



AnunciaYA v3.0 - Roadmap Maestro

Fecha: 16 Enero 2026

Versión: 5.1

Estado Global: Fases 1-4 | Fase 5.0-5.3.2 | **Fase 5.4** 27% | **Fase 6.0** 10%

Resumen Ejecutivo

Progreso por Fase

Fase	Nombre	Estado	Fecha
1	Monorepo Setup	<input checked="" type="checkbox"/> 100%	Completado
2	Base de Datos	<input checked="" type="checkbox"/> 100%	Completado
3	Backend + Auth	<input checked="" type="checkbox"/> 100%	Completado
4	Frontend Base + Auth UI	<input checked="" type="checkbox"/> 100%	Completado
5.0	Sistema de Modos Backend	<input checked="" type="checkbox"/> 100%	20/12/2024
5.1	Onboarding Backend	<input checked="" type="checkbox"/> 100%	20/12/2024
5.1.1	Onboarding Frontend	<input checked="" type="checkbox"/> 100%	26/12/2024
5.2	Toggle UI + Protección Rutas	<input checked="" type="checkbox"/> 100%	26/12/2024
5.3	Negocios Directorio	<input checked="" type="checkbox"/> 100%	02/01/2026
5.3.1	Sistema Universal Compartir	<input checked="" type="checkbox"/> 100%	02/01/2026
5.3.2	Auth Opcional + ModalAuthRequerido	<input checked="" type="checkbox"/> 100%	16/01/2026
5.4	Business Studio (4/15 módulos)	27%	En progreso
5.4.1	Catálogo CRUD	<input checked="" type="checkbox"/> 100% Completado	07/01/2026
5.4.2	Ofertas CRUD	90% Casi listo	16/01/2026
5.5	ScanYA + PWA	Registrar ventas, dar puntos	~2-3 días
5.6	CardYA + PWA	Usuario ve sus puntos, QR	~2-3 días
5.7	Clientes + Transacciones	Historial de clientes y ventas en BS	~2-3 días
5.8	Opiniones BS	Responder/Ver Reseñas	~2-3 días
5.9	Puntos	Config puntos del negocio en BS	~1 día
5.10	ChatYA + PWA	Mensajería negocio-cliente	~3-4 días
5.11	Cupones	Vista pública + CRUD en BS	~2-3 días
6.0	Ofertas	Vista pública (ruta ya existe)	~1-2 días

Fase	Nombre	Estado	Fecha
6.1	MarketPlace	Vista pública - Compra-venta usuarios	~3-4 días
6.2	Dinámicas	Vista pública + Rifas en BS	Rifas y sorteos
6.3	Empleos	Vista pública + Vacantes en BS	~2-3 días
6.4	Empleados	Gestión empleados en BS (nick + PIN)	~1-2 días
6.5	Sucursales	Agregar/editar sucursales en BS	~2 días
6.6	Reportes + Alertas	Estadísticas y notificaciones en BS	~2-3 días
6.7	Panel Admin	Gestión interna, vendedores, métricas	~1-2 semanas
7	Testing + Deploy	QA y lanzamiento a producción	~1 semana

⌚ NOVEDADES (Enero 16, 2026)

Completado Recientemente

1. Sistema de Auth Opcional (Fase 5.3.2)

Fecha: 16 Enero 2026

Duración: 1 día

Archivos: 13 modificados/creados

Implementación:

- Middleware `verificarTokenOpcional` (backend)
- ModalAuthRequerido para acciones protegidas (frontend)
- Migración de rutas públicas duplicadas a auth opcional
- Eliminación de 6 funciones duplicadas (-33% código)
- Sistema unificado para ofertas, artículos y negocios

Beneficios:

- Zero duplicación de código
- Experiencia fluida con/sin login
- Métricas confiables (solo usuarios reales)
- Conversión mejorada con CTAs contextuales

Documentación: [Auth_Opcional_Sistema_Universal_de_Compartir.md](#)

2. Catálogo CRUD - Business Studio (Fase 5.4.1)

Fecha: 7 Enero 2026

Estado: 100% Completado

Implementación:

- Backend: 7 endpoints CRUD completos
- Frontend BS: 13 componentes con sistema 3 capas
- Integración Perfil Negocio: SecciónCatálogo con cards
- Multi-sucursales: Tabla intermedia `articulo_sucursales`
- Cloudinary: Upload/delete optimizado

Lecciones Aprendidas:

- 1. Sistema de 3 Capas:** Previene race conditions
- 2. IIFE Async:** Para dueños no bloquea UI
- 3. Tabla Intermedia:** Productos pueden estar en múltiples sucursales
- 4. Cloudinary Optimista:** Upload inmediato, rollback si falla

Documentación: [DOC_02_Catalogo_Backend_Frontend_BusinessStudio.md](#)

3. Ofertas CRUD - Business Studio (Fase 5.4.2)

Fecha: 16 Enero 2026

Estado:  90% Completado

Implementación Completa:

- Backend: CRUD completo con 8 endpoints
- Frontend BS: Gestión completa de ofertas
- Integración Perfil Negocio: SecciónOfertas
- Tabla `ofertas` con 5 tipos (2x1, %, \$, combo, happy_hour)
- Sistema multi-sucursales con filtros
- Diseño Glassmorphism Moderno confirmado

Diseño Glassmorphism:

- Barra lateral 4px (identifica tipo de oferta)
- Badge pill con glass effect (valor descuento)
- Overlay oscuro en imagen para contraste
- Backdrop-blur en card (efecto glass)
- Colores por tipo:  2x1,  %,  \$,  Combo,  Happy Hour

Pendiente (10%):

- Modal detalle individual (`ModalDetalleOferta.tsx`)
- Página pública `/ofertas` (feed geolocalizado)
- Tiempo estimado: 4-5 horas

Documentación: [PROMPT_Oferatas_COMPLETO_Fase_5.4.2.md](#)

5.4 Business Studio - Desglose (15 módulos)

#	Módulo	Estado	Depende de
1	Dashboard	<input checked="" type="checkbox"/> Completado	-

#	Módulo	Estado	Depende de
2	Transacciones	⌚ Pendiente	5.5 ScanYA
3	Clientes	⌚ Pendiente	5.5 ScanYA
4	Opiniones	⌚ Pendiente	5.5 ScanYA + 5.7 Transacciones
5	Alertas	⌚ Pendiente	-
6	Catálogo	<input checked="" type="checkbox"/> 100% Completado	07/01/2026
7	Ofertas	⌚ 90% Casi listo	16/01/2026
8	Cupones	⌚ Pendiente	5.11 Cupones
9	Puntos	⌚ Pendiente	5.6 CardYA
10	Rifas	⌚ Pendiente	6.2 Dinámicas
11	Empleados	⌚ Pendiente	5.5 ScanYA
12	Vacantes	⌚ Pendiente	6.3 Empleos
13	Reportes	⌚ Pendiente	5.5 ScanYA
14	Sucursales	⌚ Pendiente	-
15	Mi Perfil	<input checked="" type="checkbox"/> 100% Completado	

⌚ Orden de Implementación Recomendado

Actualizado: 16 Enero 2026

Ya Completado

#	Fase	Estado
1	5.1 Onboarding Comercial	<input checked="" type="checkbox"/> 100%
2	5.2 Toggle UI + Protección Rutas	<input checked="" type="checkbox"/> 100%
3	5.3 Negocios Directorio	<input checked="" type="checkbox"/> 100%
4	5.3.1 Sistema Compartir (base)	<input checked="" type="checkbox"/> Parcial
5	5.4 BS - Dashboard	<input checked="" type="checkbox"/> 100%
6	5.4 BS - Mi Perfil	<input checked="" type="checkbox"/> 100%
7	5.4 BS - Catálogo	<input checked="" type="checkbox"/> 100%
8	5.3.2 Auth Opcional	<input checked="" type="checkbox"/> 100%
9	5.4.2 BS - Ofertas	<input checked="" type="checkbox"/> 90% *

- Pendiente: Modal detalle individual + Página pública [/ofertas](#)

⌚ Siguiente a Implementar

#	Fase	Qué incluye	Tiempo Est.
1	5.4.2	Modal detalle + Página pública /ofertas	
2	5.5	ScanYA + PWA	
3	5.6	CardYA + PWA	
4	5.7	Clientes + Transacciones	
5	5.8	Opiniones	
6	5.9	Puntos	
7	5.10	ChatYA + PWA	
8	5.11	Cupones	
9	6.0	Ofertas	
10	6.1	MarketPlace	
11	6.2	Dinámicas	
12	6.3	Empleos	
13	6.4	Empleados	
14	6.5	Sucursales	
15	6.6	Reportes + Alertas	
16	6.7	Panel Admin	
17	7	Testing + Deploy	

⌚ Flujo de Dependencias

```

5.4.1 Catálogo + 5.4.2 Ofertas
↓
5.5 ScanYA (registrar ventas, otorgar puntos)
↓
5.6 CardYA (usuario ve sus puntos)
↓
5.7 Clientes + Transacciones
↓
5.8 Opiniones (ver y responder reseñas)
↓
5.9 Puntos BS
↓
5.10 ChatYA (comunicación negocio-cliente)
↓
5.11 Cupones + 6.0 Ofertas Públicas
↓

```

6.1 MarketPlace ←
 6.2 Dinámicas (Rifas)
 6.3 Empleos
 ↓
 6.4 Empleados + 6.5 Sucursales + 6.6 Reportes + Alertas
 ↓
 Fase 7: Testing + Deploy

⌚ Tiempo Estimado Total

Bloque	Fases	Tiempo
Inmediato	Completar 5.4.2 Ofertas (modal + página pública)	~1 día
Core Transaccional	5.5 ScanYA + 5.6 CardYA	~4-6 días
Módulos BS dependientes	5.7 Clientes, Transacciones, 5.8 Opiniones, 5.9 Puntos	~4-6 días
Comunicación	5.10 ChatYA	~3-4 días
Cupones/Ofertas	5.11 + 6.0	~3-5 días
Secciones Públicas	6.1, 6.2, 6.3	~8-11 días
BS Restantes	6.4 Empleados, 6.5 Sucursales, 6.6 Reportes, Alertas	~6-8 días
Deploy	Testing + Producción	~1 semana
Total Restante		~6-7 semanas

📊 Progreso General

Módulo	Completado	Total	%
Business Studio	4	15	27%
Fase 5 (Frontend)	~4	~11	~36%
Proyecto Total	Fases 1-4 + parcial 5	7 fases	~60%

🏗 PARTE 1: REQUISITOS ARQUITECTÓNICOS

⚠️ **IMPORTANTE:** Estos requisitos son TRANSVERSALES a todo el proyecto.

1.1 🌎 Geolocalización Global

Aspecto	Descripción
Auto-detección	Al entrar, la app detecta la ciudad automáticamente (GPS → IP/WiFi)
Filtrado automático	TODAS las secciones muestran contenido de la ciudad del usuario

Aspecto	Descripción
Cambio manual	Si el usuario cambia ciudad → TODO se refresca instantáneamente
Store central	<code>useGpsStore</code> es la fuente única de verdad
Backend	PostGIS con <code>ST_DWithin</code> para búsquedas por radio

Secciones afectadas: /negocios, /marketplace, /ofertas, /dinamicas, /empleos

1.2 💬 ChatYA Persistente

Aspecto	Descripción
Montaje	FUERA del <code><Outlet /></code> en MainLayout
Persistencia	NO se cierra al navegar entre secciones
Cierre	<input checked="" type="checkbox"/> o cerrar sesión
Tecnología	Socket.io + MongoDB

1.3 🗃 Sistema de Modos de Cuenta

Aspecto	Descripción
Modelo	1 correo = 1 cuenta = 2 modos posibles
Modo Personal	Siempre disponible (gratis)
Modo Comercial	Requiere pago (\$449 MXN/mes)
Alternancia	Toggle en UI para cambiar de modo
ChatYA	Unificado (mismo historial, diferente "rol")

Modelo de Datos

```
interface Usuario {
    // Identificación
    id: string;
    correo: string;
    nombre: string;

    // Sistema de Modos
    tieneModoComercial: boolean; // True si pagó suscripción
    modoActivo: 'personal' | 'comercial';

    // Datos Modo Personal
    puntosDisponibles: number;
    puntosAcumuladosLifetime: number;
    nivelCardya: 'bronze' | 'plata' | 'oro';
```

```

// Datos Modo Comercial
negocioId?: string;
onboardingCompletado?: boolean;

// Suscripción
stripeCustomerId?: string;
suscripcionActiva: boolean;
}

```

Visibilidad por Modo

Elemento	Personal	Comercial
CardYA	✓	✗
Mis Cupones	✓	✗
MarketPlace (ver)	✓	✓
MarketPlace (publicar)	✓	✗
Favoritos	✓	✓
ChatYA	✓	✓
Business Studio	✗	✓
ScanYA	✗	✓
Mi Negocio	✗	✓

Flujo de Activación Comercial

```

Usuario registrado (modo Personal por defecto)
    ↓
Click "Conocer planes" / "🔒 Comercial"
    ↓
Modal de planes → Pago Stripe ($449/mes)
    ↓
tieneModoComercial = true
    ↓
Redirige a Onboarding Wizard
    ↓
Crea negocio → negocioId asignado
    ↓
Toggle disponible para alternar modos

```

⚠ Degradación Automática por Falta de Pago

Cuando un comerciante deja de pagar:

```

Suscripción vencida/rechazada
↓
Sistema degrada automáticamente a Modo Personal
↓
suscripcionActiva = false
tieneModoComercial = false (temporalmente)
modoActivo = 'personal' (forzado)
↓
BLOQUEO DE DATOS COMERCIALES:
└─ Negocio despublicado (no visible en directorio)
└─ Productos/servicios ocultos
└─ Ofertas despublicadas
└─ Cupones desactivados
└─ Empleos despublicados
└─ Rifas pausadas
└─ Empleados sin acceso a ScanYA
└─ Business Studio bloqueado
↓
DATOS PRESERVADOS (no eliminados):
└─ Información del negocio ✓
└─ Catálogo de productos ✓
└─ Base de clientes ✓
└─ Historial de ventas ✓
└─ Empleados registrados ✓
└─ Configuraciones ✓
↓
Usuario sigue activo en Modo Personal
└─ Puede seguir usando la app como usuario normal

```

Cuando el comerciante paga de nuevo:

```

Pago procesado exitosamente
↓
suscripcionActiva = true
tieneModoComercial = true
↓
REACTIVACIÓN AUTOMÁTICA:
└─ Negocio republicado (visible en directorio)
└─ Productos/servicios visibles
└─ Ofertas reactivadas
└─ Cupones activados
└─ Empleos republicados
└─ Rifas reactivadas
└─ Empleados recuperan acceso
└─ Business Studio desbloqueado
↓
Todo vuelve EXACTAMENTE como estaba
└─ Sin pérdida de datos ni configuraciones

```

Modelo de Datos Actualizado:

```
interface Usuario {
    // Suscripción
    suscripcionActiva: boolean;
    fechaVencimientoSuscripcion?: Date;
    estadoSuscripcion: 'activa' | 'vencida' | 'cancelada';

    // Modo degradado
    modoDegradado: boolean; // True cuando se degrada por falta de pago
    fechaDegradacion?: Date;
}

interface Negocio {
    // Estado
    publicado: boolean; // False cuando suscripción vencida
    motivoDespublicacion?: 'falta_pago' | 'usuario' | 'admin';
}
```

Cron Job de Verificación:

```
// Ejecutar diariamente
async function verificarSuscripcionesVencidas() {
    const hoy = new Date();

    const vencidas = await db.usuarios.find({
        suscripcionActiva: true,
        fechaVencimientoSuscripcion: { $lt: hoy }
    });

    for (const usuario of vencidas) {
        await degradarCuenta(usuario.id);
    }
}

async function degradarCuenta(usuarioId: string) {
    // 1. Actualizar usuario
    await db.usuarios.updateOne({ id: usuarioId }, {
        suscripcionActiva: false,
        tieneModoComercial: false,
        modoActivo: 'personal',
        modoDegradado: true,
        fechaDegradacion: new Date()
    });

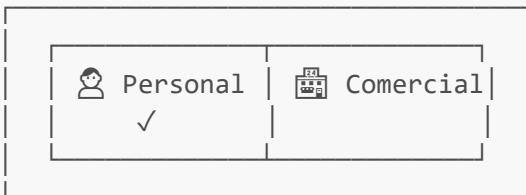
    // 2. Despublicar negocio
    await db.negocios.updateOne({ usuarioId }, {
        publicado: false,
        motivoDespublicacion: 'falta_pago'
    });

    // 3. Desactivar empleados
}
```

```
await db.empleados.updateMany(  
  { negocioId: usuario.negocioId },  
  { activo: false }  
)  
  
// 4. Notificar usuario  
await enviarNotificacion(usuarioId, 'suscripcion_vencida');  
}  
  
async function reactivarCuenta(usuarioId: string) {  
  // Proceso inverso - todo vuelve a la normalidad  
  await db.usuarios.updateOne({ id: usuarioId }, {  
    suscripcionActiva: true,  
    tieneModoComercial: true,  
    modoDegradado: false  
  });  
  
  await db.negocios.updateOne({ usuarioId }, {  
    publicado: true,  
    motivoDespublicacion: null  
  });  
  
  await db.empleados.updateMany(  
    { negocioId: usuario.negocioId },  
    { activo: true }  
  );  
}
```

UI del Toggle

Ambos modos disponibles:



Solo Personal (no ha pagado):



1.4 Negocios Solo Físicos

Decisión: 6 Enero 2026

Aspecto	Descripción
Regla	Todos los negocios requieren ubicación física
Eliminado	Tipo "Online" y columna <code>requiere_direccion</code>
Agregado	<code>tiene_servicio_domicilio</code> y <code>tiene_envio_domicilio</code> en <code>negocio_sucursales</code>

Justificación

Sin local físico	Con local físico
Publica GRATIS en Empleos	Publica en Negocios (PAGO)
Publica GRATIS en MarketPlace	Usa Business Studio
No necesita CardYA	CardYA requiere escaneo presencial

Nuevos Campos

Campo	Tabla	Descripción
<code>tiene_envio_domicilio</code>	<code>negocio_sucursales</code>	Envías productos al cliente
<code>tiene_servicio_domicilio</code>	<code>negocio_sucursales</code>	Tú vas al domicilio del cliente

Documentación: [Eliminación_de_Negocios_Online.md](#)

PARTE 2: FASES DE DESARROLLO

FASE 4: Frontend Base + Auth UI - COMPLETADA

Componentes Implementados

- Setup React + Vite + Tailwind v4
- Stores Zustand (auth, gps, ui, notificaciones)
- Sistema de rutas protegidas
- Landing page con i18n (ES/EN)
- ModalLogin (login, 2FA, recuperar)
- PaginaRegistro + Stripe
- MainLayout responsive
- Navbar desktop (5 tabs + ChatYA)
- MobileHeader + BottomNav
- Sistema GPS con fallback
- Safe areas configuradas

- MenuDrawer (diseño premium)
 - Columnalzquierda (diseño con CardYA + niveles, CTA negocio)
 - ColumnaDerecha (Destacados + Fundadores)
 - PanelNotificaciones (UI + mock)
 - ChatOverlay (UI + mock)
 - Páginas placeholder (cupones, favoritos, publicaciones)
-

FASE 5: Secciones de la App

5.0 Sistema de Modos - Backend COMPLETADO

Implementado el 20 Diciembre 2024

Componente	Estado
Migración BD (2 campos + constraints + índices)	<input checked="" type="checkbox"/>
Schema Drizzle actualizado	<input checked="" type="checkbox"/>
JWT con modoActivo (7 archivos)	<input checked="" type="checkbox"/>
Middleware validarModo.ts	<input checked="" type="checkbox"/>
PATCH /api/auth/modo	<input checked="" type="checkbox"/>
GET /api/auth/modo-info	<input checked="" type="checkbox"/>
Webhook cancelación Stripe	<input checked="" type="checkbox"/>

Endpoints Implementados

```
PATCH /api/auth/modo      → Cambiar modo activo
GET   /api/auth/modo-info → Info del modo actual
```

Middleware Disponibles

```
requiereModoPersonal // Bloquea si está en comercial
requiereModoComercial // Bloquea si está en personal o no pagó
requiereAccesoComercial // Solo verifica que tenga suscripción
```

Archivos Creados/Modificados

- apps/api/src/middleware/validarModo.ts ← NUEVO
- apps/api/src/db/schemas/schema.ts
- apps/api/src/utils/jwt.ts
- apps/api/src/services/auth.service.ts
- apps/api/src/services/pago.service.ts

- `apps/api/src/controllers/auth.controller.ts`
- `apps/api/src/routes/auth.routes.ts`

5.1 Onboarding Wizard - Backend COMPLETADO + Sistema Sucursales

Implementado el 20 Diciembre 2024

Cambio Arquitectónico: Sistema de Sucursales

ANTES:

`negocio` → dirección, teléfono, whatsapp, horarios

AHORA:

```
negocio → ciudad general, correo, sitio_web
  └── sucursales (N)
    ├── Principal (creada en onboarding)
    │   └── dirección, teléfono, whatsapp, horarios, ubicación PostGIS
    └── Adicionales (futuras)
```

Decisiones Arquitectónicas:

- Sistema de Puntos: A nivel NEGOCIO (compartido entre sucursales)
- Transacciones/Canjes: A nivel SUCURSAL (trazabilidad completa)
- Onboarding: Crea solo SUCURSAL PRINCIPAL automáticamente
- Horarios: Independientes por sucursal
- Empleados: Asignados a sucursales específicas

Backend Completado

Componente	Estado
Migración BD + tabla <code>negocio_sucursales</code>	<input checked="" type="checkbox"/>
Schema Drizzle (8 cambios)	<input checked="" type="checkbox"/>
Relations Drizzle (8 cambios)	<input checked="" type="checkbox"/>
15 endpoints REST	<input checked="" type="checkbox"/>
Servicios onboarding	<input checked="" type="checkbox"/>
Controllers	<input checked="" type="checkbox"/>
Routes	<input checked="" type="checkbox"/>
Middleware <code>verificarPropietarioNegocio</code>	<input checked="" type="checkbox"/>
Validaciones Zod	<input checked="" type="checkbox"/>
Compilación TypeScript	<input checked="" type="checkbox"/> 0 errores

Endpoints Implementados

CATEGORÍAS:

GET /api/categorias → Lista 11 categorías
 GET /api/categorias/:id/subcategorias → Lista subcategorías

ONBOARDING (15 endpoints):

POST /api/onboarding/iniciar	→ Crea negocio borrador
POST /api/onboarding/:id/categorias	→ Paso 1: Asignar subcategorías (máx 3)
POST /api/onboarding/:id/ubicacion	→ Paso 2: Crea sucursal principal
POST /api/onboarding/:id/contacto	→ Paso 3: Teléfono/WhatsApp sucursal
POST /api/onboarding/:id/horarios	→ Paso 4: Horarios sucursal
POST /api/onboarding/:id/logo	→ Paso 5a: Logo
POST /api/onboarding/:id/portada	→ Paso 5b: Portada
POST /api/onboarding/:id/galeria	→ Paso 5c: Galería (1-10)
POST /api/onboarding/:id/metodos-pago	→ Paso 6: Métodos de pago
POST /api/onboarding/:id/puntos	→ Paso 7: ¿Participa en puntos?
POST /api/onboarding/:id/articulos	→ Paso 8: Productos (mín 3)
POST /api/onboarding/:id/finalizar	→ Publica negocio
GET /api/onboarding/:id/progreso	→ Estado del onboarding

Archivos Creados/Modificados

```

apps/api/src/
├── db/schemas/
│   ├── schema.ts          # 8 cambios (nueva tabla negocio_sucursales)
│   └── relations.ts       # 8 cambios (relaciones sucursales)
├── services/
│   └── onboarding.service.ts
├── controllers/
│   ├── categorias.controller.ts
│   └── onboarding.controller.ts
├── routes/
│   ├── categorias.routes.ts
│   ├── onboarding.routes.ts
│   └── index.ts           # Actualizado
├── middleware/
│   └── negocio.middleware.ts
└── schemas/
    ├── categorias.schema.ts
    └── onboarding.schema.ts

```

5.1.1 Estandarización Nomenclatura COMPLETADO

Implementado el 21 Diciembre 2024

Cambios Arquitectónicos

Antes:

```
// PostgreSQL usaba snake_case
// TypeScript usaba mezcla de estilos
// API responses inconsistentes
```

