.- Embajadores comerciales / fuerza de ventas directa.- Marketing digital segmentado por nicho y ubicación. |
| **Relación con clientes** | - Soporte técnico y atención personalizada.- Panel de administración fácil de usar.- Actualizaciones periódicas incluidas en la licencia.- Tutoriales y documentación. |
| **Fuentes de ingresos** | - Venta de licencias (mensual/anual).- Tarifas adicionales por personalización avanzada.- Servicios opcionales: hosting premium, marketing digital, integración logística. |
| **Recursos clave** | - Backend robusto (NestJS + microservicios).- Infraestructura en Kubernetes + Docker.- App React Native personalizable.- Equipo de soporte y onboarding.- Sistema de pagos integrado multi-moneda. |
| **Actividades clave** | - Desarrollo y mantenimiento de la plataforma.- Implementar personalizaciones para cada cliente.- Integrar pasarelas de pago y métodos de entrega.- Capacitación y soporte. |
| **Socios clave** | - Proveedores de hosting y cloud (AWS/GCP/Azure).- Pasarelas de pago (PayU, Wompi, Stripe).- Empresas de logística local.- Diseñadores y agencias de marketing. |
| **Estructura de costos** | - Desarrollo y mantenimiento.- Infraestructura en la nube.- Personal de soporte.- Marketing y adquisición de clientes.- Licencias de herramientas y APIs externas. |

**Personas (Usuarios Meta)**

**1. Juan Pérez – Dueño de tienda física**

* **Edad:** 38 años
* **Ubicación:** Medellín, Colombia
* **Objetivo:** Llevar su tienda de ropa al mundo digital sin invertir meses en desarrollo.
* **Dolores:** No sabe de tecnología, teme perder ventas por no tener tienda online.
* **Motivadores:** Solución rápida, económica y con su marca.
* **Uso de la app:** Carga productos, recibe pedidos contra entrega.

**2. María Torres – Emprendedora de productos artesanales**

* **Edad:** 29 años
* **Ubicación:** Bogotá, Colombia
* **Objetivo:** Vender sus artesanías en Colombia y también al extranjero.
* **Dolores:** Necesita multi-moneda y no quiere depender solo de Instagram.
* **Motivadores:** Escalar ventas a otros países, opción de USD y pago online.
* **Uso de la app:** Carga fotos y precios, gestiona envíos nacionales e internacionales.

### 3. ****Carlos Jiménez – Dueño de cadena de minimarkets****

* **Edad:** 45 años
* **Ubicación:** Cali, Colombia
* **Objetivo:** Digitalizar catálogo y ofrecer pedidos por la app para clientes frecuentes.
* **Dolores:** Falta de control en pedidos y problemas con apps genéricas.
* **Motivadores:** App con su logo, integración a inventario y métodos de pago mixtos.
* **Uso de la app:** Maneja múltiples sucursales con panel unificado.

Recomendación de Arquitectura en Microservicios

### ****Componentes principales****

1. **Gateway API** (NestJS) – manejo de entrada/salida, autenticación, rate limiting.
2. **Servicio de Autenticación y Licencias** – controla el estado de cada cliente, activa/desactiva tiendas.
3. **Servicio de Catálogo** – productos, categorías, precios, manejo de moneda.
4. **Servicio de Pedidos** – carrito, checkout, historial.
5. **Servicio de Pagos** – integración pasarelas y métodos offline (contra entrega).
6. **Servicio de Notificaciones** – email, push, WhatsApp.
7. **Servicio de Personalización** – colores, logos, nombre de tienda en la app.
8. **Servicio de Usuarios y Roles** – permisos para admins y vendedores.

### ****Infraestructura****

* **Docker** para empaquetar cada microservicio.
* **Kubernetes** para orquestar servicios y escalar horizontalmente.
* **Base de datos**:
  + PostgreSQL para datos transaccionales.
  + Redis para caché y sesiones.
* **Almacenamiento**:
  + S3/MinIO para imágenes de productos.
* **CI/CD** con GitHub Actions o GitLab CI.
* **Observabilidad**:
  + Prometheus + Grafana (métricas).
  + Loki o ELK (logs).