Ahora:

```
// PostgreSQL: snake_case (negocio_id, created_at)
//           ↓ Drizzle (transformación automática)
// TypeScript: camelCase (negocioId, createdAt)
//           ↓
// API Response: { success, data, message }
```

Implementación

Componente	Estado
Drizzle casing: 'snake_case' configurado	✓
Transformación automática snake_case ↔ camelCase	✓
API responses en inglés (success, data, message)	✓
439 cambios aplicados en backend	✓
0 errores TypeScript	✓
Backend compila correctamente	✓
Frontend compila correctamente	✓

Archivos Modificados**Configuración Drizzle:**

```
// apps/api/drizzle.config.ts
export default {
  casing: 'snake_case' // ← NUEVO
}
```

Estándar API Response:

```
// Todas las respuestas ahora usan:
{
  success: boolean,
  data?: any,
```

```
    message?: string
}
```

5.1.2 Onboarding Wizard - Frontend COMPLETADO

Completado el 26 Diciembre 2024

Progreso Actual - 100% COMPLETADO

Componente	Estado
Layout estilo Stripe	<input checked="" type="checkbox"/>
Indicador de 8 pasos	<input checked="" type="checkbox"/>
Botones Navegación con validación	<input checked="" type="checkbox"/>
Modal Pausar con guardado	<input checked="" type="checkbox"/>
useOnboardingStore completo	<input checked="" type="checkbox"/>
Paso 1 (Categorías)	<input checked="" type="checkbox"/>
Paso 2 (Ubicación Dual)	<input checked="" type="checkbox"/>
Paso 3 (Contacto + Lada)	<input checked="" type="checkbox"/>
Paso 4 (Horarios)	<input checked="" type="checkbox"/>
Paso 5 (Imágenes + Cloudinary)	<input checked="" type="checkbox"/>
Paso 6 (Métodos de Pago)	<input checked="" type="checkbox"/>
Paso 7 (Sistema de Puntos)	<input checked="" type="checkbox"/>
Paso 8 (Productos/Servicios)	<input checked="" type="checkbox"/>
Sistema de Finalización	<input checked="" type="checkbox"/>
Redirección según onboardingCompletado	<input checked="" type="checkbox"/>
JWT incluye onboardingCompletado	<input checked="" type="checkbox"/>
Cloudinary upload/delete optimista	<input checked="" type="checkbox"/>

Estructura de Archivos

```
apps/web/src/
pages/
  └── private/
    └── business/
      └── onboarding/
        ├── PaginaOnboarding.tsx
        └── pasos/
```



index.ts	✓
PasoCategoria.tsx	✓
PasoUbicacion.tsx	✓ (Sistema Dual)
PasoContacto.tsx	✓ (Lada Editable)
PasoHorarios.tsx	✓
PasoImagenes.tsx	✓ (Cloudinary + Optimización)
PasoMetodosPago.tsx	✓
PasoPuntos.tsx	✓
PasoProductos.tsx	✓
ModalAgregarProducto.tsx	✓
componentes/	
└ onboarding/	
└ LayoutOnboarding.tsx	✓
└ IndicadorPasos.tsx	✓ (7/8 dinámico)
└ BotonesNavegacion.tsx	✓ (Finalizar + Guardar Atrás)
└ ModalPausar.tsx	✓
stores/	
useOnboardingStore.ts	✓

Funcionalidades Implementadas

Layout y Navegación:

- Diseño estilo Stripe (moderno y limpio)
- Indicador visual dinámico (7 u 8 pasos según tipo negocio)
- Botones Atrás/Siguiente con validación y guardado
- Modal "Pausar" con confirmación
- Prevención de salida sin guardar
- Botón "Finalizar y Publicar" en paso final

Paso 1 - Categorías:

- Selección de hasta 3 subcategorías
- Búsqueda y filtrado
- Validación mínimo 1 subcategoría
- Estado persistente en store
- Guardado optimista

Paso 2 - Ubicación:

- Mapa Leaflet interactivo
- Marcador arrastrable
- Detección GPS automática
- Campo ciudad + dirección
- Extracción PostGIS (ST_X/ST_Y)

Paso 3 - Contacto:

- Lada editable (+52, +1, +34, +593, etc.)

- Validación teléfono (10 dígitos)
- Checkbox "Usar mismo número para WhatsApp"
- Campo correo electrónico
- Campo sitio web (opcional)
- Soporte internacional

Paso 4 - Horarios:

- Selector horarios por día (Lun-Dom)
- Toggle 24/7
- Toggle "Cerrado"
- Horario de comida (pausa)
- Copiar horario a todos los días
- Validación rangos

Paso 5 - Imágenes:

- Upload logo (opcional, 500x500px)
- Upload portada (obligatorio, 1600x900px)
- Upload galería (1-10 imágenes, 1200x1200px)
- Drag & drop en las 3 zonas
- Optimización automática a .webp (~90% ahorro)
- Upload optimista (preview instantáneo)
- Eliminación dual (Cloudinary + BD)

Paso 6 - Métodos de Pago:

- Checkbox Efectivo
- Checkbox Tarjeta de crédito/débito
- Checkbox Transferencia bancaria
- Validación mínimo 1 método

Paso 7 - Sistema de Puntos (CardYA):

- Toggle participar/no participar
- Explicación del sistema CardYA
- Sin validación obligatoria

Paso 8 - Productos/Servicios:

- Formulario agregar producto/servicio
- Lista de productos agregados
- Editar producto existente
- Eliminar producto
- Upload imagen por producto (optimizada a .webp)
- Validación mínimo 3 productos para publicar
- Validación mínimo 1 producto para guardar borrador
- Botón "Finalizar y Publicar"

Sistema de Finalización:

- Endpoint POST /api/onboarding/:id/finalizar
- Actualiza onboarding_completado = true
- Actualiza es_borrador = false
- Redirección a Business Studio
- Prevención de loop infinito

Cloudinary:

- Upload de imágenes optimista
- Delete de imágenes optimista
- Previsualización inmediata
- Manejo de errores con rollback
- Optimización automática a .webp
- Upload diferido (evita huérfanos)

Redirección:

- Primera vez con modo comercial → [/business/onboarding](#)
- Usuario puede abortar onboarding y navegar como usuario normal
- Al intentar entrar a [/business-studio](#) se verifica `onboardingCompletado`
- Si `onboardingCompletado: false` → Redirige a [/business/onboarding](#)
- Si `onboardingCompletado: true` → Acceso completo a Business Studio

Endpoints Nuevos Creados:

Método	Ruta	Función
GET	/api/negocios/:id	Obtener negocio
GET	/api/negocios/:id/galeria	Obtener galería
POST	/api/onboarding/:id/logo	Guardar logo
POST	/api/onboarding/:id/portada	Guardar portada
POST	/api/onboarding/:id/galeria	Guardar galería
DELETE	/api/negocios/:id/logo	Eliminar logo
DELETE	/api/negocios/:id/portada	Eliminar portada
DELETE	/api/negocios/:id/galeria/:imagenId	Eliminar imagen

 Decisiones Arquitectónicas - Fase 5.1.1**1. Negocios Solo Físicos (Actualizado 06/01/2026)**

- Todos los negocios requieren ubicación física
- 8 pasos de onboarding (mapa obligatorio)
- Eliminado tipo "Online" y columna `requiere_direccion`
- Agregado `tiene_servicio_domicilio` y `tiene_envio_domicilio`

2. Optimización de Imágenes Client-Side

```
Logo: maxWidth: 500px, quality: 0.85, format: webp
Portada: maxWidth: 1600px, quality: 0.85, format: webp
Galería: maxWidth: 1200px, quality: 0.85, format: webp
Productos: maxWidth: 800px, quality: 0.85, format: webp
```

3. Upload Diferido

- Preview local inmediato (URL.createObjectURL)
- Upload a Cloudinary solo al confirmar
- Evita imágenes huérfanas

4. Validación Flexible de Productos

- Guardar borrador: mínimo 1 producto
- Publicar negocio: mínimo 3 productos

⚠️ Pendientes Documentados - Para Fase 6

Funcionalidad	Estado	Prioridad
Endpoint DELETE /api/negocios/:id	✗ No existe	Media
Limpieza Cloudinary al eliminar negocio	✗ No existe	Media
Endpoint DELETE /api/usuarios/:id	✗ No existe	Baja

Decisión: Implementar a nivel de NEGOCIO (no usuario). Al eliminar negocio → Limpiar Cloudinary → CASCADE en BD.

⌚ Estado y Recomendaciones por Fase

✓ Fases Completadas

Fase	Estado	Fecha
5.1 Onboarding Comercial	✓ 100%	Dic 2024
5.2 Toggle UI + Protección Rutas	✓ 100%	26/12/2024
5.3 Negocios Directorio	✓ 100%	02/01/2026
5.3.1 Sistema Compartir (base)	✓ Parcial	02/01/2026
5.3.2 Auth Opcional + ModalAuthRequerido	✓ 100%	16/01/2026
5.4 BS - Dashboard	✓ 100%	02/01/2026

Fase	Estado	Fecha
5.4 BS - Mi Perfil	<input checked="" type="checkbox"/> 100%	06/01/2026
5.4.1 Catálogo CRUD	<input checked="" type="checkbox"/> 100%	07/01/2026
5.4.2 Ofertas CRUD	<input type="checkbox"/> 90%	16/01/2026

☒ Fase Actual: 5.4 Business Studio (27%)

Completado:

- Layout y navegación
- Router con 15 rutas
- Dashboard con KPIs
- Mi Perfil (6 tabs)
- Catálogo CRUD completo
- Ofertas CRUD (90% - falta modal detalle + página pública)
- Arquitectura de sucursales
- Interceptor Axios automático
- Service centralizado (`negocioManagement.service.ts`)

Siguiente inmediato:

- Completar Ofertas (modal detalle + página pública) - ~1 día
- ScanYA + PWA - ~2-3 días

⌚ Orden de Implementación Recomendado

#	Fase	Descripción	Tiempo	Depende de
5.4.2	Completar Ofertas	Modal detalle + Página pública /ofertas	~1 día	-
5.5	ScanYA + PWA	Registrar ventas, dar puntos	~2-3 días	-
5.6	CardYA + PWA	Usuario ve sus puntos, QR	~2-3 días	5.5 ScanYA
5.7	Clientes + Transacciones	Historial de clientes y ventas en BS	~2-3 días	5.5 ScanYA
5.8	Opiniones	Ver y responder reseñas en BS	~2-3 días	5.5 ScanYA + 5.7
5.9	Puntos	Config puntos del negocio en BS	~1 día	5.6 CardYA
5.10	ChatYA + PWA	Mensajería negocio-cliente	~3-4 días	-
5.11	Cupones	Vista pública + CRUD en BS	~2-3 días	5.5 ScanYA
6.0	Ofertas Públicas	Vista pública (ruta ya existe)	~1-2 días	5.4.2
6.1	MarketPlace	Vista pública, compra-venta usuarios	~3-4 días	-

#	Fase	Descripción	Tiempo	Depende de
6.2	Dinámicas	Vista pública + Rifas en BS	~3-4 días	-
6.3	Empleos	Vista pública + Vacantes en BS	~2-3 días	-
6.4	Empleados	Gestión empleados en BS (nick + PIN)	~1-2 días	5.5 ScanYA
6.5	Sucursales	Agregar/editar sucursales en BS	~2 días	-
6.6	Reportes + Alertas	Estadísticas y notificaciones en BS	~2-3 días	5.5 ScanYA
7	Testing + Deploy	QA y lanzamiento a producción	~1 semana	Todo lo anterior

💡 Recomendaciones Técnicas por Fase

5.4.1 Catálogo:

- Reutilizar `negocioManagement.service.ts` para CRUD
- Modal de creación/edición similar a Onboarding
- Upload de imágenes con optimismo (ya implementado)
- Filtros por tipo (producto/servicio) y categoría
- Tabla `articulos` ya existe en BD

5.4.2 Ofertas:

- Reutilizar `negocioManagement.service.ts` para CRUD
- Tipos: 2x1, %, \$, combo, happy hour
- Configurar días y horarios aplicables
- Toggle activar/pausar oferta
- Vista previa de cómo se verá en público
- Conectar con ruta pública `/p/oferta/:id` (ya existe)

5.5 ScanYA + PWA:

- PWA standalone para uso rápido en caja
- Login empleados: nick + PIN (no email)
- Validar QR con expiración de 2 minutos
- Calcular puntos: `monto / valorPunto * multiplicadorNivel`

5.6 CardYA + PWA:

- QR dinámico que se regenera cada 2 min
- Niveles calculados por `puntos_lifetime` global
- Puntos son específicos por negocio (no transferibles)
- 3 diseños de tarjeta (Bronce, Plata, Oro)

5.7 Clientes + Transacciones:

- Historial de clientes que han comprado
- Lista de transacciones con filtros

- Detalle por cliente (visitas, puntos, nivel)
- Requiere datos de ScanYA (5.5)

5.8 Opiniones:

- Ver todas las reseñas del negocio con filtros
- Responder reseñas de clientes verificados
- Validar que cliente haya comprado antes (requiere ScanYA)
- Métricas: promedio calificación, tasa respuesta
- Templates de respuestas profesionales
- Sistema de reportes para reseñas inapropiadas

5.9 Puntos:

- Configurar valor del punto (1 punto = \$X pesos)
- Activar/desactivar sistema de puntos
- Simulador de acumulación por nivel
- Estadísticas de puntos otorgados

5.10 ChatYA + PWA:

- Socket.io + MongoDB para mensajes
- Identidad según modo (usuario personal o negocio)
- Overlay persistente (no ruta dedicada)
- Reutilizar `obtenerDatosNegocio()` para avatar comercial

Fase 7 - Testing + Deploy:

- Variables de entorno en Railway/Vercel
- Stripe en modo live
- Cloudinary con folder de producción
- Redis para sesiones
- Dominio personalizado + SSL

Progreso General

Área	Completado	Total	%
Fases 1-4 (Base)	4	4	100%
Fase 5 (Frontend)	~4	~11	~36%
Business Studio	4	15	27%
Proyecto Total			~60%

Tiempo estimado restante: ~6-7 semanas

Métricas Fase 5.1.1

Métrica	Valor
---------	-------

Métrica	Valor
Archivos creados/modificados	~20
Líneas de código	~5,000
Endpoints nuevos	8
Bugs resueltos	12
Tiempo de desarrollo	~5 días

5.2 Toggle UI + Protección de Rutas COMPLETADO (26/12/2024)

Permite a usuarios con ambos modos alternar entre Personal y Comercial.

Elemento	Descripción
Función	Alternar modo activo desde el frontend
Quién usa	Usuarios con <code>tiene_modo_comercial: true</code>
Backend	Ya implementado (PATCH /api/auth/modo)

Componentes Creados

Componente	Acción	Descripción
<code>ToggleModoUsuario.tsx</code>	<input checked="" type="checkbox"/> CREADO	Toggle [Personal] [Comercial] reutilizable
<code>ModalCambiarModo.tsx</code>	<input checked="" type="checkbox"/> CREADO	Confirmación antes de cambiar modo
<code>ModoGuard.tsx</code>	<input checked="" type="checkbox"/> CREADO	Guard de protección de rutas
<code>ColumnaIzquierda.tsx</code>	<input checked="" type="checkbox"/> MODIFICADO	Contenido dinámico por modo
<code>MenuDrawer.tsx</code>	<input checked="" type="checkbox"/> MODIFICADO	Toggle + avatar dinámico
<code>BottomNav.tsx</code>	<input checked="" type="checkbox"/> MODIFICADO	Market ↔ Business según modo
<code>Navbar.tsx</code>	<input checked="" type="checkbox"/> MODIFICADO	Toggle + NAV_ITEMS dinámicos

Store Zustand

```
// useAuthStore - Implementado 
interface AuthState {
  tieneModoComercial: boolean;
  modoActivo: 'personal' | 'comercial';
  negocioId?: string;
  // Datos del negocio (modo comercial)
  nombreNegocio?: string;
  correoNegocio?: string;
  logoNegocio?: string;
  fotoPerfilNegocio?: string;
```

```
cambiarModo: (modo: 'personal' | 'comercial') => Promise<void>;
}
```

UI del Toggle

Ambos modos: [👤 Personal ✓] [💻 Comercial]
 Solo Personal: [👤 Personal ✓] [🔓 Desbloquear]

Protección de Rutas Frontend

```
// Rutas PÚBLICAS (sin login requerido)
/dinamicas/rifa/:id/sorteo-publico    // Ver sorteo en vivo (rifas offline)
/dinamicas/resultado/:id              // Ver resultado de sorteo público

// Rutas que requieren modo Personal
/marketplace/publicar
/puntos/canjear
/card

// Rutas que requieren modo Comercial
/business/*
/scan/*

// Rutas que requieren estar logueado (cualquier modo)
/dinamicas/*                      // Excepto las públicas de arriba
/negocios/*
/ofertas/*
/chat/*
/perfil/*
```

⚠️ Excepción Importante: Rifas Offline Públicas

Contexto: Las rifas offline permiten participantes sin registro en la app.

Rutas públicas necesarias:

Ruta	Acceso	Propósito
/dinamicas/rifa/:id/sorteo-publico	SIN LOGIN	Ver sorteo en vivo transmitido
/dinamicas/resultado/:id	SIN LOGIN	Ver resultado del sorteo
/dinamicas/compartir/:id	SIN LOGIN	Vista compatible de la rifa

Implementación:

```
// routes.jsx
const publicRoutes = [
  {
    path: '/dinamicas/rifa/:id/sorteo-publico',
    element: <SorteoPublicoLive />,
    auth: false // ← No requiere login
  },
  {
    path: '/dinamicas/resultado/:id',
    element: <ResultadoPublico />,
    auth: false // ← No requiere login
  }
];

const privateRoutes = [
  {
    path: '/dinamicas/*',
    element: <Dinamicas />,
    auth: true, // ← Requiere login para crear/administrar
    guard: <AuthGuard />
  }
];
```

Flujo de Acceso:

Usuario NO logueado:

- └─ Puede ver sorteo en vivo (link compartido)
- └─ Puede ver resultado (link compartido)
- └─ NO puede crear rifas ni gestionar

Usuario logueado:

- └─ Puede crear rifas
- └─ Puede gestionar rifas
- └─ Puede compartir links públicos
- └─ Puede ver sorteos públicos

Componente de Sorteo Público:

```
// SorteoPublicoLive.tsx
export default function SorteoPublicoLive() {
  const { rifaId } = useParams();
  const { data, loading } = useSorteoPublico(rifaId);

  // No requiere auth
  // Muestra sorteo en tiempo real
  // Link: /dinamicas/rifa/abc123/sorteo-publico

  return (
    <LayoutPublico> {/* Sin navbar de usuario */}
  )
}
```

```

<SorteoEnVivoPublico rifa={data} />
<BannerRegistro /> {/* CTA para registrarse */}
</LayoutPublico>
);
}

```

Banner de Invitación:

¿Quieres organizar tus propias rifas?
 [Regístrate gratis] [Conocer más]

5.3 Negocios Locales ([/negocios](#)) COMPLETADO (02/01/2026)

Implementado:

- Vista Mapa con Leaflet
- Tarjeta preview en mapa (carrusel, rating, contacto)
- Filtros (distancia, categoría, CardYA, envío)
- Perfil completo (header, métricas, horario)
- Catálogo desde BD con búsqueda y paginación
- Galería con paginación
- Reseñas desde BD
- Mapa en sidebar del perfil
- Redes sociales dinámicas
- Métodos de pago
- WhatsApp + ChatYA botones
- 3 resoluciones (móvil, laptop, desktop)

Pendientes (Bloqueados):

Pendiente	Bloqueador
Conectar ChatYA con Negocios	Requiere 5.10 ChatYA
Validar compra para reseña	Requiere 6.1 ScanYA

Elemento	Descripción
Función	Directorio de negocios geolocalizados
Quién publica	Usuarios con modo comercial (via Onboarding)
Geolocalización	<input checked="" type="checkbox"/> PostGIS - ST_DWithin

Componentes Frontend:

```

pages/private/negocios/
├── index.ts
└── PaginaNegocios.tsx          # Lista + Mapa
└── PaginaPerfilNegocio.tsx     # Perfil del negocio

components/negocios/
├── index.ts
├── ModalDetalleItem.tsx        # Modal producto/servicio
├── ModalHorarios.tsx           # Modal horarios completos
├── PanelFiltros.tsx            # Filtros laterales
└── SeccionCatalogo.tsx         # Catálogo del negocio

hooks/
├── useListaNegocios.ts
├── usePerfilNegocio.ts
├── useHorariosNegocio.ts
└── useVotos.ts

stores/
└── useFiltrosNegociosStore.ts

services/
└── negociosService.ts

```

Backend:

```

controllers/
├── negocios.controller.ts
├── resenas.controller.ts
└── votos.controller.ts
└── metricas.controller.ts

services/
├── negocios.service.ts
├── resenas.service.ts
└── votos.service.ts
└── metricas.service.ts

routes/
├── negocios.routes.ts
├── resenas.routes.ts
└── votos.routes.ts
└── metricas.routes.ts

```

Funcionalidades:

- Lista de negocios cercanos (suscripto a useGpsStore)
- Filtros por categoría/subcategoría
- Búsqueda por nombre

- Vista mapa con marcadores
- Perfil de negocio (galería, horarios, catálogo)
- Botón "Contactar" → ChatYA (*bloqueado por 5.8*)
- Reseñas validadas por compra (*bloqueado por 6.1 ScanYA*)

Restricción de Reseñas:

Para dejar reseña/calificación:

- └─ Usuario debe tener pedido completado con el negocio
- └─ Backend valida historial de pedidos
- └─ Si no hay pedidos → Botón "Calificar" deshabilitado

Validación:

```
SELECT COUNT(*) FROM pedidos
WHERE usuario_id = ?
    AND negocio_id = ?
    AND estado = 'completado'
```

5.3.1 Sistema Universal de Compartir PARCIAL (02/01/2026)

Se implementó la base del sistema. Rutas adicionales se agregan conforme se completan las fases correspondientes.

TODAS las publicaciones pueden compartirse fuera de la app y verse SIN REGISTRO

Rutas Públicas para Todo el Contenido

Tipo de Contenido	Ruta Pública	Estado	Fase
Negocio	/p/negocio/:id	<input checked="" type="checkbox"/> Implementado	5.3
Artículo	/p/articulo/:id	<input checked="" type="checkbox"/> Implementado	5.3
Oferta	/p/oferta/:id	<input checked="" type="checkbox"/> Implementado	5.3
Cupón	/p/cupon/:codigo	<input type="checkbox"/> Pendiente	5.11
Marketplace	/p/marketplace/:id	<input type="checkbox"/> Pendiente	5.9
Empleo	/p/empleo/:id	<input type="checkbox"/> Pendiente	5.11
Rifa	/p/rifa/:id	<input type="checkbox"/> Pendiente	5.10

Cada vista pública incluye:

- Contenido completo visible
- Imágenes y galería
- Información del publicador
- Botón de registro/descarga de app
- Metadatos Open Graph (preview en redes sociales)

Archivos Creados

Frontend:

```

components/compartir/
└── index.ts
└── BannerRegistro.tsx      # CTA para usuarios no logueados
    └── DropdownCompartir.tsx  # Menú WhatsApp, Facebook, Twitter, Copiar

components/layout/
└── LayoutPublico.tsx       # Layout sin navbar para vistas públicas

hooks/
└── useOpenGraph.ts          # Meta tags dinámicos

pages/public/
└── PaginaNegocioPublico.tsx
└── PaginaArticuloPublico.tsx
└── PaginaOfertaPublico.tsx

```

Backend:

GET /api/negocios/publico/:id	✓
GET /api/articulos/publico/:id	✓
GET /api/ofertas/publico/:id	✓

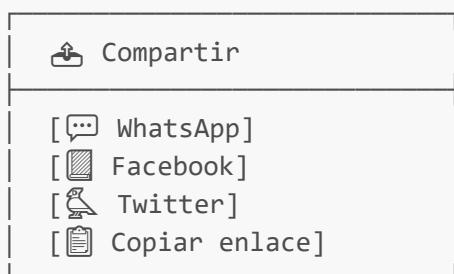
Componente "Compartir" Universal

```

// En TODAS las pantallas de detalle
<DropdownCompartir
  url={`/p/negocio/${id}`}
  titulo="Pizzería Roma"
/>

```

// Dropdown al hacer click



Ejemplo: Vista Pública de Negocio

URL: <https://anunciaya.com/p/negocio/abc123>

 Pizzería Roma
★ 4.8 (234 reseñas) • 🗺 CDMX
[Galería de imágenes]

 INFORMACIÓN
⌚ Horario: Lun-Dom 1pm - 11pm
📞 Teléfono: 55-1234-5678
🌐 Web: www.pizzerraroma.com
📍 Dirección: Av. Principal 123
[Ver en mapa]

 MENÚ (6 productos)
• Margarita \$120
• Pepperoni \$140
[Ver menú completo]

 OFERTAS ACTIVAS (2)
• 2x1 Martes
• 20% Happy Hour

 ¿Quieres acumular puntos y usar cupones?
[Descargar AnunciaYA] [Registrarse gratis]

Ejemplo: Vista Pública de Oferta

URL: <https://anunciaya.com/p/oferta/xyz789>

 2x1 en Hamburguesas
📍 Burger House
[Imagen de la oferta]

 VIGENCIA
• Válido: Todos los martes
• Horario: 1pm - 11pm

 CONDICIONES
• Solo hamburguesas sencillas
• No aplica con otras promociones
• Dine-in y para llevar

 UBICACIÓN
Burger House
Av. Reforma 456, CDMX
[Cómo llegar]

[Regístrate para guardar esta oferta
[Descargar app] [Registrarse]

Metadata Open Graph (Preview en Redes)

```
<!-- Negocio -->
<meta property="og:title" content="Pizzería Roma | AnunciaYA" />
<meta property="og:description" content="⭐ 4.8 - La mejor pizza de la ciudad" />
<meta property="og:image"
content="https://cdn.anunciaya.com/negocios/abc/portada.jpg" />

<!-- Oferta -->
<meta property="og:title" content="2x1 en Hamburguesas - Burger House" />
<meta property="og:description" content="Todos los martes 1pm-11pm" />
<meta property="og:image"
content="https://cdn.anunciaya.com/ofertas/xyz/imagen.jpg" />

<!-- Marketplace -->
<meta property="og:title" content="iPhone 12 Pro - $8,000" />
<meta property="og:description" content="128GB, excelente estado" />
<meta property="og:image"
content="https://cdn.anunciaya.com/marketplace/qwe/foto1.jpg" />
```

Backend - Endpoints Públicos

```
// GET /api/public/:tipo/:id
// NO requiere autenticación
app.get('/api/public/:tipo/:id', async (req, res) => {
  const { tipo, id } = req.params;

  const data = await getContenidoPublico(tipo, id);

  // Verificar que esté publicado
  if (!data || !data.publicado) {
    return res.status(404).json({ error: 'Contenido no encontrado' });
  }

  // Incrementar contador de vistas
  await incrementarVistas(tipo, id, req.headers['referer']);

  res.json(data);
});
```

Analytics para Comerciantes (*Pendiente - Módulo Reportes BS*)

ESTADÍSTICAS DE COMPARTIDOS

Tu negocio:

- 234 vistas desde links compartidos
- 45 clicks en "Ver ubicación"
- 18 clicks en "Descargar app"
- 12 registros atribuidos a tu link

Origen de visitas:

- WhatsApp: 65%
- Facebook: 25%
- Directo: 10%

[\[Ver detalles\]](#) [\[Compartir de nuevo\]](#)

Funcionalidades del Sistema

- Botón "Compartir" visible
- Generar link público
- Vista pública sin login
- Banner de registro
- Metadatos Open Graph
- Tracking de vistas (*pendiente*)
- Deep linking a la app (*pendiente*)
- Analytics para comerciantes (*pendiente - módulo Reportes BS*)

Restricciones:

- Ver contenido: SIN LOGIN
- Contactar/Aplicar/Canjear: CON LOGIN
- Descargar app desde vista pública

5.4 Business Studio - Panel de Control EN PROGRESO (15%)

Centro de administración completo para negocios. Gestión de perfil, catálogos, clientes, ventas, cupones, ofertas, empleos, rifas, y análisis de negocio.

Acceso: [/business-studio/*](#) (Requiere modo Comercial)

Progreso: 4 de 15 módulos completados

Resumen de Módulos Business Studio

#	Módulo	Estado	Fase	Depende de
1	Dashboard	<input checked="" type="checkbox"/> Completado	-	-
2	Transacciones	 Pendiente	5.7	5.5 ScanYA

#	Módulo	Estado	Fase	Depende de
3	Clientes	<input type="checkbox"/> Pendiente	5.7	5.5 ScanYA
4	Opiniones	<input type="checkbox"/> Pendiente	5.8	5.5 ScanYA + 5.7
5	Alertas	<input type="checkbox"/> Pendiente	-	-
6	Catálogo	<input checked="" type="checkbox"/> 100% Completado	5.4.1	07/01/2026
7	Ofertas	<input type="checkbox"/> 90% Casi listo	5.4.2	16/01/2026
8	Cupones	<input type="checkbox"/> Pendiente	5.11	5.5 ScanYA
9	Puntos	<input type="checkbox"/> Pendiente	5.9	5.6 CardYA
10	Rifas	<input type="checkbox"/> Pendiente	6.2	6.2 Dinámicas
11	Empleados	<input type="checkbox"/> Pendiente	6.4	5.5 ScanYA
12	Vacantes	<input type="checkbox"/> Pendiente	6.3	6.3 Empleos
13	Reportes	<input type="checkbox"/> Pendiente	6.6	5.5 ScanYA
14	Sucursales	<input type="checkbox"/> Pendiente	6.5	-
15	Mi Perfil	<input checked="" type="checkbox"/> 100% Completado	-	-

Arquitectura Implementada (01-06 Enero 2026)

Decisión Arquitectónica - Negocios Solo Físicos (06/01/2026):

- Eliminado tipo "Online" - Todos los negocios requieren ubicación física
- Columna `requiere_direccion` eliminada de BD
- Agregado `tiene_servicio_domicilio` en `negocio_sucursales`
- Agregado `tiene_envio_domicilio` en `negocio_sucursales`
- Documentación: `Eliminación_de_Negocios_Online.md`

Sistema de Sucursales:

- Arquitectura donde usuario ve "negocio" pero internamente opera sobre SUCURSALES
- Middleware `verificarNegocio` corregido para dueños Y gerentes
- Middleware `validarAccesoSucursal` para validar acceso por sucursal
- Interceptor Axios agrega `?sucursalId=` automático en modo comercial
- Filtros SQL manuales en backend con `WHERE sucursal_id = ?`

Service Centralizado:

- `negocioManagement.service.ts` con 15 funciones CRUD reutilizables
- Onboarding y Business Studio comparten las mismas funciones
- Controllers solo llaman a services, nunca duplican lógica

Archivos Creados

Frontend - Layout y Navegación:

```
components/layout/
├── DrawerBusinessStudio.tsx      # Drawer móvil con menú BS
├── MenuBusinessStudio.tsx        # Menú lateral desktop
└── PanelPreviewNegocio.tsx       # Preview del negocio con tabs

router/
└── index.tsx                      # 15 rutas de BS configuradas
```

Frontend - Dashboard:

```
pages/private/business-studio/dashboard/
├── PaginaDashboard.tsx
└── componentes/
    ├── index.ts
    ├── HeaderDashboard.tsx
    ├── KPIPrincipal.tsx
    ├── KPISecundario.tsx
    ├── GraficaVentas.tsx
    ├── PanelActividad.tsx
    ├── PanelAlertas.tsx
    ├── PanelCampanas.tsx
    ├── PanelOpiniones.tsx
    ├── FooterAcciones.tsx
    └── SelectorSucursalesInline.tsx

stores/
└── useDashboardStore.ts

services/
└── dashboardService.ts
```

Frontend - Mi Perfil:

```
pages/private/business-studio/perfil/
├── PaginaPerfil.tsx
└── hooks/
    └── usePerfil.ts

└── components/
    ├── index.ts
    ├── TabInformacion.tsx
    ├── TabUbicacion.tsx
    ├── TabContacto.tsx
    ├── TabHorarios.tsx
    ├── TabOperacion.tsx      # Métodos pago + Servicios entrega
    └── TabImagenes.tsx
```

```

    └── SelectorCategoria.tsx
    └── CardYA.tsx

```

Backend:

```

controllers/
└── dashboard.controller.ts

services/
├── dashboard.service.ts
└── negocioManagement.service.ts # 15 funciones CRUD centralizadas

routes/
└── dashboard.routes.ts

middleware/
└── sucursal.middleware.ts          # validarAccesoSucursal

```

Endpoints Dashboard:

```

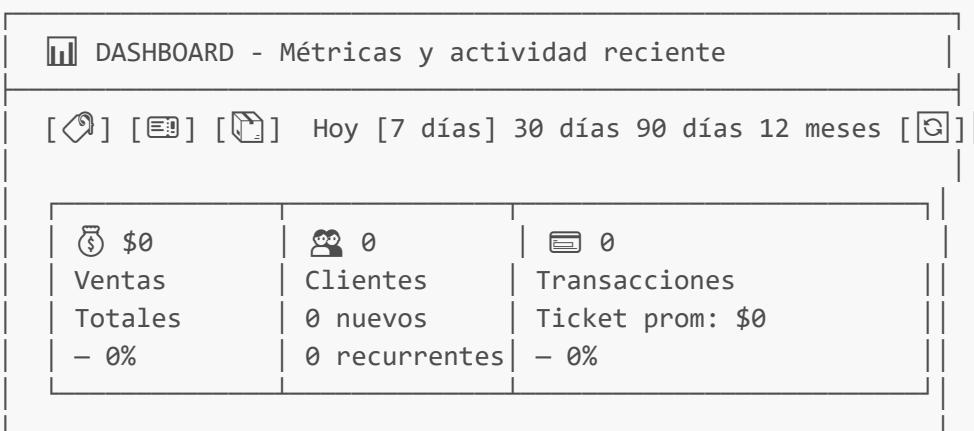
GET /api/dashboard/kpis
GET /api/dashboard/grafica-ventas
GET /api/dashboard/actividad-reciente
GET /api/dashboard/alertas
GET /api/dashboard/campanas
GET /api/dashboard/opiniones
GET /api/dashboard/resumen

```

Módulos de Business Studio

1. Dashboard ([/business-studio](#)) - Vista General COMPLETADO

Vista Implementada:



0	3	2	2	0	8
Cup.	Ofr.	Flw.	Lks.	Rtg.	Vist

<input checked="" type="checkbox"/> Ventas del Periodo Evolución diaria Promedio/día: \$0 Mejor: - [Gráfica vacía]	Cupones y Ofertas 3 activas	
	<ul style="list-style-type: none"> En todas tus compras REVISAR... DESAYUNOS 	
	0 0 0 2d 0 0 0 23d 0 0 0 50d	
	[Ver Cupones] [Ver Ofrt.]	
Interacciones Actividad	Opiniones Reseñas recientes	Alertas Al día
Juan Manuel Comenzó a seguirte (2h)	Sin reseñas recientes	Todo bien
Juan Manuel Le gustó (2h)		Sin alert. pendientes

KPIs Implementados:

Principales (3 cards grandes):

- Ventas Totales - Monto total del período
- Clientes - Total, nuevos y recurrentes
- Transacciones - Total y ticket promedio

Secundarios (6 cards pequeños):

- Cupones Canjeados
- Ofertas Activas
- Followers (seguidores)
- Likes del negocio
- Rating Perfil - Promedio y total reseñas
- Vistas del Perfil

Paneles Implementados:

1. Ventas del Periodo:

- Evolución diaria con gráfica de líneas
- Promedio por día
- Mejor día de ventas

- Estado vacío: "No hay datos de ventas para este período"

2. Cupones y Ofertas:

- Lista de cupones/ofertas activos (3 visibles)
- Métricas por ítem: vistas, usos, compartidos, tiempo restante
- Botones: [Ver Cupones] [Ver Ofertas]

3. Interacciones:

- Actividad de clientes (follows, likes)
- Timestamp relativo (2h, 1d)
- Avatar del cliente

4. Opiniones:

- Últimas reseñas recibidas
- Estado vacío: "Sin reseñas recientes"

5. Alertas:

- Estado actual: "Todo bien" o pendientes
- Lista de alertas con prioridad
- Estado vacío: "Sin alertas pendientes"

Selectores Superiores:

- 3 botones de acción rápida (🔗 📁 📄)
- Selector de período: Hoy, 7 días, 30 días, 90 días, 12 meses
- Botón refresh para actualizar datos

Componentes Implementados:

- HeaderDashboard - Título + descripción
- SelectorPeriodo - Botones de tiempo
- KPIPrincipal - 3 cards grandes (Ventas, Clientes, Transacciones)
- KPISecundario - 6 cards pequeños (métricas adicionales)
- GraficaVentas - Panel con gráfica de líneas
- PanelCuponesOfertas - Lista de promociones activas
- PanelInteracciones - Actividad de clientes
- PanelOpiniones - Últimas reseñas
- PanelAlertas - Estado de alertas

2. Transacciones ([/business-studio/transacciones](#))  PENDIENTE (Requiere 6.1 ScanYA)

Historial Completo de Ventas:

 TRANSAZIONI
[Hoy] [Semana] [Mes] [Rango personalizzato]

Total del período: \$45,280 • 89 transacciones

#0089 • 22 Dic 2024 - 2:45 PM \$325.00
Cliente: Juan Pérez 🎖 • +32 puntos otorgados
Productos: 2 hamburguesas, 1 refresco
Método: Efectivo • [Ver detalle]

Funcionalidades:

- Ver todas las transacciones
- Filtrar por fecha
- Filtrar por método de pago
- Filtrar por sucursal
- Ver detalle de transacción
- Exportar reporte (CSV/PDF)
- Estadísticas de ventas
- Ticket promedio
- Productos más vendidos

3. Clientes ([/business-studio/clientes](#)) PENDIENTE (Requiere 6.1 ScanYA)

Gestión de Base de Clientes:

 BASE DE CLIENTES
Total: 234 clientes • Nuevos este mes: 18

 Buscar... [Filtros: Todos | Bronce | Plata | Oro]

 María González Nivel: ORO
1,250 puntos • 15 visitas •  Cliente desde
Última compra: hace 3 días Feb 2024
[Ver Perfil] [Historial] [ Chat]

Información por Cliente:

- Nombre y foto
- Nivel CardYA (Bronce/Plata/Oro)
- Puntos acumulados EN TU NEGOCIO
- Total de visitas
- Última compra
- Cliente desde (fecha)

- Historial de compras
- Cupones canjeados
- Botón para chatear

Filtros:

- Por nivel CardYA
- Por puntos
- Por frecuencia de visita
- Por fecha de registro
- Clientes inactivos (>30 días sin comprar)

4. Opiniones ([/business-studio/opiniones](#)) - Gestión de Reseñas PENDIENTE

Permite ver y responder reseñas de clientes, mejorando la reputación y engagement del negocio.

Acceso:

- Business Studio (web) - Dueños y gerentes
- ScanYA (PWA) - Empleados autorizados

Requiere: Fase 5.5 ScanYA (validación de compras)

Vista Propuesta:

💬 OPINIONES - Reseñas y Calificaciones

☆ 4.3 promedio • 24 reseñas • 87% tasa de respuesta

[ Estadísticas] [ Responder pendientes (3)]

Filtros: [Todas ▼] [Calificación ▼] [Sucursal ▼]

 María López  Hace 2 días
"Excelente servicio, muy rápida la atención..."

💬 Tu respuesta:
"¡Gracias María! Nos alegra que hayas disfrutado..."
Respondida por: Juan (Empleado) • Hace 1 día
[ Editar] [ Eliminar]

 Juan Pérez  Hace 5 días
"Buen producto pero tardó mucho la entrega"

[ Responder] [ Usar plantilla] [ Reportar]

[1] [2] [3] ... [8] Mostrando 1-10 de 24 reseñas

Funcionalidades Principales:

1. Dashboard de Opiniones:

- Promedio de calificación general (⭐ 1-5)
- Total de reseñas recibidas
- Tasa de respuesta (% respondidas)
- Distribución de calificaciones (gráfico de barras)
- Tiempo promedio de respuesta
- Comparativa entre sucursales

2. Gestión de Reseñas:

- Lista completa de reseñas con paginación
- Filtros por:
 - Calificación (1-5 estrellas)
 - Estado (todas, sin responder, respondidas)
 - Sucursal (multi-branch)
 - Fecha (más recientes, más antiguas)
- Búsqueda por nombre de cliente
- Ordenar por: fecha, calificación, sin responder primero

3. Responder Reseñas:

- Campo de texto para escribir respuesta
- Templates pre-escritos sugeridos:
 - Agradecimiento positivo
 - Disculpa profesional
 - Solicitud de contacto
 - Invitación a regresar
- Preview antes de publicar
- Editar respuestas existentes
- Eliminar respuestas propias
- Contador de caracteres (máx 500)
- **Atribución de respuesta:** Sistema registra quién respondió (dueño/gerente/empleado)

4. Acceso desde ScanYA (PWA):

Vista Móvil Simplificada:

Reseñas Pendientes

3 sin responder

☆☆☆☆☆

Maria López • Hace 2 días
"Excelente servicio..."
[Responder] [Template]

☆☆☆☆☆

Juan Pérez • Hace 5 días
"Buen producto pero..."
[Responder] [Template]

Funcionalidades en ScanYA:

- Ver reseñas sin responder (solo sucursal asignada)
- Responder con templates rápidos
- Notificación push de nuevas reseñas
- Badge con contador en menú ScanYA
- Interface optimizada para móvil
- Permisos por empleado (configurables desde BS)

Sistema de Permisos:

```
// Tabla empleados
interface Empleado {
  id: string;
  negocio_id: string;
  sucursal_id: string;

  // Permisos
  puede_responder_reseñas: boolean; // ← Nuevo permiso
  puede_editar_respuestas: boolean; // ← Solo sus propias respuestas
  puede_ver_todas_reseñas: boolean; // ← O solo de su sucursal
}
```

Flujo de Respuesta desde ScanYA:

Empleado abre ScanYA
 ↓
 Ve badge "3 reseñas pendientes"
 ↓
 Entra a módulo Opiniones
 ↓
 Ve solo reseñas de su sucursal
 ↓
 Selecciona reseña sin responder
 ↓

```

    Escoge template o escribe respuesta
    ↓
    Publica respuesta
    ↓
    Sistema registra: "Respondida por Juan (Empleado)"
    ↓
    Cliente recibe notificación
    ↓
    Respuesta aparece en perfil público del negocio
  
```

5. Validación de Clientes:

- Solo clientes con compras pueden reseñar
- Verificación automática en backend:

```

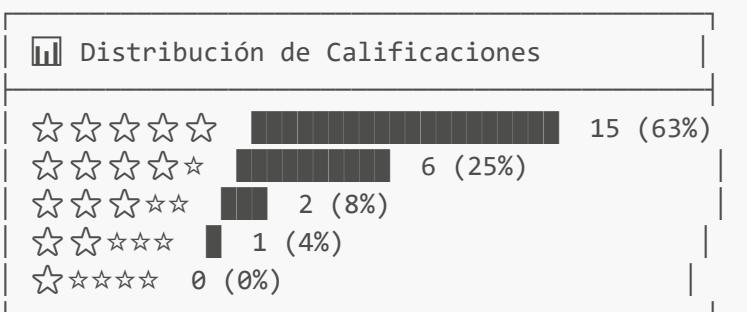
SELECT COUNT(*) FROM transacciones
WHERE usuario_id = ?
  AND negocio_id = ?
  AND sucursal_id = ?
  
```

- Badge "Compra verificada" en reseñas válidas
- Límite: 1 reseña cada 30 días por cliente

6. Moderación:

- Reportar reseñas inapropiadas
- Motivos: spam, lenguaje ofensivo, falsa, duplicada
- Envío a revisión de admin
- Opción de responder antes de reportar
- Solo dueño/gerente puede reportar (no empleados)

7. Estadísticas Detalladas:



Métricas clave:

- Promedio general: 4.3/5.0
- Total de reseñas: 24
- Respondidas: 21 (87%)
- Sin responder: 3
- Tiempo promedio respuesta: 4.2 horas

- Última reseña: Hace 2 días

Respuestas por:

- Dueño: 15 (71%)
- Gerente: 4 (19%)
- Empleados: 2 (10%)

Componentes Frontend:

Business Studio (Web):

```
pages/private/business-studio/opiniones/
├── PaginaOpiniones.tsx
└── hooks/
    ├── useOpiniones.ts      (Sistema 3 capas)
    └── useMetricasOpiniones.ts (Estadísticas)
└── components/
    ├── HeaderOpiniones.tsx      (Título + promedio)
    ├── EstadisticasOpiniones.tsx (Dashboard métricas)
    ├── FiltrosOpiniones.tsx     (Filtros y búsqueda)
    ├── ListaOpiniones.tsx       (Grid de reseñas)
    ├── CardOpinion.tsx         (Card individual)
    ├── ModalResponder.tsx       (Modal para responder)
    ├── ModalEditarRespuesta.tsx (Editar respuesta)
    ├── TemplatesRespuesta.tsx   (Sugerencias)
    ├── EstrellaCalificacion.tsx (★★★★★)
    ├── DistribucionCalificaciones.tsx (Gráfico barras)
    ├── BadgeEstado.tsx          (Respondida/Sin responder)
    └── BotonReportar.tsx        (Reportar reseña)
```

ScanYA (PWA):

```
apps/scan-ya/pages/opiniones/
├── PaginaOpinionesScan.tsx      (Vista móvil simplificada)
└── hooks/
    └── useOpinionesScan.ts       (Solo sucursal asignada)
└── components/
    ├── ListaResenasMovil.tsx    (Lista optimizada)
    ├── CardResenaMovil.tsx      (Card compacta)
    ├── ModalResponderRapido.tsx (Modal móvil)
    └── TemplatesRapidos.tsx     (Templates comunes)
```

Backend Endpoints:

```
GET  /api/resenas           # Listar reseñas del negocio (con filtros)
GET  /api/resenas/:id        # Ver detalle de una reseña
POST /api/resenas/:id/responder # Responder a una reseña
```

```

PUT      /api/resenas/:id/respuesta # Editar respuesta existente
DELETE   /api/resenas/:id/respuesta # Eliminar respuesta
GET      /api/resenas/metricas    # Estadísticas de reseñas
PATCH    /api/resenas/:id/leida   # Marcar como leída
POST     /api/resenas/:id/reportar # Reportar reseña inapropiada

# Específicos para ScanYA
GET      /api/resenas/sucursal/:id/pendientes # Solo sin responder de una sucursal
POST     /api/resenas/pleado/:id/responder   # Responder como empleado

```

Service Functions:

```

// resenas.service.ts
obtenerResenasNegocio(negocioId, sucursalId?, filtros?)
obtenerResenaSucursal(sucursalId, soloSinResponder?)
obtenerResenaPorId(resenaId)
responderResena(resenaId, respuesta, usuarioId, tipoUsuario)
responderComoEmpleado(resenaId, respuesta, empleadoId)
editarRespuesta(resenaId, nuevaRespuesta, usuarioId)
eliminarRespuesta(resenaId, usuarioId)
obtenerMetricasResenas(negocioId, sucursalId?)
marcarComoLeida(resenaId)
reportarResena(resenaId, motivo)
verificarComprasUsuario(usuarioId, negocioId, sucursalId)
verificarPermisoEmpleado(empleadoId, permiso)

```

Tabla de Respuestas:

```

ALTER TABLE negocio_resenas ADD COLUMN respondida_por_tipo VARCHAR(20);
-- Valores: 'dueño', 'gerente', 'empleado'

ALTER TABLE negocio_resenas ADD COLUMN respondida_por_nombre VARCHAR(100);
-- Nombre del empleado que respondió

ALTER TABLE negocio_resenas ADD COLUMN respondida_desde VARCHAR(20);
-- Valores: 'business_studio', 'scan_ya', 'app_movil'

```

Templates de Respuesta Sugeridos:

Positiva (☆☆☆☆☆):

- "¡Gracias [nombre]! Nos alegra que hayas disfrutado tu experiencia. ¡Te esperamos pronto!"
- "Apreciamos mucho tu comentario. Es un placer atenderte."

Neutral (☆☆☆☆☆):

- "Gracias por tu feedback [nombre]. Trabajaremos para mejorar en..."
- "Tomamos nota de tus comentarios para seguir mejorando."

Negativa (☆☆☆☆):

- "Lamentamos que tu experiencia no haya sido la esperada. Nos gustaría conversar contigo para resolver esto. Por favor contáctanos al [teléfono]."
- "Sentimos mucho lo ocurrido. Hemos tomado medidas para que no vuelva a suceder."

UX Considerations:

1. Respuesta Rápida:

- Botón "Responder" visible desde la lista
- Modal simple con textarea grande
- Guardado con Ctrl+Enter (web) o botón "Enviar" (móvil)

2. Notificaciones:

- Badge en menú BS/ScanYA con reseñas sin responder
- Push notification cuando llega reseña nueva
- Email al propietario (configurable)
- Notificación a empleados en su turno

3. Orden de Prioridad:

- Mostrar sin responder primero
- Luego más recientes
- Resaltar reseñas negativas (☆☆☆☆)

4. Guía de Tono:

- Tooltip con mejores prácticas
- Agradecer siempre
- Ser profesional pero amigable
- No ser defensivo con críticas
- Ofrecer soluciones

5. Responsive:

- Móvil (ScanYA): Cards full-width, UI simplificada
- Laptop (BS): 2 columnas
- Desktop (BS): 3 columnas + sidebar stats

6. Permisos Granulares:

- Dueño: Puede TODO (responder, editar cualquier respuesta, reportar, configurar permisos)
- Gerente: Responder, editar sus respuestas, reportar
- Empleado: Responder (si tiene permiso), solo editar sus respuestas

Dependencias Técnicas:

Requiere OBLIGATORIAMENTE:

- Tabla **transacciones** (Fase 5.5 ScanYA)

- Validación de compras del cliente
- Sistema de notificaciones push
- ScanYA PWA funcional con módulo de Opiniones
- Sistema de permisos por empleado

Flujo Completo:

```

    Cliente compra en negocio (ScanYA)
    ↓
    Sistema registra transacción
    ↓
    Cliente puede dejar reseña (24h después)
    ↓
    Reseña aparece en BS del negocio + ScanYA
    ↓
    Notificación a dueño/gerente/empleados autorizados
    ↓
    Propietario responde desde BS (web)
    O
    Empleado responde desde ScanYA (móvil)
    ↓
    Sistema registra quién respondió y desde dónde
    ↓
    Cliente recibe notificación de respuesta
    ↓
    Reseña + respuesta visible en perfil público
  
```

Restricciones:

- Sin compras = No puede reseñar
- Solo 1 reseña cada 30 días por cliente
- Negocio puede responder ilimitado
- Dueño/Gerente puede responder desde BS o ScanYA
- Empleado solo puede responder desde ScanYA (si tiene permiso)
- Cada usuario solo puede editar sus propias respuestas
- Cliente puede editar su reseña (7 días)
- Negocio NO puede eliminar reseñas (solo reportar)
- Solo dueño/gerente pueden reportar reseñas

Configuración de Permisos:

Desde Business Studio → Empleados → Editar empleado:

Permisos: Juan Pérez

- | | |
|-------------------------------------|------------------|
| <input checked="" type="checkbox"/> | Registrar ventas |
| <input checked="" type="checkbox"/> | Canjear cupones |
| <input checked="" type="checkbox"/> | Ver clientes |

- | |
|---|
| <input checked="" type="checkbox"/> Responder opiniones |
| <input type="checkbox"/> Editar respuestas de otros |
| <input type="checkbox"/> Reportar reseñas |
| <input type="checkbox"/> Ver estadísticas completas |

5. Alertas ([/business-studio/alertas](#)) PENDIENTE

Centro de Notificaciones:

-  ALERTAS Y NOTIFICACIONES
[Todas] [No leídas (4)] [Importantes]

 3 cupones vencen mañana
Envía recordatorio a los clientes
hace 1 hora • [Ver cupones] [Marcar como leída]

 Nueva aplicación a "Mesero/a"
Juan Pérez aplicó a tu vacante
hace 2 horas • [Ver aplicación]

 Cliente VIP cumple años en 2 días
María González - Cliente Oro
hace 3 horas • [Enviar cupón especial]

 Nuevo mensaje de cliente
Tienes 2 mensajes sin leer en ChatYA
hace 5 horas • [Ver mensajes]

Tipos de Alertas:

- Cupones por vencer
- Nuevas aplicaciones a empleos
- Cumpleaños de clientes
- Nueva reseña
- Cliente inactivo (>30 días)
- Rifas próximas a cerrar
- Nuevos mensajes (ChatYA)

6. Catálogo ([/business-studio/catalogo](#)) COMPLETADO

Gestión de Productos y Servicios:

Vista Principal:

Producto	Categoría	Precio
Orden de 3 Tacos	Bebidas	\$150
Corte de Pelo	Cena	\$200

Modal de Edición:

Editar Artículo [X]

[IMAGEN] [X]	Nombre * [Orden de 3 Tacos_____]
Descripción (opcional) [Asada, Adobada, Cabeza...]	
Categoría [Bebidas ▼]	Precio * [\$ 150]
<input type="checkbox"/> Mostrar como "Desde \$150"	
<input checked="" type="checkbox"/> Visible <input checked="" type="checkbox"/> Destacado	
<input type="button" value="Cancelar"/>	<input type="button" value="Guardar"/>

Funcionalidades Implementadas:

- ✓ Crear artículo unificado (Producto/Servicio)
- ✓ Editar artículo
- ✓ Eliminar artículo
- ✓ Duplicar artículo (copiar)

- Subir imagen única (optimista a Cloudinary)
- Eliminar imagen con botón X rojo
- Categorizar por dropdown
- Definir precio
- Toggle visible/oculto (👁️ verde)
- Toggle destacado (⭐ amarillo)
- Checkbox "Mostrar como 'Desde \$X'" para precios variables
- Búsqueda por nombre
- Filtros por categoría (tabs superiores)
- Métricas por artículo (👁️ vistas, 🛒 carrito/ventas)
- Grid responsive de tarjetas
- Badge tipo (Producto/Servicio) con íconos distintivos
- UI optimista en todos los cambios

Pendiente:

- Importar/Exportar CSV
- Galería de imágenes (actualmente sólo 1 imagen)

Campos por Artículo (Unificados):

- Nombre (requerido)
- Descripción (opcional)
- Precio (requerido)
- Imagen única (opcional)
- Categoría (requerido, dropdown)
- Tipo: Producto o Servicio (badge visual)
- Visible (toggle, default: true)
- Destacado (toggle, default: false)
- Mostrar como "Desde \$X" (checkbox, default: false)

Notas Técnicas:

- La distinción Producto/Servicio es visual únicamente (badge y emoji)
- No hay separación de formularios entre productos y servicios
- Las métricas (vistas/carrito) se actualizan en tiempo real
- La búsqueda filtra por nombre en tiempo real
- Los filtros de categoría son tabs que filtran instantáneamente
- Todas las operaciones son optimistas con rollback automático en caso de fallo

7. Ofertas ([/business-studio/ofertas](#)) COMPLETADO

Gestión de Ofertas Permanentes:

Vista Principal:

[Total: 8] [Activas: 4] [Inactivas: 2]

[📅 Próximas: 0] [⌚ Vencidas: 2]

Buscar por título... [+ Nueva Oferta]

[% Porcentaje] [\$ Monto fijo] [🛍️ 2×1] [🛍️ 3×2] [🚚 Envío gratis]
[⭐ Otro]

Inactiva 2×1

Tacos
 12 Ene - 18 Ene
👁️ 0 ❤️ 0 ⚒ 0

HAPPY HOUR Activa 3×2

REVISAR 17 ENERO
 14 Ene - 16 Ene
👁️ 0 ❤️ 0 ⚒ 0

Activa 25% DESCUENTO

REVISAR SI CAMBIO A VENCIDA
 14 Ene - 7 Feb
👁️ 0 ❤️ 0 ⚒ 0

Activa \$ \$100

DESAYUNOS
 10 Ene - 6 Mar
👁️ 0 ❤️ 0 ⚒ 0

Modal Nueva Oferta:

Nueva Oferta Activa [

IMAGEN DE LA OFERTA
Click subir
[+]

Título *
[Ej: 50% de descuento en pizzas_____]

Descripción (opcional)
[Términos y condiciones..._____]

Fecha inicio * Fecha fin *

<p>[14/01/2026] [15/01/2026]</p> <p>Tipo de oferta *</p> <p>[2x1] [3x2] [Envío] [Desc. %] [Monto \$] [Otro]</p> <p>Valor *</p> <p>[% 10_____]</p> <p>Compra mínima (opcional)</p> <p>[\\$ 0_____]</p> <p style="text-align: center;">[Cancelar] [Crear]</p>	
--	--

Modal Editar Oferta (con tipo "Otro"):

<p>Editar Oferta <input checked="" type="checkbox"/> Activa [X]</p> <p>Título * [IMAGEN] [X]</p> <p>Descripción (opcional) [REVISAR SI MANTIENE EN ACTIVA_____]</p> <p>Tipo de oferta *</p> <p>[2x1] [3x2] [Envío] [Desc. %] [Monto \$] [Otro]</p> <p>Concepto * [Happy Hour_____]</p> <p>Fecha inicio * [14/01/2026] Fecha fin * [15/01/2026]</p> <p>Compra mínima (opcional) [\\$ 0.00_____]</p> <p style="text-align: center;">[Cancelar] [Guardar]</p>	
---	--

Funcionalidades Implementadas:

- Dashboard con 5 contadores de estado:
 - Total de ofertas (azul)
 - Activas (verde)
 - Inactivas (rojo)
 - Próximas (amarillo)
 - Vencidas (gris)
- Crear oferta

- Editar oferta
- Eliminar oferta
- Duplicar oferta (copiar)
- Toggle visible/oculto por oferta (⌚)
- Toggle activa/inactiva global (esquina superior derecha)
- Búsqueda por título
- Filtros por tipo de oferta (6 tabs):
 - % Porcentaje
 - \$ Monto fijo
 -  2×1
 -  3×2
 -  Envío gratis
 -  Otro
- Subir imagen (optimista a Cloudinary)
- Eliminar imagen con botón X rojo
- Métricas por oferta (⌚ vistas, ❤ guardadas, 📈 conversiones)
- Grid responsivo de tarjetas
- Badges de estado (Activa/Inactiva/Vencida)
- Badges de tipo de oferta en esquina superior derecha
- UI optimista en todos los cambios

Tipos de Oferta Soportados:

1. **2x1** - Paga 1, lleva 2
2. **3x2** - Paga 2, lleva 3
3. **Envío gratis** - Sin costo de envío
4. **Desc. %** - Descuento porcentual (requiere valor)
5. **Monto \$** - Descuento en pesos (requiere valor)
6. **Otro** - Oferta personalizada (requiere concepto)

Campos por Oferta:

- Título (requerido)
- Descripción (opcional - términos y condiciones)
- Imagen (opcional)
- Tipo de oferta (requerido - selección única)
- Valor (requerido solo para Desc. % y Monto \$)
- Concepto (requerido solo para tipo "Otro")
- Fecha inicio (requerido)
- Fecha fin (requerido)
- Compra mínima (opcional, default: \$0)
- Activa (toggle, default: true)

Estados de Oferta:

- **Activa** (verde): Dentro del rango de fechas y toggle activo
- **Inactiva** (rojo): Toggle desactivado manualmente
- **Próxima** (amarillo): Fecha inicio futura

- **Vencida** (gris): Fecha fin pasada

Pendiente:

- Ver lista de usuarios que guardaron la oferta
- Exportar estadísticas de ofertas
- Días/horarios específicos aplicables (actualmente solo rango de fechas)
- Aplicar oferta a productos/categorías específicas
- Límite de canjes por usuario
- Stock limitado de la oferta

Notas Técnicas:

- Las ofertas se filtran automáticamente por sucursal (interceptor Axios)
- El cambio de estado es optimista con rollback en caso de fallo
- Las imágenes son opcionales (se muestra ícono por defecto según tipo)
- El sistema calcula automáticamente el estado según fechas y toggle
- La búsqueda filtra en tiempo real por título
- Los filtros de tipo son excluyentes (una oferta = un tipo)
- Las métricas se actualizan cuando usuarios interactúan con ofertas públicas

8. Cupones ([/business-studio/cupones](#)) PENDIENTE (Requiere 5.6 Cupones)

Gestión de Cupones:

The screenshot shows a user interface for managing coupons. At the top, there's a header with a 'CUPONES' button and three links: '[+ Crear Cupón]', '[Cupones activos]', and '[Cupones expirados]'. Below this is a detailed view of a specific coupon card. The card displays the following information:
- Title: 30% Descuento - Navidad
- Validity: Válido: 20 Dic - 25 Dic 2024
- Metrics: Enviados: 45 • Canjeados: 12 • Tasa: 26.7%
- Actions: [Editar], [Eliminar], [Enviar a más clientes]

Crear Cupón:

- Título del cupón
- Descripción/condiciones
- Tipo de descuento (% o \$)
- Valor del descuento
- Fecha de vencimiento
- Límite de usos por cliente
- Imagen del cupón (opcional)
- Enviar a:
 - Todos los clientes

- Por nivel (Bronce/Plata/Oro)
- Clientes específicos
- Clientes inactivos

Funcionalidades:

- Crear cupón
- Editar cupón activo
- Eliminar cupón
- Ver estadísticas de canje
- Lista de quien canjeó
- Reenviar cupón
- Duplicar cupón

9. Puntos ([/business-studio/puntos](#)) PENDIENTE (Requiere 5.5 CardYA)

Configuración del Sistema de Lealtad:

 CONFIGURACIÓN DE PUNTOS

¿Participas en el sistema de puntos?

Sí No

 VALOR DE TUS PUNTOS:

1 punto = \$[____] pesos

Ejemplo: Si configuras \$10, tus clientes acumulan:

- Bronce: 1 punto por cada \$10 de compra
- Plata: 1.25 puntos por cada \$10 (+25%)
- Oro: 1.5 puntos por cada \$10 (+50%)

 Estadísticas de Puntos:

- Total puntos otorgados este mes: 4,580
- Promedio por transacción: 45 pts
- Valor equivalente: \$45,800 en compras
- Top 10 clientes por puntos acumulados

[Guardar configuración]

Configuración Detallada:

A) Valor del Punto:

Configura cuánto vale 1 punto en tu negocio

1 punto = \$[____] pesos

Ejemplos comunes:

- \$5 → Cliente acumula 2 puntos por cada \$10
- \$10 → Cliente acumula 1 punto por cada \$10 (recomendado)
- \$20 → Cliente acumula 1 punto por cada \$20
- \$100 → Cliente acumula 1 punto por cada \$100

 Una vez configurado, se aplica a todas las compras

[Guardar]

B) Simulador de Acumulación:

SIMULADOR - Con tu configuración actual (1 punto = \$10)

Si un cliente compra \$500:

-  Bronce (x1.0): 50 puntos
-  Plata (x1.25): 62 puntos (+25%)
-  Oro (x1.5): 75 puntos (+50%)

Cambiar monto de prueba: \$[____] [Calcular]

Funcionalidades:

- Configurar valor del punto (1 punto = \$X pesos)
- Activar/desactivar sistema de puntos
- Ver simulador de acumulación
- Ver estadísticas de puntos otorgados
- Ver valor equivalente en pesos
- Top clientes por puntos
- Historial de puntos otorgados
- Exportar reporte de puntos

Cálculo Automático:

```
// Backend - Cálculo de puntos al registrar venta
function calcularPuntos(
  montoCompra: number,
  valorPunto: number, // Lo configura el negocio
  nivelCliente: 'bronze' | 'plata' | 'oro'
): number {
  // Base: 1 punto por cada $X configurados
  const puntosBase = montoCompra / valorPunto;
```

```

// Multiplicador según nivel
const multiplicador = {
  bronce: 1.0,
  plata: 1.25,
  oro: 1.5
}[nivelCliente];

return Math.floor(puntosBase * multiplicador);
}

// Ejemplo:
// Compra: $500
// Valor punto: $10 (configurado por negocio)
// Cliente: Oro
// Resultado: (500 / 10) * 1.5 = 75 puntos

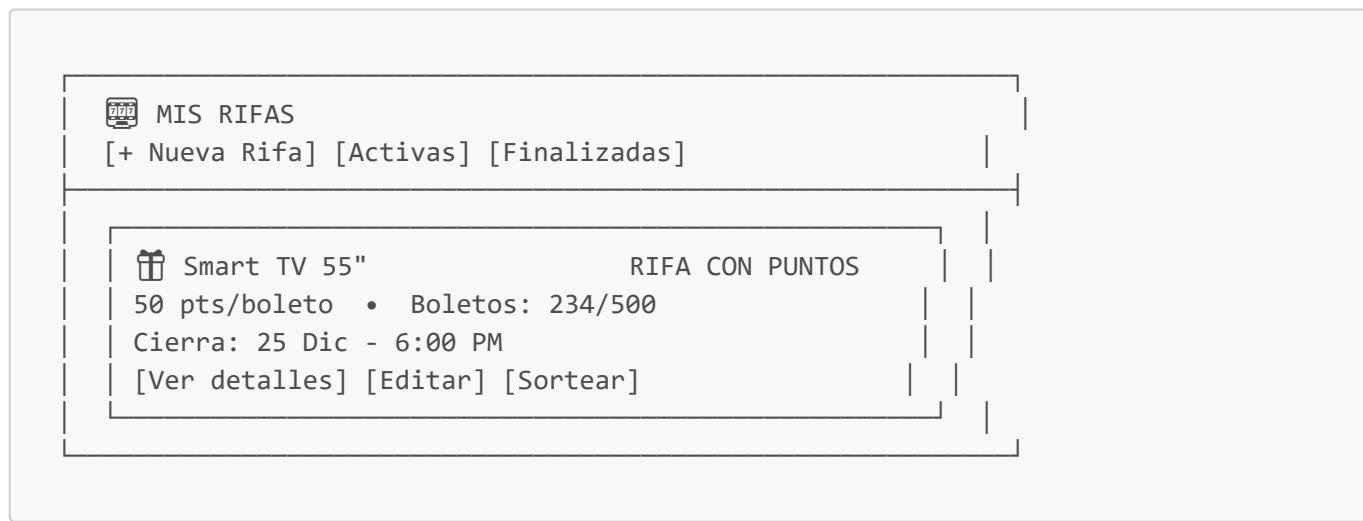
```

Restricciones:

- Valor mínimo: \$1 peso por punto
- Valor máximo: \$1,000 pesos por punto
- Una vez configurado, aplica a todas las transacciones
- Se puede cambiar en cualquier momento
- El cambio NO afecta puntos ya otorgados

10. Rifas ([/business-studio/rifas](#)) PENDIENTE (Requiere 5.10 Dinámicas)

Gestión de Dinámicas:



The screenshot shows the 'RIFAS' section in the Business Studio. At the top, there's a header with a gift icon, 'MIS RIFAS', and buttons for '[+ Nueva Rifa]', '[Activas]', and '[Finalizadas]'. Below this, a specific raffle is detailed: it's for a 'Smart TV 55"', priced at '50 pts/boleto', with 'Boletos: 234/500'. It's set to 'Cierra: 25 Dic - 6:00 PM' and includes links for '[Ver detalles]', '[Editar]', and '[Sortear]'.

Crear Rifa:

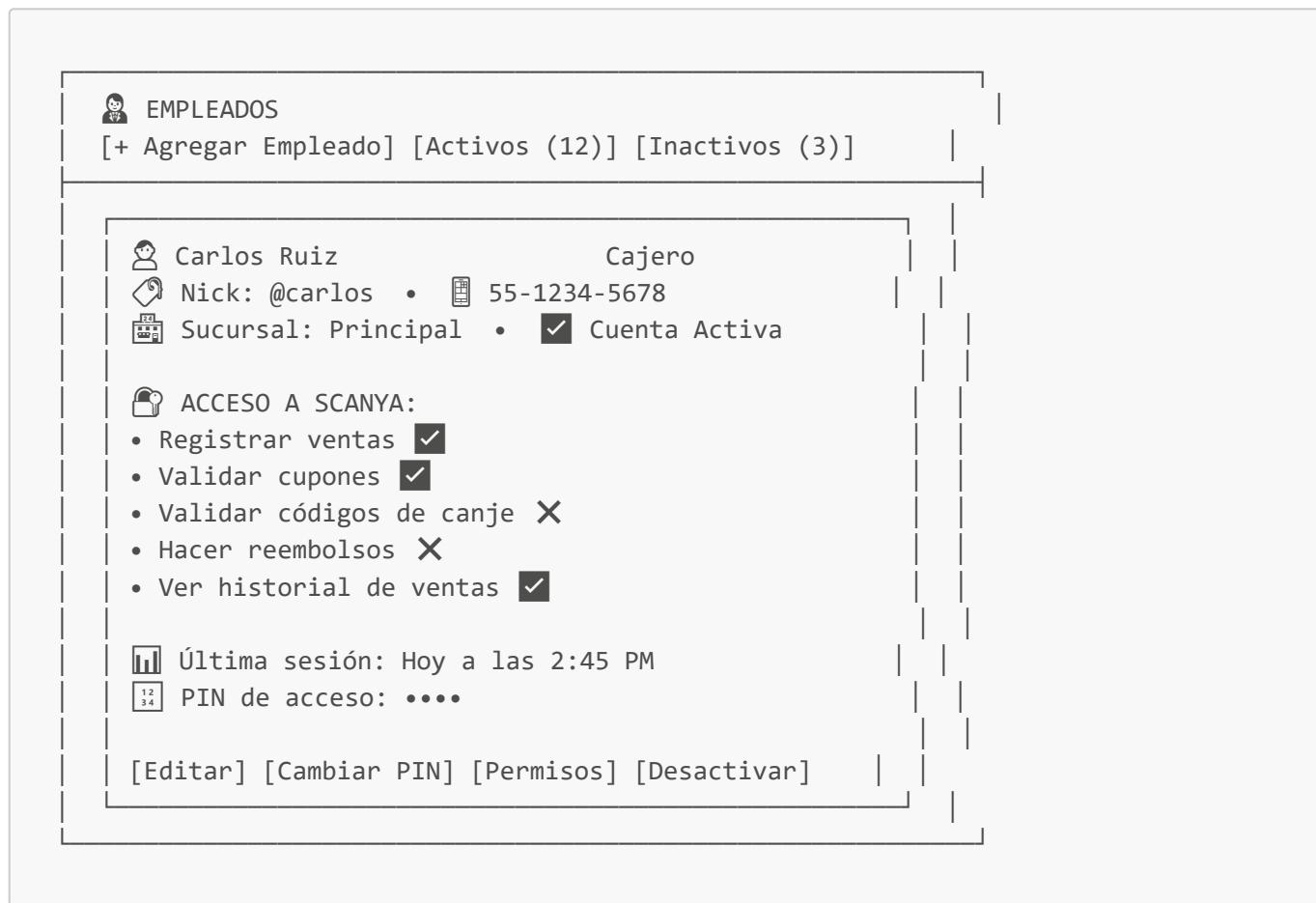
- Tipo (con puntos, con dinero, offline)
- Premio (descripción e imagen)
- Costo por boleto
- Total de boletos
- Fecha de cierre
- (Offline) Registro manual de participantes

Funcionalidades:

- Ver lista de participantes
 - Ver boletos vendidos
 - Realizar sorteo
 - Ver ganador
 - Marcar premio como entregado
 - Compartir link público (offline)
 - Historial de rifas
-

11. Empleados ([/business-studio/empleados](#))  PENDIENTE (Requiere 6.1 ScanYA)

Gestión de Personal con Acceso a ScanYA:



The screenshot shows a user profile for 'Carlos Ruiz' (Cajero). It includes contact information (Nick: @carlos, Phone: 55-1234-5678) and location (Sucursal: Principal, Cuenta Activa checked). Under 'ACCESO A SCANYA', it lists permissions: Registrar ventas (checked), Validar cupones (checked), Validar códigos de canje (X), Hacer reembolsos (X), and Ver historial de ventas (checked). Below this, it shows the last session (Hoy a las 2:45 PM) and PIN de acceso (*****). At the bottom are buttons for Editar, Cambiar PIN, Permisos, and Desactivar.

Sistema de Acceso para Empleados

 **IMPORTANTE:** Los empleados SOLO tienen acceso a la app ScanYA, NO al panel completo de Business Studio.

Características:

- Acceso exclusivo a ScanYA (app móvil/web)
- Login con **nick + PIN** (4-6 dígitos)
- Permisos configurables dentro de ScanYA
- Registro de actividad completo
- Restricción por sucursal (opcional)

Agregar Empleado - Formulario

CREAR CUENTA DE EMPLEADO (ACCESO A SCANYA)	
DATOS PERSONALES:	
Nombre completo: [_____]	
Teléfono: [_____]	
Puesto/Rol: [_____]	
ASIGNACIÓN:	
Sucursal: [▼ Principal]	
<input type="checkbox"/> Puede trabajar en todas las sucursales	
CREDENCIALES DE ACCESO:	
Nick de usuario: [_____]	
└ Ejemplo: carlos, juan123, ana_m	
└ Solo letras, números y guión bajo	
└ Mínimo 3 caracteres, máximo 20	
PIN de acceso (4-6 dígitos): [_____]	
Confirmar PIN: [_____]	
<p>⚠ El empleado usará su NICK y PIN para acceder únicamente a la app ScanYA</p>	
[Cancelar] [Crear cuenta y asignar permisos →]	

Permisos Dentro de ScanYA

PERMISOS EN SCANYA - Carlos Ruiz	
🕒 OPERACIONES DE VENTA:	
<input checked="" type="checkbox"/> Escanear QR de clientes	
<input checked="" type="checkbox"/> Registrar ventas	
<input checked="" type="checkbox"/> Aplicar multiplicador de puntos (automático)	
<input checked="" type="checkbox"/> Ver información del cliente	
🎫 CUPONES Y CANJES:	
<input checked="" type="checkbox"/> Validar cupones	
<input type="checkbox"/> Validar códigos de canje de puntos	
💰 AJUSTES Y CORRECCIONES:	
<input type="checkbox"/> Hacer reembolsos	
<input type="checkbox"/> Cancelar transacciones	
<input type="checkbox"/> Modificar montos	
📊 CONSULTAS:	

- Ver su historial de ventas del día
- Ver historial completo de ventas
- Ver estadísticas del negocio

[Cancelar] [Guardar permisos]

Flujo de Acceso del Empleado

1. Empleado descarga/abre app ScanYA
 - └ Pantalla de login exclusiva para empleados
2. Ingresa credenciales:
 - └ Nick: carlos
 - └ PIN: 1234
3. Sistema valida:
 - └ Nick existe ✓
 - └ PIN correcto ✓
 - └ Cuenta activa ✓
 - └ Es empleado (NO dueño) ✓
 - └ Permisos asignados ✓
 - └ Sucursal asignada ✓
4. Acceso concedido a ScanYA:
 - └ Interfaz limitada según permisos
 - └ NO tiene acceso a Business Studio
 - └ Solo puede usar funciones permitidas
5. Interfaz de ScanYA para empleado:

SCANYA
@carlos - Carlos Ruiz
Sucursal: Principal
[Cerrar sesión]

[Escanear QR del cliente]

Última venta: \$325.00
Juan Pérez - Nivel ORO
+48 puntos otorgados

Mi resumen de hoy:

- Ventas: 12
- Total: \$2,450
- Puntos otorgados: 245

[Ver mi historial]

Diferencias: Dueño vs Empleado en ScanYA

Dueño del Negocio:

- Acceso completo a Business Studio
- Acceso completo a ScanYA
- Todos los permisos
- Ver todas las transacciones
- Modificar configuraciones

Empleado:

- NO tiene acceso a Business Studio
- Solo acceso a ScanYA
- Permisos limitados configurables
- Solo ve sus propias transacciones (opcional)
- NO puede modificar configuraciones

Registro de Actividad

ACTIVIDAD - Carlos Ruiz (Empleado)
[Hoy] [Semana] [Mes]

🕒 Hoy - 22 Dic 2024

2:45 PM • Registró venta de \$325.00 (Cliente: Juan Pérez)
2:30 PM • Validó cupón "30% descuento"
1:15 PM • Registró venta de \$150.00 (Cliente: Ana López)
10:00 AM • Inició sesión en ScanYA

📊 RESUMEN DEL DÍA:
• Ventas registradas: 12
• Total en ventas: \$2,450
• Clientes atendidos: 12
• Puntos otorgados: 245
• Cupones validados: 3
• Tiempo de sesión: 4h 45m

[Exportar actividad] (solo dueño)

Funcionalidades del Módulo

Para el Comerciante (dueño):

- Crear cuenta de empleado (solo acceso a ScanYA)
- Asignar nick único y PIN inicial
- Configurar permisos dentro de ScanYA

- Asignar sucursal específica o todas
- Cambiar PIN del empleado
- Cambiar nick del empleado
- Ver actividad del empleado (log completo)
- Activar/desactivar cuenta temporalmente
- Eliminar cuenta de empleado
- Ver resumen de ventas por empleado
- Exportar reporte de actividad

Para el Empleado (en ScanYA):

- Iniciar sesión con nick + PIN
- Escanear QR de clientes
- Registrar ventas (según permisos)
- Validar cupones (según permisos)
- Ver su propio historial del día
- Cambiar su PIN (opcional, si tiene permiso)
- Cerrar sesión

Permisos Predefinidos (Templates)

 TEMPLATE: Cajero Básico	
└ Escanear QR	✓
└ Registrar ventas	✓
└ Validar cupones	✓
└ Ver historial propio	✓
└ Todo lo demás	✗
 TEMPLATE: Cajero Avanzado	
└ Escanear QR	✓
└ Registrar ventas	✓
└ Validar cupones	✓
└ Validar canjes	✓
└ Ver historial propio	✓
└ Reembolsos	✓
└ Ver estadísticas	✗
 TEMPLATE: Supervisor	
└ Todas las operaciones	✓
└ Reembolsos	✓
└ Cancelaciones	✓
└ Ver historial completo	✓
└ Ver estadísticas	✓

Modelo de Datos

```
interface Empleado {  
  id: string;
```

```
negocioId: string;
nombre: string;
nick: string; // Único por negocio
telefono?: string;
puesto: string;
pin: string; // Hasheado con bcrypt

// Asignación
sucursalId?: string; // null = todas las sucursales
todasLasSucursales: boolean;

// Acceso
tipoAcceso: 'scanya_only'; // Solo ScanYA, NO Business Studio

// Permisos dentro de ScanYA
permisosScanYA: {
    operaciones: {
        escanearQR: boolean;
        registrarVentas: boolean;
        verInfoCliente: boolean;
    };
    cupones: {
        validarCupones: boolean;
        validarCanjes: boolean;
    };
    ajustes: {
        reembolsos: boolean;
        cancelaciones: boolean;
        modificarMontos: boolean;
    };
    consultas: {
        verHistorialPropio: boolean;
        verHistorialCompleto: boolean;
        verEstadisticas: boolean;
    };
};

// Estado
activo: boolean;
createdAt: Date;
ultimaSesion?: Date;
}

interface ActividadEmpleado {
    id: string;
    empleadoId: string;
    tipo: 'login' | 'logout' | 'venta' | 'cupon' | 'reembolso';
    descripcion: string;
    metadata?: {
        ventaId?: string;
        monto?: number;
        clienteId?: string;
    };
}
```

```
    timestamp: Date;  
}
```

Validaciones de Nick

```
// Reglas para el nick  
const nickRegex = /^[a-zA-Z0-9_]{3,20}$/;  
  
function validarNick(nick: string): boolean {  
    // Solo letras, números y guión bajo  
    // Mínimo 3 caracteres, máximo 20  
    return nickRegex.test(nick);  
}  
  
// Backend - Verificar unicidad por negocio  
async function nickDisponible(  
    nick: string,  
    negocioId: string  
): Promise<boolean> {  
    const existe = await db.empleados.findOne({  
        nick: nick.toLowerCase(),  
        negocioId  
    });  
  
    return !existe;  
}  
  
// Login de empleado  
async function loginEmpleado(  
    nick: string,  
    pin: string,  
    negocioId: string  
): Promise<Empleado | null> {  
    const empleado = await db.empleados.findOne({  
        nick: nick.toLowerCase(),  
        negocioId,  
        activo: true  
    });  
  
    if (!empleado) return null;  
  
    const pinValido = await bcrypt.compare(pin, empleado.pin);  
  
    if (!pinValido) return null;  
  
    // Actualizar última sesión  
    await db.empleados.updateOne(  
        { id: empleado.id },  
        { ultimaSesion: new Date() }  
    );
```

```
    return empleado;
}
```

Validaciones de Seguridad

```
// Backend - Validar que es empleado y tiene permiso
async function validarAccesoScanYA(
  empleadoId: string,
  accion: string
): Promise<boolean> {
  const empleado = await getEmpleado(empleadoId);

  // Verificar cuenta activa
  if (!empleado.activo) return false;

  // Verificar que es empleado (no dueño)
  if (empleado.tipoAcceso !== 'scanya_only') return false;

  // Verificar permiso específico en ScanYA
  const tienePermiso = empleado.permisosScanYA[categoría]?.[accion];

  // Registrar intento
  await logAcceso(empleadoId, 'scanya', accion, tienePermiso);

  return tienePermiso;
}

// Bloquear acceso a Business Studio
async function validarAccesoBusinessStudio(
  usuarioId: string
): Promise<boolean> {
  const usuario = await getUsuario(usuarioId);

  // Si es empleado, DENEGAR acceso
  if (usuario.tipoAcceso === 'scanya_only') {
    return false;
  }

  // Si es dueño, PERMITIR
  return true;
}
```

Pantalla de Login de ScanYA



 ACCESO PARA EMPLEADOS

Nick de usuario:

[_____]

PIN:

[• • • •]

[Iniciar sesión]

 Solo personal autorizado

¿Eres dueño del negocio?

[Ir a Business Studio]

12. Vacantes ([/business-studio/vacantes](#))  PENDIENTE (Requiere 5.11 Empleos)**Gestión de Ofertas de Empleo:** OFERTAS DE EMPLEO

[+ Publicar Vacante] [Activas] [Cerradas]

Mesero/a - Tiempo Completo

Publicada: hace 5 días • 12 aplicaciones

[Ver aplicaciones] [Editar] [Cerrar vacante]

Publicar Vacante:

- Título del puesto
- Descripción
- Tipo de jornada
- Rango salarial
- Horario
- Requisitos
- Beneficios
- Sucursal (si aplica)

Gestión de Aplicaciones:

- Ver lista de aplicantes
 - Ver CV/perfil del aplicante
 - Marcar estado (vista, proceso, rechazada, aceptada)
 - Contactar por ChatYA
 - Filtrar aplicaciones por estado
 - Cerrar vacante
-

13. Reportes ([/business-studio/reportes](#)) PENDIENTE (Requiere 6.1 ScanYA)

Generación de Reportes:

REPORTES Y EXPORTACIONES

Selecciona el tipo de reporte:

- Ventas
 - └ Rango: [Del] [____] [A1] [____]
 - [Exportar PDF]
 - [Exportar CSV]
- Clientes
 - └ [Exportar lista completa CSV]
- Cupones
 - └ [Exportar estadísticas PDF]
- Puntos
 - └ [Exportar reporte de puntos CSV]
- Empleos
 - └ [Exportar aplicaciones CSV]

Reportes Disponibles:

- Ventas por período
- Ventas por producto/servicio
- Ventas por sucursal
- Base de clientes completa
- Clientes por nivel
- Cupones: creados vs canjeados
- Puntos otorgados
- Aplicaciones a empleos
- Estadísticas de rifas
- Reseñas recibidas

Formatos de Exportación:

- PDF (reportes visuales)
 - CSV (datos tabulares)
 - Excel (análisis avanzado)
-

14. Sucursales ([/business-studio/sucursales](#)) PENDIENTE

Gestión de Múltiples Ubicaciones:

- Ver todas las sucursales
 - Agregar nueva sucursal
 - Editar sucursal existente
 - Eliminar sucursal
 - Configurar sucursal principal
 - Por cada sucursal:
 - Nombre
 - Dirección
 - Coordenadas (mapa Leaflet)
 - Teléfono específico
 - Horarios específicos
 - Estado (activa/inactiva)
 - Empleados asignados
-

15. Mi Perfil ([/business-studio/perfil](#)) COMPLETADO

6 Tabs implementados:

A) Datos del Negocio (TabInformacion):

- Nombre del Negocio *
- Descripción (opcional)
- Categoría y Subcategorías (máx 3) *
- Panel lateral CardYA:
 - Toggle activar/desactivar sistema de lealtad
 - Beneficios (Clientes Recurrentes, Mayor Ticket, Sin Costo)
 - ¿Cómo Funciona? (4 pasos)
 - Botón "Ir a Puntos →"

B) Contacto (TabContacto):

- Teléfono con lada (+52)
- WhatsApp con lada (+52)
- Correo Electrónico
- Sitio Web (opcional)
- Redes Sociales (Facebook, Instagram, TikTok, Twitter/X)

C) Ubicación (TabUbicacion):

- Calle y Colonia *

- Ciudad *
- Botón "Usar mi ubicación" (GPS)
- Mapa Leaflet interactivo con marcador arrastrable
- Tip explicativo de uso

D) Horarios (TabHorarios):

- Selector de días (Lun-Dom) con indicadores
- Botón "Duplicar Horario"
- Botón "Abierto 24/7"
- Toggle Estado del día (Abierto/Cerrado)
- Hora de Apertura y Cierre (HH:MM AM/PM)
- Checkbox "¿Tienes horario de comida/break?"
- Hora de Salida y Regreso (break)

E) Imágenes (TabImagenes):

- Logo del Negocio (PNG/JPG máx 2MB)
- Foto de Perfil - ChatYA/Avatar (PNG/JPG máx 2MB)
- Imagen de Portada (1600×900px)
- Galería de Fotos (máx 10, 1200×1200px recomendado)
- Contador de imágenes (ej: 8/10)
- Upload optimista a Cloudinary
- Optimización automática a .webp

F) Operación (TabOperacion):

- Métodos de Pago * (mínimo 1 requerido):
 - Efectivo, Tarjeta, Transferencia
- Opciones de Entrega:
 - Envío a Domicilio (envías productos)
 - Servicio a Domicilio (vas al cliente)
- Nota explicativa para usuarios

⌚ Componentes de Business Studio

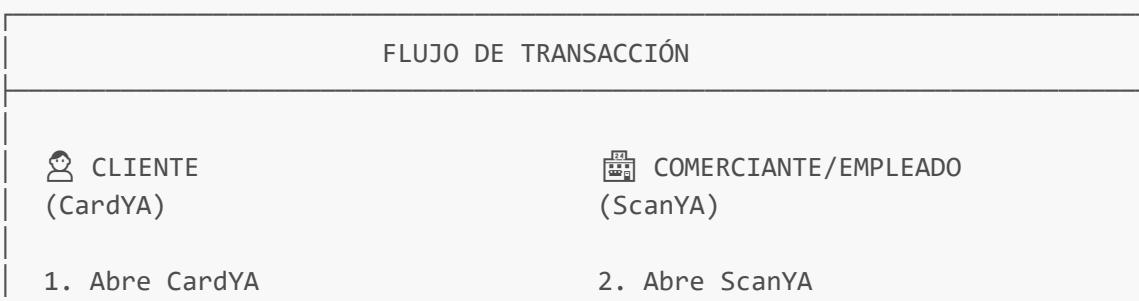
pages/private/business-studio/		
└── dashboard/		
└── PaginaDashboard.tsx	✓	
└── componentes/	✓	(10 componentes)
└── perfil/		
└── PaginaPerfil.tsx	✓	
└── hooks/usePerfil.ts	✓	
└── components/	✓	(8 componentes)
└── catalogo/		
└── PaginaCatalogo.tsx	✗	
└── FormProducto.tsx	✗	
└── FormServicio.tsx	✗	
└── ofertas/		

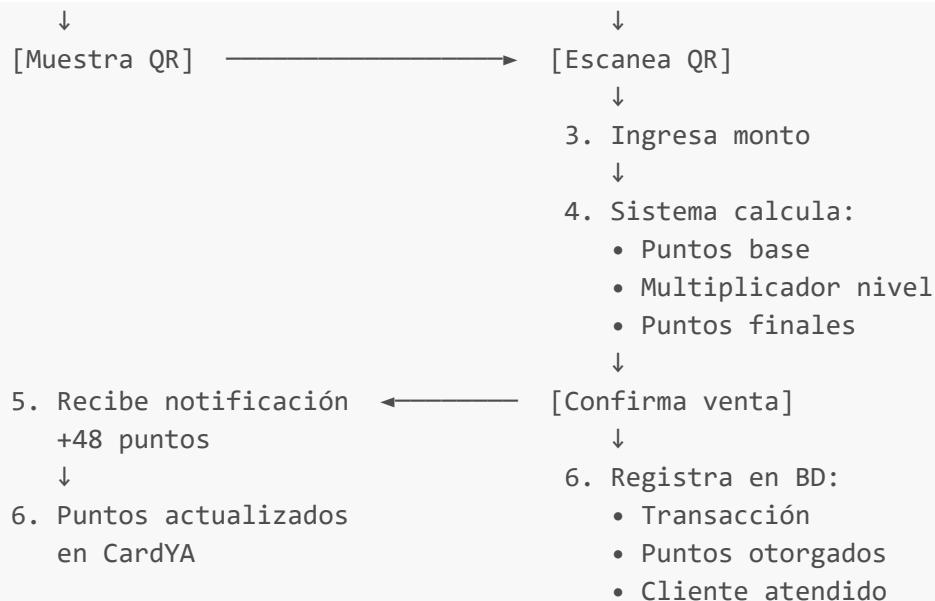
PaginaOfertas.tsx	⌚
FormOferta.tsx	⌚
clientes/	
PáginaClientes.tsx	⌚
PerfilCliente.tsx	⌚
HistorialCliente.tsx	⌚
transacciones/	
PáginaTransacciones.tsx	⌚
DetalleTransaccion.tsx	⌚
cupones/	
PáginaCupones.tsx	⌚
FormCupon.tsx	⌚
puntos/	
PáginaPuntos.tsx	⌚
ConfiguracionPuntos.tsx	⌚
empleados/	
PáginaEmpleados.tsx	⌚
FormEmpleado.tsx	⌚
PermisosEmpleado.tsx	⌚
vacantes/	
PáginaVacantes.tsx	⌚
FormVacante.tsx	⌚
ListaAplicaciones.tsx	⌚
reportes/	
PáginaReportes.tsx	⌚
rifas/	
PáginaRifas.tsx	⌚
FormRifa.tsx	⌚
alertas/	
PáginaAlertas.tsx	⌚
sucursales/	
PáginaSucursales.tsx	⌚
FormSucursal.tsx	⌚

⌚ Sistema de Lealtad: CardYA + ScanYA

CardYA (usuario) y **ScanYA** (comerciante) son dos caras de la misma moneda. Juntos forman el sistema de lealtad de AnunciaYA.

Flujo General del Sistema





5.5 ScanYA - Punto de Venta Digital PENDIENTE

Aplicación para comerciantes y empleados que registra ventas, otorga puntos y valida cupones.

Elemento	Descripción
Función	Punto de venta para registrar compras y otorgar puntos
Quién usa	Dueños (modo Comercial) + Empleados (con permisos)
Ruta App	/scanya
Ruta Widget	/scanya-widget (PWA standalone)
Requiere	Modo Comercial o cuenta de empleado

Tipos de Acceso

Tipo	Login	Acceso BS	Permisos
Dueño	Email + Contraseña	<input checked="" type="checkbox"/> Completo	Todos
Gerente Sucursal	Email + Contraseña	<input checked="" type="checkbox"/> Su sucursal	Configurables
Empleado	Nick + PIN	<input type="checkbox"/> Solo ScanYA	Configurables

Interfaz Principal





@carlos (Cajero)

[Cerrar sesión]

ESCANEAR QR

Toca para abrir
la cámara

[Ingresar código manual]

MI RESUMEN DE HOY

Ventas: 12 Total: \$4,580

Puntos otorgados: 458

Cupones validados: 3

[Ver historial completo]

ACCIONES RÁPIDAS

[Validar Cupón] [Validar Canje] [Historial]

Flujo de Venta Completo

PASO 1: ESCANEAR QR DEL CLIENTE

[Cámara activa]

Apunta al QR del cliente

QR válido por: 1:45



PASO 2: CLIENTE IDENTIFICADO

Juan Pérez

Nivel ORO - Multiplicador 1.5x

▀ En tu negocio:

- Puntos disponibles: 1,250
- Visitas totales: 23
- Última visita: hace 5 días

▀ Cupones disponibles: 2

- 20% descuento (vence mañana)
- 2x1 martes

[Continuar sin cupón] [Aplicar cupón]

**PASO 3: REGISTRAR VENTA****▀ Monto de la compra:**

\$ [500.00]

▀ Descripción (opcional):

2 pizzas grandes + refrescos

▀ Método de pago:

[Efectivo] [Tarjeta] [Transferencia]

▀ Foto de evidencia: (opcional)

[Tomar foto]

**PASO 4: CONFIRMAR PUNTOS****▀ RESUMEN DE LA TRANSACCIÓN**

Cliente: Juan Pérez

Monto: \$500.00

▀ CÁLCULO DE PUNTOS:

Valor del punto en tu negocio: \$10

Puntos base: $500 / 10 = 50$ puntos

Multiplicador ORO: x1.5



PUNTOS A OTORGAR: 75

[← Modificar]

[✓ Confirmar venta]

PASO 5: VENTA COMPLETADA 

¡Venta registrada!

Cliente: Juan Pérez

Monto: \$500.00

Puntos otorgados: +75

[...] Se envió notificación al cliente

[Nueva venta] [Ver detalles]

Validación de Cupones

 VALIDAR CUPÓN

Escanea el QR del cupón o ingresa el código:

[NAVIDAD-2024-ABC123]

[Escanear QR]

[Validar código]

 CUPÓN VÁLIDO

30% Descuento Navidad

Cliente: María González

Válido hasta: 25 Dic 2024

Condiciones: Compra mínima \$200

[Cancelar]

[✓ Aplicar cupón]

Validación de Canje de Puntos

 VALIDAR CANJE

El cliente generó un código de canje desde su CardYA.
Ingresá el código para validar:

[CANJE-7X9K2M]

[Validar]

↓

 CANJE VÁLIDO

 Cliente: Juan Pérez 

 Recompensa: Café gratis

 Puntos canjeados: 100

 Código válido por: 14:32 minutos

[Cancelar]

[✓ Confirmar entrega]

Acceso de Empleados

Login de Empleado:

 SCANYA

 ACCESO PARA EMPLEADOS

Nick de usuario:

carlos

PIN (4-6 dígitos):

• • • •

[Iniciar sesión]

Solo personal autorizado

¿Eres dueño del negocio?

[Ir a Business Studio]

Permisos Configurables (desde BS → Empleados):

Permiso	Descripción
escanearQR	Escanear QR de clientes
registrarVentas	Registrar ventas y otorgar puntos
verInfoCliente	Ver información del cliente
validarCupones	Validar cupones
validarCanjes	Validar códigos de canje
hacerReembolsos	Procesar devoluciones
cancelarTransacciones	Anular ventas
verHistorialPropio	Ver solo sus transacciones
verHistorialCompleto	Ver todas las transacciones
verEstadisticas	Ver métricas del negocio

Modelo de Datos

Tabla: transacciones

```
CREATE TABLE transacciones (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    negocio_id UUID REFERENCES negocios(id),
```

```

sucursal_id UUID REFERENCES negocio_sucursales(id),
cliente_id UUID REFERENCES usuarios(id),
empleado_id UUID REFERENCES empleados(id), -- null si fue el dueño

-- Datos de la venta
monto DECIMAL(10,2) NOT NULL,
descripcion TEXT,
metodo_pago VARCHAR(20), -- 'efectivo' | 'tarjeta' | 'transferencia'
foto_evidencia_url TEXT,

-- Puntos
puntos_base INTEGER,
multiplicador DECIMAL(3,2),
puntos_otorgados INTEGER,
nivel_cliente VARCHAR(10),

-- Cupón aplicado
cupon_id UUID REFERENCES cupones(id),
descuento_aplicado DECIMAL(10,2),

-- Estado
estado VARCHAR(20) DEFAULT 'completada', -- 'completada' | 'reembolsada' |
'anulada'

created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_transacciones_negocio ON transacciones(negocio_id);
CREATE INDEX idx_transacciones_sucursal ON transacciones(sucursal_id);
CREATE INDEX idx_transacciones_cliente ON transacciones(cliente_id);
CREATE INDEX idx_transacciones_fecha ON transacciones(created_at);

```

Tabla: empleados (ya documentada en módulo 9 de BS)

█ Componentes Frontend

```

pages/private/scanya/
├── PaginaScanYA.tsx           # Vista principal
└── PaginaLoginEmpleado.tsx    # Login nick + PIN
  └── components/
    ├── EscanerQR.tsx          # Cámara + lector QR
    ├── InfoCliente.tsx        # Card con datos del cliente
    ├── FormularioVenta.tsx    # Monto + descripción + método pago
    ├── ResumenPuntos.tsx      # Cálculo de puntos a otorgar
    ├── ConfirmacionVenta.tsx  # Pantalla de éxito
    ├── ValidarCupon.tsx       # Validación de cupones
    ├── ValidarCanje.tsx        # Validación de canjes
    ├── ResumenDia.tsx         # Métricas del día
    └── HistorialVentas.tsx    # Lista de transacciones

```

```

    └── AccionesRapidas.tsx      # Botones de acciones

hooks/
├── useScanYA.ts            # Estado general
├── useEscanerQR.ts         # Control de cámara
├── useTransaccion.ts       # Registro de ventas
└── useEmpleadoAuth.ts      # Auth de empleados

stores/
└── useScanYAStore.ts       # Estado global ScanYA

```

Endpoints Backend

```

// ===== TRANSACCIONES =====

// POST /api/scanya/venta
// Body: { clienteId, monto, descripcion, metodoPago, cuponId?, fotoEvidencia? }
// Retorna: transacción creada + puntos otorgados

// GET /api/scanya/historial
// Query: ?fecha=2024-01-06&empleadoId=xxx&page=1&limit=20
// Retorna: lista de transacciones

// POST /api/scanya/reembolso/:transaccionId
// Body: { motivo }
// Retorna: transacción actualizada

// ===== CUPONES =====

// POST /api/scanya/validar-cupon
// Body: { codigo }
// Retorna: info del cupón + cliente

// POST /api/scanya/aplicar-cupon
// Body: { cuponId, transaccionId }
// Retorna: cupón marcado como usado

// ===== CANJES =====

// POST /api/scanya/validar-canje
// Body: { codigo }
// Retorna: info del canje + recompensa

// POST /api/scanya/confirmar-canje
// Body: { canjeId }
// Retorna: canje confirmado

// ===== CLIENTES =====

// POST /api/scanya/identificar-cliente

```

```

// Body: { qrPayload }
// Retorna: info del cliente + puntos + cupones

// ===== EMPLEADOS =====

// POST /api/scanya/login-empleado
// Body: { nick, pin }
// Retorna: token de sesión + permisos

// GET /api/scanya/mi-resumen
// Retorna: métricas del día del empleado actual

```

Checklist ScanYA

Frontend:

- Pantalla principal con escáner QR
- Login empleado (nick + PIN)
- Formulario de registro de venta
- Cálculo visual de puntos
- Validación de cupones
- Validación de canjes
- Resumen del día
- Historial de transacciones
- Confirmación de venta exitosa
- Manejo de errores (QR expirado, cliente no encontrado)

Backend:

- Tabla **transacciones**
- Endpoint POST **/api/scanya/venta**
- Endpoint POST **/api/scanya/identificar-cliente**
- Endpoint POST **/api/scanya/validar-cupon**
- Endpoint POST **/api/scanya/validar-canje**
- Endpoint POST **/api/scanya/login-empleado**
- Middleware de permisos por empleado
- Trigger: actualizar puntos del cliente
- Trigger: actualizar nivel si corresponde
- Notificación push al cliente

5.6 CardYA - Tarjeta de Lealtad Digital PENDIENTE

Sistema de puntos con niveles (Bronce → Plata → Oro) donde el usuario acumula puntos en cada negocio participante.

Elemento	Descripción
----------	-------------

Elemento	Descripción
Función	Tarjeta de lealtad digital con niveles
Quién usa	Usuarios en modo Personal
Ruta App	/cardya
Ruta Widget	/cardya-widget (PWA standalone)
Niveles	bronze → silver → gold

⚠ IMPORTANTE - Puntos por Negocio

- Cada negocio configura el valor de sus puntos (1 punto = \$X pesos)
- Los puntos son **específicos por negocio** (NO se transfieren entre negocios)
- El nivel CardYA es **GLOBAL** (suma puntos de todos los negocios)
- Los multiplicadores (1x, 1.25x, 1.5x) aplican en cada negocio

Ejemplo Práctico:

Usuario con nivel PLATA compra en dos negocios:

⌚ Pizzería Roma (valor punto: \$10)
 Compra: \$500
 Cálculo: $500 / 10 = 50$ puntos base
 Con multiplicador Plata (1.25x): 62 puntos
 → +62 puntos en Pizzería Roma

⌚ Café Central (valor punto: \$5)
 Compra: \$500
 Cálculo: $500 / 5 = 100$ puntos base
 Con multiplicador Plata (1.25x): 125 puntos
 → +125 puntos en Café Central

Puntos lifetime del usuario: +187 (suma para calcular nivel global)

🏆 Sistema de Niveles

⌚ BRONCE (Nivel Inicial)

Requisito:	0 - 4,999 puntos lifetime
Multiplicador:	1.0x
Color:	Café/Bronce (#CD7F32)
Badge:	⌚ BRONCE

Beneficios:

- Acumulación estándar de puntos (1x)
 - Acceso a rifas públicas
 - Cupones básicos
 - CardYA digital con QR
-

PLATA (Nivel Medio)

Requisito: 5,000 - 14,999 puntos lifetime
Multiplicador: 1.25x
Color: Plata (#C0C0C0)
Badge:  PLATA

Beneficios:

- +25% puntos extra en todas las compras
 - Cupones exclusivos mensuales
 - Prioridad en rifas (2 boletos por 1)
 - Badge de Plata en perfil
 - Acceso anticipado a promociones (1 día antes)
-

ORO (Nivel Premium)

Requisito: 15,000+ puntos lifetime
Multiplicador: 1.5x
Color: Dorado (#FFD700)
Badge:  ORO

Beneficios:

- +50% puntos extra en todas las compras
- Cupones premium exclusivos
- Rifas VIP exclusivas (premios mayores)
- Prioridad máxima (3 boletos por 1)
- Badge de Oro en perfil
- Acceso anticipado a promociones (3 días antes)

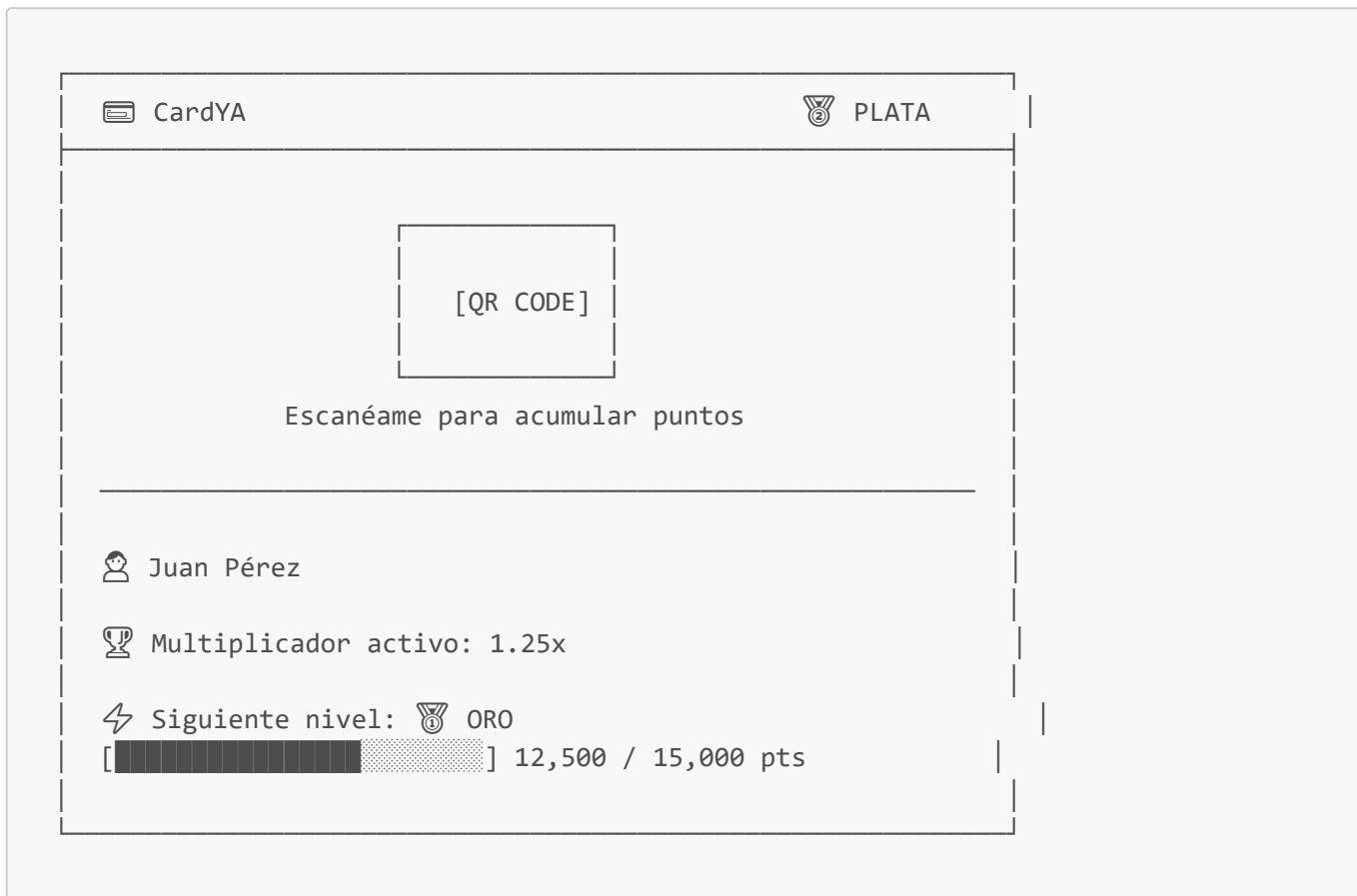
 **NOTA:** Los beneficios son propuestas iniciales. La decisión final se tomará durante la implementación.

Diseño Visual CardYA

Tarjeta por Nivel

Nivel	Gradiente Tailwind	Características
🏅 Bronce	from-amber-700 via-amber-800 to-amber-900	Tonos cálidos, sólido
🏅 Plata	from-gray-300 via-gray-400 to-gray-500	Patrón puntos radiales
🏅 Oro	from-yellow-400 via-yellow-500 to-yellow-600	Patrón diagonal premium

Estructura Visual



Vista "Mis Puntos por Negocio"



Total lifetime: 12,500 puntos (Nivel PLATA)

QR Dinámico con Expiración

Seguridad del QR:

- QR contiene: `userId + timestamp + hash`
- Expira cada **2 minutos**
- Se regenera automáticamente
- Hash validado en backend

```
// Generación de QR seguro
interface QRpayload {
  usuarioId: string;
  timestamp: number;
  hash: string; // SHA256(usuarioId + timestamp + SECRET)
}

// Validación en ScanYA
function validarQR(payload: QRpayload): boolean {
  const ahora = Date.now();
  const dosMinutos = 2 * 60 * 1000;

  // Verificar expiración
  if (ahora - payload.timestamp > dosMinutos) {
    throw new Error('QR expirado');
  }

  // Verificar hash
  const hashEsperado = generarHash(payload.usuarioId, payload.timestamp);
  if (payload.hash !== hashEsperado) {
    throw new Error('QR inválido');
  }

  return true;
}
```

Modelo de Datos

Tabla: usuarios (campos CardYA)

```
-- Agregar a tabla usuarios existente
ALTER TABLE usuarios ADD COLUMN puntos_lifetime INTEGER DEFAULT 0;
```

```

ALTER TABLE usuarios ADD COLUMN nivel_cardya VARCHAR(10) DEFAULT 'bronze';

-- puntos_lifetime: Total histórico (NUNCA baja, solo sube)
-- nivel_cardya: 'bronze' | 'plata' | 'oro' (calculado automáticamente)

```

Tabla: puntos_por_negocio

```

CREATE TABLE puntos_por_negocio (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    usuario_id UUID REFERENCES usuarios(id),
    negocio_id UUID REFERENCES negocios(id),
    sucursal_id UUID REFERENCES negocio_sucursales(id),
    puntos_disponibles INTEGER DEFAULT 0, -- Puede bajar al canjear
    puntos_acumulados INTEGER DEFAULT 0, -- Total en este negocio
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),
    UNIQUE(usuario_id, negocio_id)
);

```

Tabla: historial_puntos

```

CREATE TABLE historial_puntos (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    usuario_id UUID REFERENCES usuarios(id),
    negocio_id UUID REFERENCES negocios(id),
    sucursal_id UUID REFERENCES negocio_sucursales(id),
    tipo VARCHAR(20) NOT NULL, -- 'ganado' | 'canjeado' | 'expirado'
    puntos INTEGER NOT NULL,
    monto_compra DECIMAL(10,2),
    multiplicador DECIMAL(3,2),
    nivel_al_momento VARCHAR(10),
    transaccion_id UUID,
    descripcion TEXT,
    created_at TIMESTAMP DEFAULT NOW()
);

```

■ Lógica de Cálculo

```

// Calcular nivel basado en puntos lifetime
function calcularNivel(puntosLifetime: number): NivelCardYA {
    if (puntosLifetime >= 15000) return 'oro';
    if (puntosLifetime >= 5000) return 'plata';
    return 'bronze';
}

// Obtener multiplicador por nivel

```

```

function getMultiplicador(nivel: NivelCardYA): number {
  const multiplicadores = { bronce: 1.0, plata: 1.25, oro: 1.5 };
  return multiplicadores[nivel];
}

// Calcular progreso hacia siguiente nivel
function getProgresoNivel(puntosLifetime: number): ProgresoNivel {
  if (puntosLifetime >= 15000) {
    return { nivel: 'oro', progreso: 100, siguiente: null, faltantes: 0 };
  }
  if (puntosLifetime >= 5000) {
    const progreso = ((puntosLifetime - 5000) / 10000) * 100;
    return {
      nivel: 'plata',
      progreso,
      siguiente: 'oro',
      faltantes: 15000 - puntosLifetime
    };
  }
  return {
    nivel: 'bronce',
    progreso: (puntosLifetime / 5000) * 100,
    siguiente: 'plata',
    faltantes: 5000 - puntosLifetime
  };
}

```

█ Componentes Frontend

```

pages/private/cardya/
├── PaginaCardYA.tsx          # Vista principal
└── components/
    ├── TarjetaCardYA.tsx      # Card con diseño dinámico por nivel
    ├── QRDinamico.tsx        # QR que expira en 2 min
    ├── ProgresoNivel.tsx     # Barra hacia siguiente nivel
    ├── BadgeNivel.tsx         # Badge 🏆🏆🏆
    ├── ResumenPuntos.tsx      # Total y multiplicador
    ├── PuntosPorNegocio.tsx   # Lista de negocios con puntos
    ├── HistorialPuntos.tsx    # Timeline de movimientos
    ├── ModalCanjear.tsx       # Modal para canjear puntos
    └── ListaRecompensas.tsx    # Recompensas disponibles por negocio

hooks/
├── useCardYA.ts               # Estado de tarjeta y puntos
├── useQRDinamico.ts           # Generación y refresh de QR
└── useNivel.ts                 # Cálculo de nivel y progreso

stores/
└── useCardYAStore.ts          # Estado global CardYA

```

🔗 Endpoints Backend

```
// GET /api/cardya/mi-tarjeta
// Retorna: info del usuario, nivel, puntos por negocio

// GET /api/cardya/qr
// Retorna: QR payload con hash y timestamp

// GET /api/cardya/puntos/:negocioId
// Retorna: puntos disponibles en un negocio específico

// GET /api/cardya/historial
// Query: ?negocioId=xxx&page=1&limit=20
// Retorna: historial de movimientos

// POST /api/cardya/canjeear
// Body: { negocioId, recompensaId, puntos }
// Retorna: código de canje

// GET /api/cardya/recompensas/:negocioId
// Retorna: recompensas disponibles para canjear
```

✓ Checklist CardYA

Frontend:

- 3 diseños de TarjetaCardYA (Bronce, Plata, Oro)
- QR dinámico con countdown de 2 min
- Barra de progreso hacia siguiente nivel
- Badge de nivel visible
- Multiplicador mostrado prominentemente
- Lista de puntos por negocio
- Historial de movimientos
- Modal de canje de puntos
- Notificación de subida de nivel

Backend:

- Migración: agregar `puntos_lifetime`, `nivel_cardya` a usuarios
- Tabla `puntos_por_negocio`
- Tabla `historial_puntos`
- Service: `calcularNivel()`, `getMultiplicador()`
- Endpoint: GET `/api/cardya/mi-tarjeta`
- Endpoint: GET `/api/cardya/qr`
- Endpoint: GET `/api/cardya/historial`
- Endpoint: POST `/api/cardya/canjeear`

- Trigger: actualizar nivel automáticamente al sumar puntos
 - Notificación push de subida de nivel
-

5.6.1 PWA y Widgets Descargables

La app principal y componentes clave pueden instalarse como apps independientes en el home screen.

Aplicaciones Disponibles

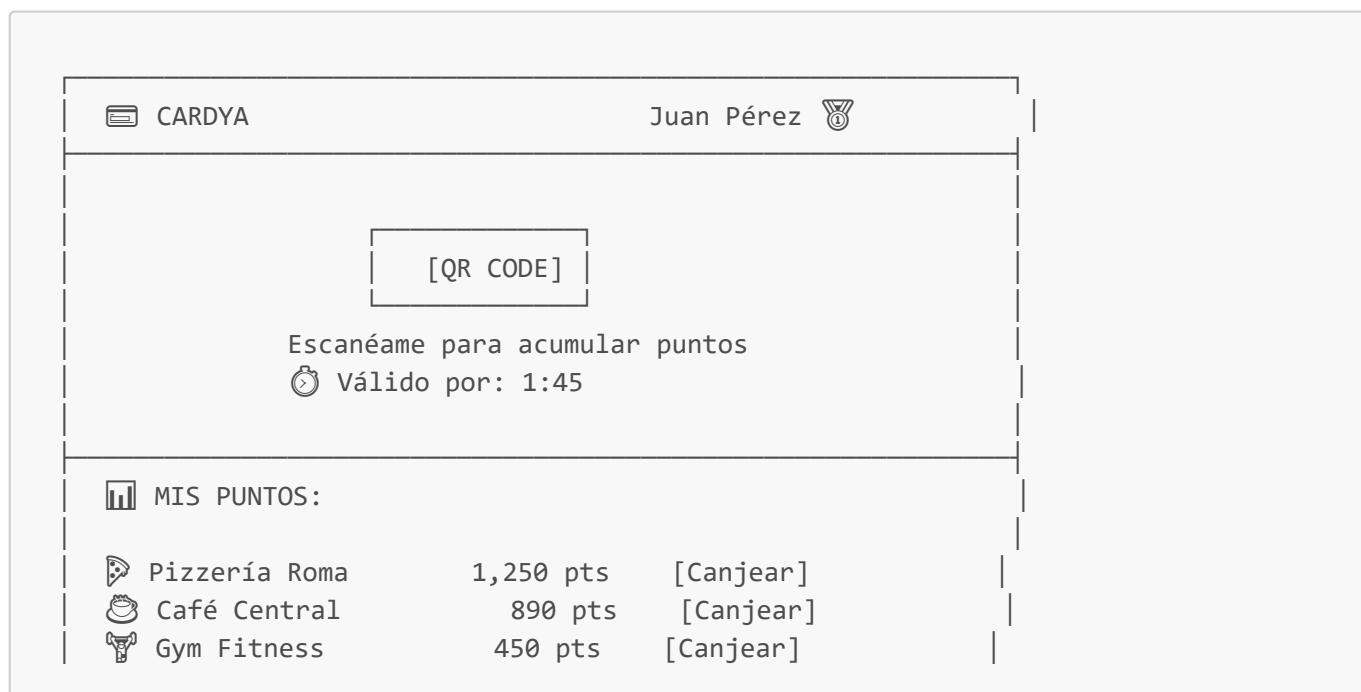
App/Widget	Descripción	Usuarios	Ruta
AnunciaYA	App principal completa	Todos	/
CardYA Widget	Solo tarjeta y puntos	Modo Personal	/cardya-widget
ScanYA Widget	Solo punto de venta	Comercial + Empleados	/scanya-widget

CardYA Widget (Usuarios)

Características:

- App standalone ultra-ligera (~50KB)
- Acceso directo desde home screen
- Solo muestra:
 - Tarjeta CardYA con QR dinámico
 - Puntos disponibles por negocio
 - Cupones activos
 - Códigos de canje generados
- Sincronización en tiempo real
- Funciona offline (muestra último estado guardado)

Interfaz:



[Ver todos]

MIS CUPONES (3)

30% dto - Pizzería Roma [Ver QR]
2x1 Café [Ver QR]

[Ver todos]

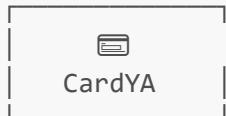
[Ir a app completa]

Instalación:

Desde AnunciaYA:

- Mi Perfil → Configuración → Widgets
- "Instalar CardYA en pantalla de inicio"
- Sistema muestra diálogo PWA nativo
- Usuario confirma instalación

Resultado en home screen:



ScanYA Widget (Comerciantes/Empleados)

Características:

- App standalone para punto de venta
- Acceso directo desde home screen
- Interfaz simplificada sin navegación compleja
- Login persistente (no pide credenciales cada vez)
- Solo funciones esenciales:
 - Escanear QR
 - Registrar venta
 - Validar cupones
 - Ver resumen del día
- Ideal para dispositivos compartidos en caja

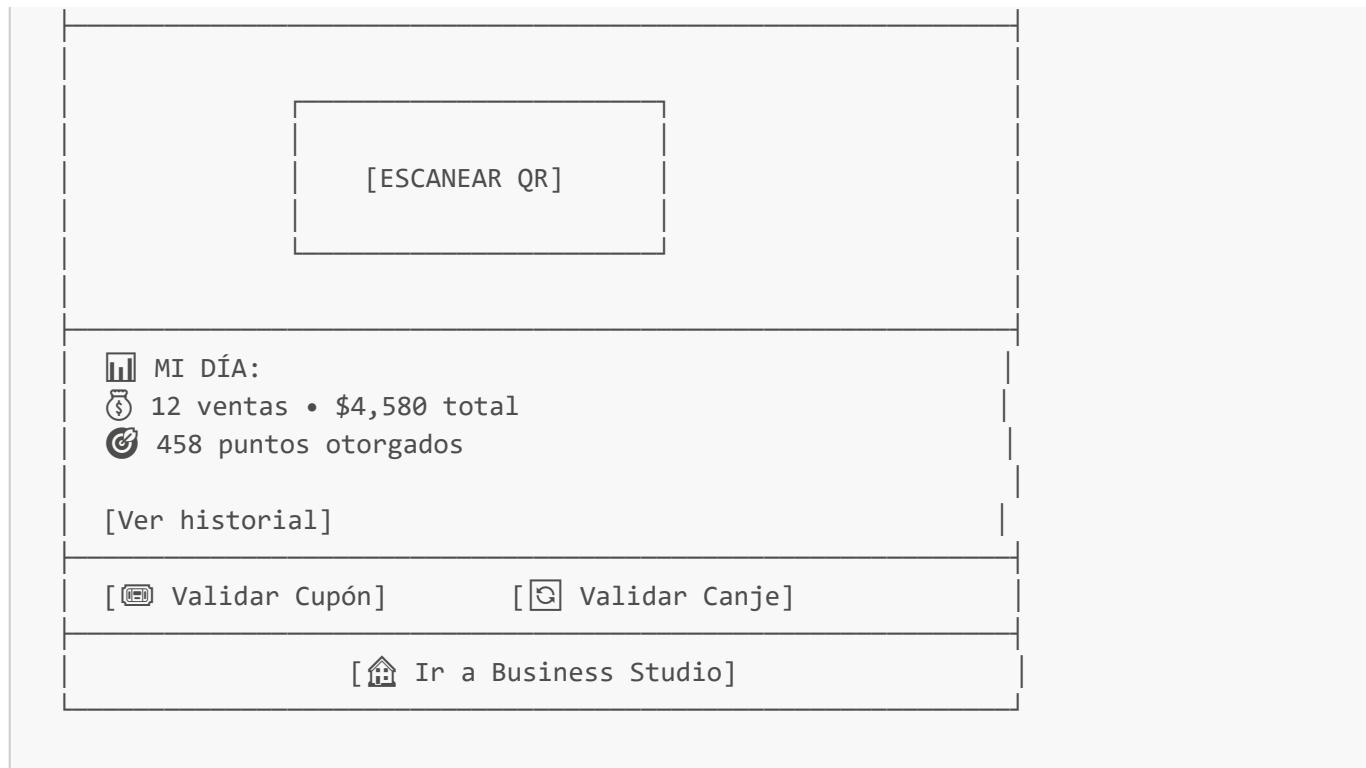
Interfaz:

SCANYA

Pizzería Roma

@carlos - Cajero

[Cerrar sesión]

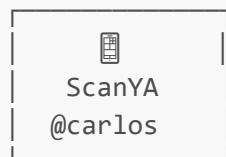


Instalación para Empleados:

El dueño desde Business Studio:

- Empleados → Seleccionar empleado
- "Generar QR de instalación"
- Empleado escanea el QR con su celular
- Se abre página de instalación PWA
- Empleado instala + queda pre-autenticado
- Ícono aparece en su home screen

Resultado:



🔧 Implementación Técnica

Manifests PWA:

```
// CardYA - manifest.cardya.json
{
  "name": "CardYA - Tu Tarjeta de Puntos",
  "short_name": "CardYA",
  "description": "Tu tarjeta de lealtad digital",
  "start_url": "/cardya-widget",
  "scope": "/cardya-widget",
```

```

"display": "standalone",
"orientation": "portrait",
"background_color": "#ffffff",
"theme_color": "#4F46E5",
"icons": [
  { "src": "/icons/cardya-192.png", "sizes": "192x192", "type": "image/png" },
  { "src": "/icons/cardya-512.png", "sizes": "512x512", "type": "image/png" }
]
}

// ScanYA - manifest.scanya.json
{
  "name": "ScanYA - Punto de Venta",
  "short_name": "ScanYA",
  "description": "Registra ventas y otorga puntos",
  "start_url": "/scanya-widget",
  "scope": "/scanya-widget",
  "display": "standalone",
  "orientation": "portrait",
  "background_color": "#1F2937",
  "theme_color": "#10B981",
  "icons": [
    { "src": "/icons/scanya-192.png", "sizes": "192x192", "type": "image/png" },
    { "src": "/icons/scanya-512.png", "sizes": "512x512", "type": "image/png" }
  ]
}

```

Service Worker - Offline First:

```

// sw-cardya.ts
const CACHE_NAME = 'cardya-v1';
const ASSETS = [
  '/cardya-widget',
  '/cardya-widget/qr',
  '/icons/cardya-192.png',
  '/icons/cardya-512.png'
];

self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => cache.addAll(ASSETS))
  );
});

self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request).then((cached) => {
      // Network first, fallback to cache
      return fetch(event.request)
        .then((response) => {
          // Actualizar cache con respuesta fresca
          const clone = response.clone();

```

```

        caches.open(CACHE_NAME).then((cache) => cache.put(event.request,
clone));
        return response;
    })
    .catch(() => cached); // Si falla network, usar cache
})
);
});
});
```

Deep Linking:

```

// Abrir app completa desde widget
const abrirAppCompleta = () => {
    // Intentar deep link primero
    window.location.href = 'anunciaya://home';

    // Fallback a web después de 500ms
    setTimeout(() => {
        window.location.href = 'https://anunciaya.com';
    }, 500);
};

// Abrir widget desde app
const abrirCardYAWidget = () => {
    window.location.href = '/cardya-widget';
};
```

Rutas en Router:

```

// router/index.tsx
const widgetRoutes = [
{
    path: '/cardya-widget',
    element: <CardYAWidget />,
    // Sin layout principal, standalone
},
{
    path: '/scanya-widget',
    element: <ScanYAWidget />,
    // Requiere auth (empleado o dueño)
}
];
```

Checklist PWA y Widgets

Configuración:

- Manifest para app principal
- Manifest para CardYA widget
- Manifest para ScanYA widget
- Service workers por widget
- Íconos en múltiples tamaños

CardYA Widget:

- Vista QR dinámico
- Lista de puntos por negocio
- Lista de cupones
- Botón "Ir a app completa"
- Modo offline con datos cacheados

ScanYA Widget:

- Login empleado persistente
- Escáner QR
- Registro de venta simplificado
- Resumen del día
- Botón "Ir a Business Studio"
- QR de instalación para empleados

General:

- Deep linking entre apps
 - Detección de instalación PWA
 - Prompt de instalación personalizado
 - Actualización automática de service worker
-

5.10 ChatYA Completo - OCTAVO PENDIENTE

Elemento	Descripción
Posición	Overlay persistente, minimizable (solo PC)
Tecnología	Socket.io + MongoDB + WebRTC
Historial	Unificado por usuario (ambos modos)
Persistencia	Recuerda posición de cierre

Comportamiento por Modo

Modo Activo	Usuario se muestra como
Personal	Nombre personal + avatar
Comercial	Nombre negocio + logo

Funcionalidades Core

Mensajería:

- Mensajes en tiempo real (Socket.io)
- Estados: enviado → entregado → leído
- Responder mensajes (reply)
- Reenviar mensajes
- Mensajes fijados
- Buscar en conversación

Multimedia:

- Enviar imágenes (upload optimista a Cloudinary)
- Previsualización de imágenes en input antes de enviar
- Envío optimista sin retrasos (imagen se muestra inmediatamente)
- Enviar audios
- Video llamadas (WebRTC)
- Llamadas de voz (WebRTC)

Organización:

- Lista de conversaciones
- Chats fijados
- Agregar a contactos
- Notas personales (como WhatsApp)
- Filtrado de mensajes según sección de contacto
- Buscar usuarios y negocios

Notificaciones:

- Badge de mensajes no leídos
- Sonidos de llegada de mensajes
- Vibraciones (vista móvil)
- Notificaciones push

Contexto por Sección:

Usuario contacta desde:

```
|--- /negocios → "Hola, vi tu negocio en el directorio"  
|--- /marketplace → "Hola, me interesa tu publicación: [título]"  
|--- /ofertas → "Hola, vi tu oferta: [nombre oferta]"  
|--- /empleos → "Hola, me interesa la vacante: [puesto]"  
|--- /dinamicas → "Hola, quiero info sobre: [rifa/sorteo]"
```

Perfil de Usuario en Chat**Vista Personal:**

- Nombre
- Foto de perfil

- Rating promedio
- Información básica

Vista Comercial (Negocio):

- Nombre del negocio
- Logo
- Rating promedio
- Catálogo de productos
- Catálogo de servicios
- Horarios de atención
- Direcciones (múltiples sucursales)
- Mapa "Cómo llegar" (Leaflet)
- Información detallada del negocio

Sistema de Calificaciones en Chat

Comerciante puede calificar a Usuario (OPCIONAL):

Desde el perfil del usuario en el chat:

- └── Comerciante deja calificación (1-5 estrellas)
- └── Comerciante deja reseña (texto opcional)
- └── Usuario puede ver calificación recibida
- └── Usuario puede ver reseña que le pusieron

Validación:

- Solo comerciantes pueden calificar usuarios
- Solo si tuvieron interacción (pedido/chat)
- Es completamente opcional
- Usuario debe poder ver su historial de calificaciones

Componentes

```
components/chat/
├── ChatOverlay.tsx           # Contenedor principal
├── ListaConversaciones.tsx   # Sidebar con chats
├── Conversacion.tsx          # Ventana de chat activa
├── MensajeBurbuja.tsx        # Mensaje individual
├── InputMensaje.tsx          # Input con preview
├── EstadoConexion.tsx        # Indicador online/offline
├── PreviewImagen.tsx         # Preview antes de enviar
├── MensajeFijado.tsx         # Mensaje pinneado
├── PerfilUsuario.tsx          # Perfil en chat
├── PerfilNegocio.tsx          # Perfil negocio con catálogo
├── CalificacionUsuario.tsx    # Calificar usuario (comerciante)
├── NotasPersonales.tsx        # Notas estilo WhatsApp
└── BuscadorChats.tsx          # Buscar mensajes/usuarios
```

```
└── VideoLlamada.tsx          # Interfaz video llamada  
└── LlamadaVoz.tsx           # Interfaz llamada voz
```

Flujo de Upload Optimista

```
// 1. Usuario selecciona imagen  
const handleSelectImage = (file) => {  
    // Preview inmediato  
    const preview = URL.createObjectURL(file);  
    setImagePreview(preview);  
};  
  
// 2. Usuario envía  
const handleSend = async () => {  
    const tempId = generateTempId();  
  
    // Mostrar mensaje inmediatamente  
    addMessage({  
        id: tempId,  
        type: 'image',  
        url: imagePreview, // URL temporal local  
        status: 'uploading'  
    });  
  
    // Upload a Cloudinary en background  
    try {  
        const cloudinaryUrl = await uploadToCloudinary(file);  
  
        // Actualizar con URL real  
        updateMessage(tempId, {  
            url: cloudinaryUrl,  
            status: 'sent'  
        });  
    } catch (error) {  
        // Marcar como error, permitir reintentar  
        updateMessage(tempId, { status: 'error' });  
    }  
};
```

Estado del Chat

Desktop:

- Overlay persistente (no se cierra al navegar)
- Puede minimizarse
- Recuerda posición de última vez
- Recuerda si estaba minimizado/maximizado

Mobile:

- Pantalla completa
 - Push notifications
 - Vibraciones
-

5.11 Cuponera Digital ([/mis-cupones](#)) - SEXTO

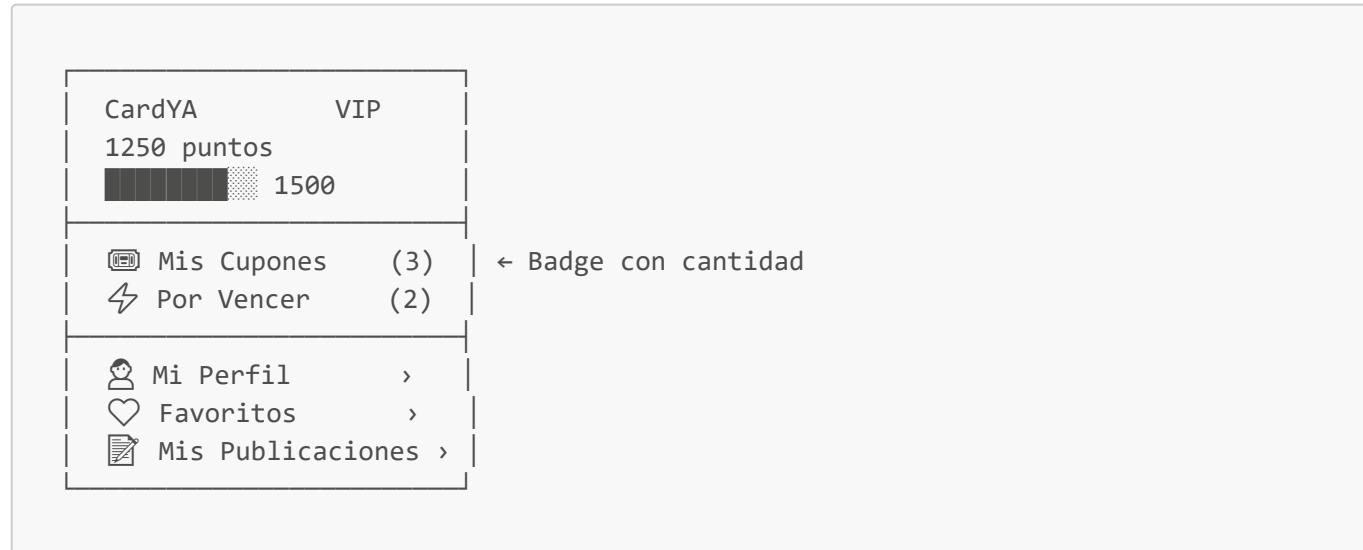
Elemento	Descripción
Función	Zona centralizada de cupones guardados
Quién usa	Usuarios en modo Personal
Ruta	/mis-cupones

Diferencia: Ofertas vs Cupones

Aspecto	Ofertas (/ofertas)	Cupones (/mis-cupones)
Vencimiento	Sin fecha / permanente	Fecha específica
Usos	Múltiples, ilimitado	1 vez por cliente
Ejemplo	"2x1 pizzas todos los martes"	"30% descuento - solo 20-25 Dic"
Propósito	Promociones regulares	Eventos especiales
Origen	Negocio crea en Studio	Negocio envía a clientes
Visibilidad	Pública para todos	Personal (guardados)
Canje	Mostrar en negocio	Escanear QR único

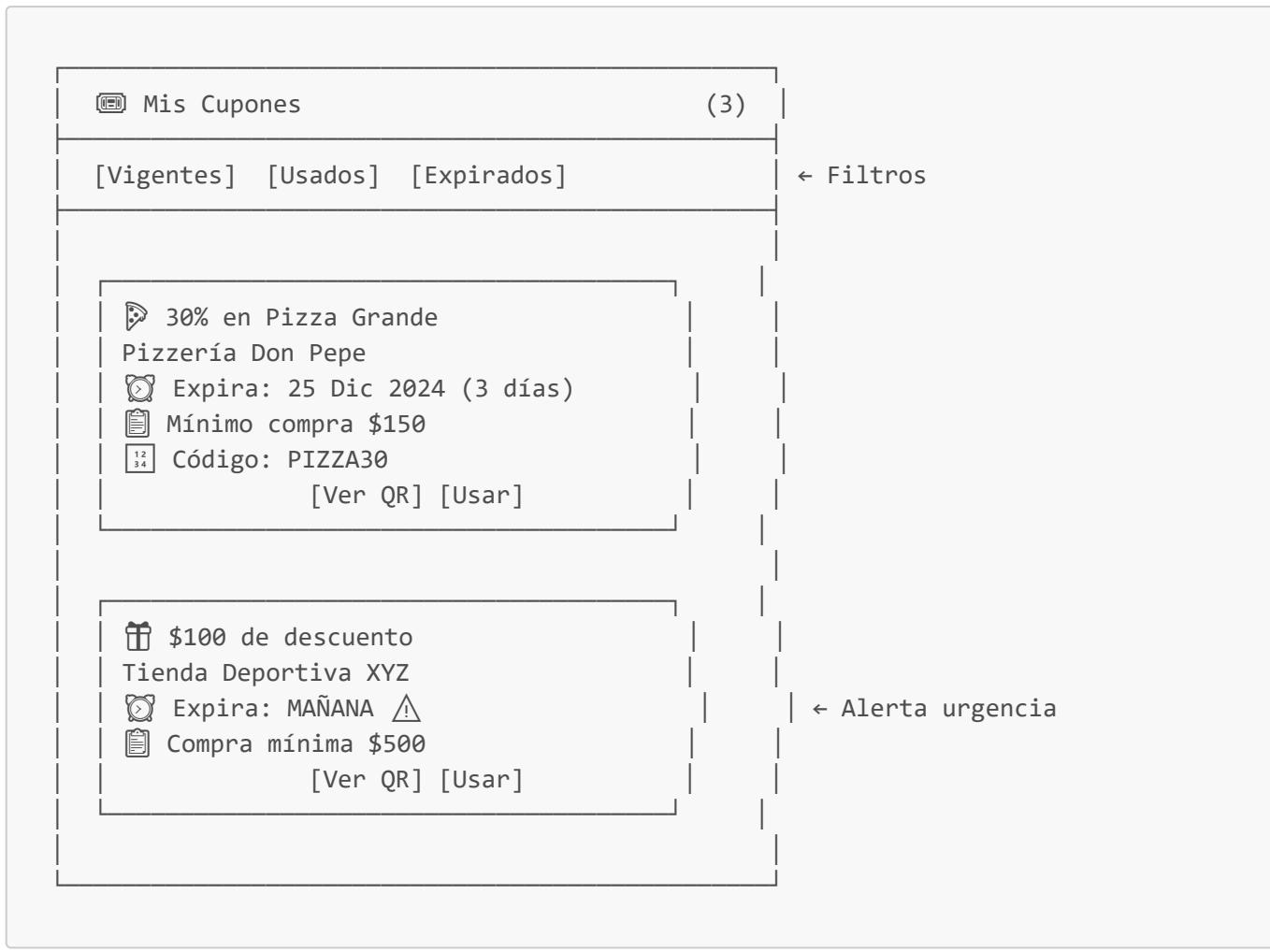
Ubicación en UI

Columna Izquierda (Desktop):



The screenshot shows the left sidebar of the CardYA app on a desktop browser. At the top, there's a summary section with 'CardYA' and 'VIP' status, '1250 puntos' (points), and a progress bar showing '1500' points reached. Below this are three categories with counts: 'Mis Cupones' (3), 'Por Vencer' (2), 'Mi Perfil' (with a user icon), 'Favoritos' (with a heart icon), and 'Mis Publicaciones' (with a document icon). To the right of the 'Mis Cupones' and 'Por Vencer' counts is a note: '← Badge con cantidad' (← badge with quantity).

Estructura de la Cuponera



Notificaciones de Cupones

Trigger	Notificación
Cupón nuevo recibido	" ¡Nuevo cupón! 30% en Pizza de Don Pepe"
3 días antes de expirar	" Tu cupón de Café Central expira en 3 días"
1 día antes de expirar	" ¡Último día! Tu cupón de \$100 expira mañana"
Cupón usado	" Cupón canjeado exitosamente"
Cupón expirado	" Tu cupón de Pizzería ha expirado"

Tablas Involucradas

Tabla	Propósito
cupones	Catálogo de cupones creados por negocios
cupon_usuarios	Cupones guardados/asignados por usuario
cupon_usos	Historial de canjes realizados

Tabla	Propósito
cupon_galeria	Imágenes del cupón (opcional)

Estructura cupon_usuarios:

```
CREATE TABLE cupon_usuarios (
    id UUID PRIMARY KEY,
    cupon_id UUID REFERENCES cupones(id),
    usuario_id UUID REFERENCES usuarios(id),
    estado VARCHAR(20) DEFAULT 'activo', -- activo, usado, expirado
    codigo_unico VARCHAR(20),
    fecha_guardado TIMESTAMP,
    fecha_usado TIMESTAMP,
    UNIQUE(cupon_id, usuario_id)
);
```

📱 Componentes Frontend Cuponera

```
pages/private/cupones/
├── PaginaMisCupones.tsx          # Lista principal
├── PaginaCuponDetalle.tsx        # Modal/página detalle
└── components/
    ├── TarjetaCupon.tsx          # Card del cupón
    ├── FiltrosCupones.tsx        # Vigentes/Usados/Expirados
    ├── QRCPon.tsx                # QR para canjear
    ├── ContadorExpiracion.tsx    # Countdown
    ├── CondicionesCupon.tsx      # Lista de condiciones
    └── HistorialCupones.tsx       # Cupones usados
```

⌚ Integración con Niveles CardYA

Nivel	Beneficio en Cupones
🏅 Bronce	Cupones básicos públicos
🏅 Plata	Cupones exclusivos mensuales
🏅 Oro	Cupones premium + acceso anticipado

```
async obtenerCuponesDisponibles(usuarioId: string) {
  const usuario = await Usuario.findById(usuarioId);

  return await Cupon.find({
    $or: [
      { nivel_requerido: null },
      { nivel_requerido: { $in: [usuario.nivel] } }
    ]
  });
}
```

```
        { nivel_requerido: usuario.nivel_cardya },
        { nivel_requerido: { $lt: getNivelNumero(usuario.nivel_cardya) } }
    ]
});
}
```

⌚ Flujo de Uso Cupones

1. Usuario recibe cupón
 - └ Negocio envía cupón específico
 - └ O usuario guarda desde sección Ofertas
2. Usuario va a "Mis Cupones"
 - └ Ve lista de cupones vigentes
 - └ Countdown de expiración visible
3. Usuario en el negocio
 - └ Abre cupón → Muestra QR
 - └ Empleado escanea con ScanYA
 - └ Cupón marcado como "Usado"
4. Post-canje
 - └ Cupón pasa a "Usados"
 - └ Se registra en historial

✓ Checklist Cuponera

Básicas:

- Lista de cupones guardados
- Filtrar: Vigentes / Usados / Expirados
- Ver condiciones de canje
- Fecha de expiración con countdown
- Generar código QR para canjear
- Marcar cupón como "Usado"

Notificaciones:

- Push "Tu cupón expira mañana"
- Badge en menú con cantidad vigentes
- Alerta visual en próximos a expirar

Historial:

- Ver cupones usados anteriormente
- Fecha y lugar de canje
- Ahorro total acumulado

6.0 Ofertas Publicas (/ofertas) PENDIENTE

Elemento	Descripción
Función	Promociones permanentes/recurrentes de negocios
Quién publica	Usuarios en modo Comercial (via Business Studio)
Quién ve	Todos (ambos modos)
Geolocalización	<input checked="" type="checkbox"/> Ofertas cercanas

Tipos de Ofertas

Tipo	Descripción	Ejemplo
Descuento %	Porcentaje de descuento	"20% en toda la tienda"
Descuento \$	Cantidad fija	"\$100 de descuento"
2x1	Dos por uno	"2x1 en hamburguesas"
Combo	Paquete especial	"Combo familiar \$299"
Happy Hour	Horario específico	"50% de 3-6pm"
Día especial	Día de la semana	"Martes de tacos \$15"

Componentes

```

pages/private/ofertas/
├── PaginaOfertas.tsx           # Feed de ofertas
├── PaginaOfertaDetalle.tsx     # Detalle completo
└── components/
    ├── TarjetaOferta.tsx        # Card de la oferta
    ├── FiltrosOfertas.tsx       # Categoría, tipo, distancia
    ├── BadgeTipoOferta.tsx      # 2x1, %, $, etc.
    ├── HorarioOferta.tsx        # Cuándo aplica
    ├── CondicionesOferta.tsx    # Restricciones
    ├── BotonGuardarCupon.tsx    # Convertir a cupón personal
    └── MapaOfertas.tsx          # Ver en mapa
  
```

Funcionalidades

- ☐ Feed de ofertas cercanas (geolocalización)
- ☐ Filtros: categoría, tipo de oferta, distancia
- ☐ Búsqueda por texto
- ☐ Ver detalle completo (condiciones, horarios)
- ☐ Guardar como cupón personal
- ☐ Ver ubicación en mapa

- Compartir oferta
- "Cómo llegar" (navegación)
- Contactar negocio → ChatYA

Estructura de una Oferta

🏷️ 2x1 EN PIZZAS GRANDES
📍 Pizzería Don Pepe • 0.8 km

🏷️ Tipo: 2x1
📅 Válido: Todos los martes
🕒 Horario: 12:00 PM - 10:00 PM
📋 Condiciones:
 • Aplica en pizzas grandes
 • Consumo en local o para llevar
 • No acumulable con otras promociones

💾 Guardar 📍 Cómo llegar 💬 Contactar

6.1 MarketPlace ([/marketplace](#)) ⏳ PENDIENTE

Elemento	Descripción
Función	Compra-venta entre usuarios (estilo Facebook Marketplace)
Quién publica	Solo usuarios en modo Personal
Quién ve	Todos los usuarios (Personal y Comercial)
Quién compra	Todos los usuarios (Personal y Comercial)
Geolocalización	<input checked="" type="checkbox"/> Filtrar por ciudad/distancia

Restricción: Modo Comercial puede VER y COMPRAR, pero NO puede PUBLICAR artículos.

Tipos de Publicaciones

Tipo	Descripción	Ejemplo
Venta	Artículo a precio fijo	"Vendo iPhone 12 - \$8,000"
Gratis	Donación/regalo	"Regalo sofá, recoger en casa"
Intercambio	Trueque	"Cambio PS4 por bicicleta"

Categorías del Marketplace

Electrónica | Vehículos | Hogar | Moda | Deportes | Niños | Mascotas | Otros

Sección Exclusiva: Vehículos

Ruta: [/marketplace/vehiculos](#)

Esta categoría tiene campos especiales debido a la naturaleza de la compra-venta de vehículos:

Campos Adicionales Obligatorios:

Campo	Tipo	Descripción
Tipo de vehículo	Select	Auto, Moto, Camioneta, Camión, etc.
Marca	String	Honda, Toyota, Nissan, etc.
Modelo	String	Civic, Corolla, etc.
Año	Number	2010-2025
Kilometraje	Number	Kilómetros recorridos
Transmisión	Select	Manual, Automática
Combustible	Select	Gasolina, Diésel, Eléctrico, Híbrido
Color	String	Color del vehículo
Placas al corriente	Boolean	Sí/No

Campos Opcionales:

Campo	Descripción
Número de puertas	2, 4, 5
Versión	EX, LX, Sport, etc.
Cilindros	4, 6, 8
Número de serie	VIN
Factura original	Sí/No
Único dueño	Sí/No
Servicios en agencia	Sí/No

Filtros Específicos:

- Por marca
- Por año (rango)
- Por kilometraje (rango)
- Por precio (rango)
- Por transmisión
- Por combustible
- Por tipo de vehículo

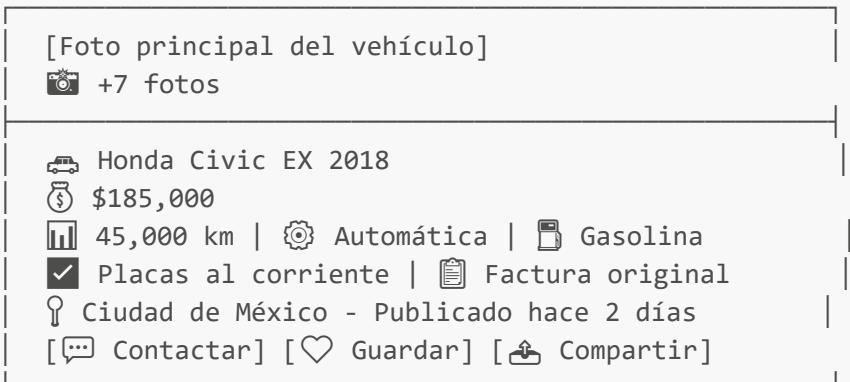
Componente Especial:

```
pages/private/marketplace/vehiculos/
└── PaginaVehiculos.tsx          # Feed de vehículos
└── PaginaVehiculoDetalle.tsx    # Ficha técnica completa
└── PaginaCrearVehiculo.tsx      # Formulario especializado
└── components/
    ├── TarjetaVehiculo.tsx      # Card con datos específicos
    ├── FiltrosVehiculos.tsx     # Filtros especializados
    ├── FichaTecnica.tsx         # Especificaciones completas
    └── ComparadorVehiculos.tsx   # Comparar hasta 3 vehículos
```

Validaciones Especiales:

```
interface ValidacionVehiculo {
  año: number;           // Entre 1990 y año actual + 1
  kilometraje: number;   // Mayor a 0
  precio: number;        // Mayor a 0
  imagenes: string[];    // Mínimo 3, máximo 10
  marca: string;         // Obligatorio
  modelo: string;        // Obligatorio
  placasAlCorriente: boolean; // Obligatorio
}
```

Vista de Tarjeta de Vehículo:



Componentes

```
pages/private/marketplace/
└── PaginaMarketplace.tsx      # Feed principal
└── PaginaPublicacionDetalle.tsx # Detalle del artículo
└── PaginaCrearPublicacion.tsx   # Formulario de publicación
└── PaginaMisPublicaciones.tsx   # Mis artículos publicados
└── components/
```

```

    ├── TarjetaArticulo.tsx      # Card del producto
    ├── FiltrosMarketplace.tsx   # Categoría, precio, distancia
    ├── GaleriaImagenes.tsx     # Carrusel de fotos
    ├── BotonContactar.tsx      # Abre ChatYA
    ├── EstadoPublicacion.tsx   # Disponible/Vendido/Reservado
    └── FormularioPublicacion.tsx # Crear/Editar

```

Funcionalidades

- Feed de publicaciones cercanas
- Filtros: categoría, precio, distancia
- Búsqueda por texto
- Crear publicación (hasta 10 fotos)
- Marcar como: Disponible / Reservado / Vendido
- Contactar vendedor → ChatYA
- Guardar en favoritos
- Reportar publicación
- Compartir (generar link)

Estados de Publicación

```

Disponible → Reservado → Vendido
      ↓
Disponible (si se cancela)

```

Validaciones

Campo	Regla
Título	10-100 caracteres
Descripción	20-2000 caracteres
Precio	> 0 (excepto gratis)
Imágenes	1-10 fotos (mínimo 3 para vehículos)
Categoría	Obligatoria

Modelo de Datos

```

// Base para todas las publicaciones
interface PublicacionBase {
  id: string;
  usuarioId: string;
  titulo: string;
  descripcion: string;
}

```

```

    precio: number;
    imagenes: string[];
    categoria: string;
    ubicacion: Point;
    estado: 'nuevo' | 'usado' | 'como_nuevo';
    disponible: boolean;
    createdAt: Date;
}

// Extensión para vehículos
interface PublicacionVehiculo extends PublicacionBase {
  categoria: 'vehiculos';
  datosVehiculo: {
    // Obligatorios
    tipoVehiculo: 'auto' | 'moto' | 'camioneta' | 'camion' | 'otro';
    marca: string;
    modelo: string;
    año: number;
    kilometraje: number;
    transmision: 'manual' | 'automatica';
    combustible: 'gasolina' | 'diesel' | 'electrico' | 'hibrido';
    color: string;
    placasAlCorriente: boolean;

    // Opcionales
    numeroPuertas?: number;
    version?: string;
    cilindros?: number;
    numeroSerie?: string;
    facturaOriginal?: boolean;
    unicoDueño?: boolean;
    serviciosAgencia?: boolean;
  };
}

```

6.2 Dinámicas - Rifas y Sorteos ([/dinamicas](#)) PENDIENTE

Elemento	Descripción
Función	Rifas y sorteos organizados por negocios y usuarios
Quién organiza	Todos los usuarios (Personal y Comercial)
Quién ve	Todos los usuarios (Personal y Comercial)
Quién participa	Depende del tipo de rifa
Moneda	Puntos CardYA (por negocio) o Dinero (fuera de la app)

SISTEMA CERRADO DE PUNTOS:

- Los puntos son ESPECÍFICOS por negocio
- Solo puedes participar en rifas usando puntos del MISMO negocio

- No se pueden usar puntos de Negocio A para rifa de Negocio B
- Cada negocio maneja su pool de puntos independiente

Tipos de Rifas y Sorteos

Tipo	Moneda	Participantes	Descripción
Con Puntos	Puntos CardYA	Solo clientes del negocio	Rifas usando puntos del mismo negocio
Con Dinero	Pesos MXN	Todos (usuarios de la app)	App organiza, pago fuera de app
Gestión Offline	Cualquiera	Sin registro necesario	App solo gestiona sorteo, participantes externos
Gratuita	Gratis	Todos	Sorteo sin costo

▣ Rifas/Sorteos - Gestión Offline (Sin Registro)

⚠ **Modo Manual:** Organizador gestiona todo, app solo sorteá

🌐 **ACCESO PÚBLICO:** El sorteo en vivo puede ser visto sin login/registro

Características:

- Participantes NO necesitan estar registrados en AnunciaYA
- Organizador registra boletos manualmente
- App solo sirve para hacer el sorteo aleatorio
- Todo se maneja fuera de la app (venta, pago, entrega)
- **Link de sorteo público:** Puede compartirse y verse sin login
- **Transmisión en vivo:** Cualquiera con el link puede ver el sorteo

Casos de uso:

Ejemplo 1: Rifa en evento presencial

- └─ Negocio vende boletos físicos en \$50 c/u
- └─ Al final del evento hace el sorteo
- └─ Usa la app solo para seleccionar ganador aleatorio

Ejemplo 2: Rifa en redes sociales

- └─ Usuario vende boletos por Facebook/Instagram
- └─ Anota nombres/números en la app
- └─ Hace sorteo en vivo usando la app

Ejemplo 3: Rifa escolar

- └─ Escuela vende boletos impresos
- └─ Captura datos en la app
- └─ Sorteo público usando la app

Flujo:

1. Organizador crea rifa tipo "Gestión Offline"
 - └ Define: premio, total de boletos, fecha sorteo
2. Organizador registra participantes MANUALMENTE
 - └ Opción A: Registrar por nombre
 - └ "Juan Pérez - Boleto #001"
 - └ Opción B: Solo número de boleto
 - └ "Boleto #001, #002, #003..."
 - └ Opción C: Importar lista (CSV)
 - └ nombre,telefono,numero_boleto
3. Al momento del sorteo
 - └ Organizador activa sorteo
 - └ Sistema selecciona ganador aleatorio
 - └ Muestra número de boleto ganador + nombre (si existe)
4. Organizador comunica ganador
 - └ Por los medios que usó para vender (presencial, redes, etc.)
 - └ Entrega de premio FUERA de la app
 - └ Marca como "Entregado" en la app (opcional)

Interfaz de Gestión:

RIFA OFFLINE: Bicicleta Montaña
 Deportes López
 GESTIÓN MANUAL - Sin registro requerido

Boletos registrados: 89 / 100
 Sorteo programado: 28 Dic 2024 - 7:00 PM
 Premio: Bicicleta Trek Mountain 29"

[Registrar boleto] [Importar CSV] [Sortear]

Últimos registrados:

- #089 - María González
- #088 - Carlos Ruiz
- #087 - Boleto sin nombre

Modal: Registrar Boleto

Registrar Boleto

Número de boleto: [_____]

Nombre (opcional): [_____]
Teléfono (opcional): [_____]
Email (opcional): [_____]
[Cancelar] [Guardar y agregar otro]
[Guardar y cerrar]

Modal: Importar CSV

Importar Lista de Participantes
Formato CSV: numero_boleto,nombre,telefono,email
Ejemplo: 001,Juan Pérez,5512345678,juan@... 002,María López,5587654321,maria@...
[Seleccionar archivo CSV] [Descargar plantilla]
[Cancelar] [Importar]

Pantalla de Sorteo en Vivo:

SORTEANDO...
[Animación de números girando]
GANADOR:
BOLETO #047 María González Tel: 55-1234-5678
[<input type="checkbox"/> Sortear de nuevo] [<input checked="" type="checkbox"/> Confirmar ganador] [Compartir]

Componentes Adicionales:

```

components/dinamicas/offline/
├── RegistroBoletoManual.tsx      # Form para registrar 1 boleto
└── ImportadorCSV.tsx            # Importar lista
└── ListaParticipantes.tsx       # Ver todos los registrados
└── SorteoEnVivo.tsx             # Pantalla de sorteo animada
└── ResultadoSorteo.tsx          # Mostrar ganador
└── HistorialSorteos.tsx         # Ver sorteos anteriores

```

⌚ Rifas con Puntos (Sistema Cerrado)

Restricción: Solo participan usuarios con puntos EN ESE NEGOCIO

Usuario tiene:

```

├── 500 pts en "Pizzería Roma"
├── 200 pts en "Café Central"
└── 100 pts en "Gym Fitness"

```

Rifas disponibles:

```

├── Rifa "Pizzería Roma" (50 pts/boleto)  PUEDE participar
├── Rifa "Café Central" (30 pts/boleto)  PUEDE participar
└── Rifa "Gym Fitness" (80 pts/boleto)  PUEDE participar (tiene 100)

```

- ✗ NO puede usar puntos de Pizzería para Café
- ✗ NO puede usar puntos de Café para Gym

Validación Backend:

```

// Verificar que usuario tenga puntos EN EL NEGOCIO de la rifa
async function puedeParticipar(usuarioid: string, rifaId: string) {
  const rifa = await getRifa(rifaId);
  const negocioId = rifa.negocioId;

  const puntosUsuario = await getPuntosEnNegocio(usuarioid, negocioId);
  const costoTotal = rifa.costoPorBoleto * cantidadBoletos;

  return puntosUsuario >= costoTotal;
}

```

🏧 Rifas con Dinero (App como Gestor)

⚠ IMPORTANTE: La app NO procesa pagos de estas rifas

Funcionalidad:

- App solo gestiona la organización y sorteo
- Pago de boletos: FUERA de la app (efectivo, transferencia, etc.)

- Entrega de premios: FUERA de la app
- App solo documenta ganador

Flujo:

1. Organizador crea rifa con dinero
 - └ Define: premio, precio boleto, total boletos
2. Usuario interesado
 - └ Contacta organizador por ChatYA
 - └ Pago se hace FUERA de la app
3. Organizador confirma pago
 - └ Marca boleto como "Pagado" en la app
 - └ Sistema asigna boleto al usuario
4. Al cerrar rifa
 - └ Sistema sortea ganador
 - └ Notifica a ganador y organizador
5. Entrega de premio
 - └ Se coordina FUERA de la app
 - └ Organizador marca como "Entregado"

Componente Especial:

```
// RifaDinero.tsx
<RifaDinero>
  <Badge>⚠️ Pago fuera de la app</Badge>
  <Warning>
    Esta rifa requiere pago en efectivo o transferencia.
    Contacta al organizador para coordinar.
  </Warning>
  <BotonContactar>💬 Contactar organizador</BotonContactar>
</RifaDinero>
```

Estructura de una Rifa con Puntos

RIFA: Smart TV 55"	
Mueblería El Hogar	
Costo: 50 puntos por boleto (solo puntos de este negocio)	
Boletos vendidos: 234 / 500	
Cierra: 25 Dic 2024 - 6:00 PM	
Premio: Smart TV Samsung 55" (valor \$12,000)	
Tus puntos en este negocio: 150	

Mis boletos: 0
 [Comprar +1] [Comprar +5] [Comprar +10]

Estructura de una Rifa con Dinero

RIFA: iPhone 15 Pro
 María García (usuario personal)
 PAGO FUERA DE LA APP

Costo: \$100 por boleto
 Boletos vendidos: 45 / 100
 Cierra: 31 Dic 2024 - 8:00 PM
 Premio: iPhone 15 Pro 256GB

El pago se coordina directamente con el organizador
 Mis boletos: 0
 [Contactar para comprar]

Componentes

```

pages/private/dinamicas/
├── PaginaDinamicas.tsx          # Lista de rifas y sorteos activos
├── PaginaDinamicaDetalle.tsx    # Detalle y compra de boletos
├── PaginaMisDinamicas.tsx       # Rifas/sorteos donde participo
└── PaginaCrearDinamica.tsx      # Crear rifa o sorteo
  └── components/
    ├── TarjetaDinamica.tsx
    ├── CompradorBoletos.tsx      # Solo para rifas con puntos
    ├── ContactoOrganizador.tsx   # Para rifas con dinero
    ├── ProgressBar.tsx           # Boletos vendidos
    ├── ContadorRegresivo.tsx
    ├── BotonCompartir.tsx
    ├── BadgeTipoRifa.tsx         # Puntos vs Dinero
    └── HistorialGanadores.tsx
  
```

Funcionalidades

Generales:

- Ver rifas y sorteos activos
- Filtrar por tipo (puntos, dinero, offline, gratis)
- Filtrar por negocio/organizador
- Ver mis boletos comprados

- Contador regresivo para cierre
- Notificación de ganador
- Historial de rifas ganadas
- Crear y administrar rifas (ambos modos)

Rifas con Puntos:

- Comprar boletos con puntos DEL MISMO NEGOCIO
- Ver balance de puntos en ese negocio
- Validación: solo participar si tienes puntos ahí

Rifas con Dinero:

- Botón "Contactar organizador"
- Organizador confirma pago manual
- Asignación de boleto post-confirmación
- Sin procesamiento de pago en app

Rifas Offline (Gestión Manual):

- Registrar boleto individual (número + datos opcionales)
- Importar lista de participantes (CSV)
- Exportar plantilla CSV
- Ver lista completa de participantes registrados
- Editar/eliminar participantes
- Sorteo en vivo con animación
- Permitir re-sorteo si es necesario
- Compartir resultado (screenshot/link)
- Historial de sorteos realizados

Flujo del Sorteo

Rifas con Puntos:

1. Negocio crea rifa (fecha cierre + premio + costo en puntos)
2. Usuarios con puntos EN ESE NEGOCIO compran boletos
3. Sistema descuenta puntos del saldo del usuario en ese negocio
4. Al cerrar → Sistema selecciona ganador aleatorio
5. Notificación push al ganador
6. Ganador reclama en el negocio
7. Negocio marca premio como entregado

Rifas con Dinero:

1. Usuario crea rifa (fecha cierre + premio + costo \$\$\$)
2. Interesados contactan por ChatYA
3. Pago se hace FUERA de la app
4. Organizador confirma pago y asigna boleto

5. Al cerrar → Sistema selecciona ganador aleatorio
6. Notificación push al ganador
7. Entrega de premio FUERA de la app
8. Organizador marca como entregado

Rifas Offline (Gestión Manual):

1. Organizador crea rifa tipo "Gestión Offline"
 - └ Define premio, total boletos, fecha sorteo
2. Organizador vende boletos por sus propios medios
 - └ Presencial, redes sociales, WhatsApp, etc.
3. Organizador registra participantes en la app:
 - └ Opción A: Manual uno por uno
 - └ Opción B: Importar lista CSV
 - └ Datos: número boleto + nombre/tel/email (opcional)
4. Organizador comparte link público del sorteo:
 - └ /dinamicas/rifa/abc123/sorteo-publico
 - └ ¡Cualquiera puede verlo SIN REGISTRO!
5. Al momento del sorteo:
 - └ Organizador activa "Sorteo en vivo"
 - └ Animación de números
 - └ Sistema selecciona ganador aleatorio
 - └ Muestra: #boleto + nombre (si existe)
 - └ Espectadores ven en tiempo real (sin login)
6. Organizador comunica ganador:
 - └ Por WhatsApp, redes, presencial, etc.
 - └ Link público del resultado
7. Entrega de premio FUERA de la app
 - └ Marca como "Entregado" (opcional)
8. Organizador puede:
 - └ Exportar lista de participantes
 - └ Ver historial de sorteos
 - └ Compartir resultado públicamente

Modelo de Datos

```
interface Rifa {
  id: string;
  organizadorId: string;
  negocioId?: string; // Solo si es rifa con puntos
  tipo: 'puntos' | 'dinero' | 'offline' | 'gratis';
  titulo: string;
```

```
descripcion: string;
premio: string;
imagenPremio: string;

// Para rifas con puntos
costoPorBoletoPuntos?: number;

// Para rifas con dinero
costoPorBoletoDinero?: number;

// Para rifas offline
gestionManual?: boolean;
permitirImportarCSV?: boolean;

totalBoletos: number;
boletosVendidos: number;
fechaCierre: Date;
estado: 'activa' | 'cerrada' | 'finalizada';
ganadorId?: string;
ganadorExterno?: { // Para rifas offline
    numeroBoleto: string;
    nombre?: string;
    telefono?: string;
    email?: string;
};
premioEntregado: boolean;
}

interface BoletaRifa {
    id: string;
    rifaId: string;
    usuarioId?: string; // null si es offline
    numeroBoleto: number;

    // Para rifas con dinero
    pagadoFueraApp?: boolean;
    confirmadoPor?: string; // organizadorId
    fechaConfirmacion?: Date;

    // Para rifas offline (participante externo)
    participanteExterno?: {
        nombre?: string;
        telefono?: string;
        email?: string;
    };
}

// Para importación CSV
interface ImportacionCSV {
    rifaId: string;
    archivo: File;
    registrosImportados: number;
    registrosErrores: number;
```

```

    errores?: string[];
}

```

6.3 Empleos ([/empleos](#)) PENDIENTE

Elemento	Descripción
Función	Plataforma bidireccional de empleos y servicios
Quién publica	Todos los usuarios (Personal y Comercial)
Quién ve	Todos los usuarios (Personal y Comercial)
Geolocalización	<input checked="" type="checkbox"/> Empleos y servicios cercanos

Sistema Bidireccional

Dos tipos de publicaciones:

Tipo	Quién publica	Quién aplica/contrata	Descripción
Oferta de Empleo	Modo Comercial	Modo Personal	Negocio busca empleado
Oferta de Servicio	Modo Personal	Todos	Usuario ofrece sus servicios

Flujos:

Negocio (Comercial) busca empleado:

- └── Publica "Oferta de Empleo"
- └── Usuarios Personal aplican
- └── Negocio revisa aplicaciones

Usuario (Personal) ofrece servicio:

- └── Publica "Oferta de Servicio"
- └── Cualquiera puede contactar
- └── Usuario recibe solicitudes

Tipos de Publicaciones

A) Ofertas de Empleo (Comercial publica):

Tipo	Descripción
Tiempo completo	Jornada completa
Medio tiempo	Horario parcial
Por proyecto	Trabajo temporal
Prácticas	Estudiantes

Tipo	Descripción
Freelance	Trabajo independiente

B) Ofertas de Servicio (Personal publica):

Categoría	Ejemplos
Profesionales	Contador, Abogado, Diseñador
Oficios	Plomero, Electricista, Carpintero
Creativos	Fotógrafo, Músico, DJ
Técnicos	Reparación PC, Celulares
Cuidados	Niñera, Cuidador de ancianos
Educación	Tutor, Profesor particular
Transporte	Chofer, Mensajería
Limpieza	Limpieza de hogar, Oficinas
Eventos	Mesero, Chef, Bartender

Estructura de una Oferta de Empleo

Mesero/a - Tiempo Completo
 Restaurante La Palapa • 1.2 km
 OFERTA DE EMPLEO

\$2,500 - \$3,500 semanal
 Lunes a Sábado, 10am - 6pm

Requisitos:

- Experiencia 6 meses mínimo
- Disponibilidad inmediata
- Buena presentación

Publicado: hace 2 días
 12 aplicaciones

[\[Aplicar\]](#) [\[Guardar\]](#) [\[Compartir\]](#)

Estructura de una Oferta de Servicio

Plomero Profesional
 Juan Pérez • ★ 4.8 (23 reseñas)

OFREZCO MI SERVICIO

Desde \$500 (según trabajo)
 Atiendo: CDMX y área metropolitana
 Disponibilidad: Lun-Sáb 8am-6pm
 Especialidades:
 • Reparación de fugas
 • Instalación de tuberías
 • Destapado de drenaje
 WhatsApp disponible

Publicado: hace 1 semana
 18 contactos recibidos
[\[Contactar\]](#) [\[Guardar\]](#) [\[Compartir\]](#)

Componentes

```

pages/private/empleos/
├── PáginaEmpleos.tsx          # Feed principal (empleos + servicios)
├── PáginaEmpleoDetalle.tsx     # Detalle de oferta de empleo
├── PáginaServicioDetalle.tsx    # Detalle de oferta de servicio
├── PáginaMisAplicaciones.tsx   # Empleos donde apliqué
├── PáginaMisServicios.tsx      # Servicios que ofrezco
├── PáginaCrearEmpleo.tsx       # (Comercial) Publicar empleo
└── PáginaCrearServicio.tsx      # (Personal) Publicar servicio
  └── components/
    ├── TarjetaEmpleo.tsx        # Card de empleo
    ├── TarjetaServicio.tsx       # Card de servicio
    ├── FiltrosEmpleos.tsx        # Tipo, categoría, salario, distancia
    ├── FormularioAplicacion.tsx  # Aplicar a empleo
    ├── FormularioServicio.tsx    # Crear servicio
    ├── EstadoAplicacion.tsx     # Estados de aplicación
    ├── ListaAplicantes.tsx       # (Comercial) Ver aplicaciones
    ├── BadgeTipoPublicacion.tsx  # Empleo vs Servicio
    └── CalificaciónServicio.tsx  # Rating para servicios

```

Funcionalidades Usuario Personal

Buscar Empleo:

- Ver ofertas de empleo cercanas
- Filtrar por tipo, salario, categoría
- Aplicar a oferta (adjuntar CV o perfil)
- Ver estado de mis aplicaciones
- Guardar ofertas interesantes
- Contactar empleador → ChatYA

Ofrecer Servicio:

- Publicar servicio profesional/oficio
- Definir categoría, tarifa, zona de atención
- Subir portafolio/fotos de trabajos
- Ver solicitudes recibidas
- Gestionar disponibilidad
- Recibir y responder calificaciones

Funcionalidades Usuario Comercial

Publicar Empleo:

- Crear oferta de empleo
- Ver lista de aplicantes
- Marcar aplicación: Vista / En proceso / Rechazada / Aceptada
- Contactar aplicante → ChatYA
- Cerrar/Pausar oferta

Contratar Servicio:

- Buscar servicios profesionales
- Ver perfil y calificaciones del prestador
- Contactar para cotizar → ChatYA
- Calificar servicio recibido

Estados de Aplicación (Empleos)

```
Enviada → Vista → En proceso → Aceptada
          ↓
          Rechazada
```

Sistema de Calificaciones (Servicios)

Cliente contrata servicio:

- Servicio se completa
- Cliente puede calificar (1-5 estrellas + reseña)
- Calificación aparece en perfil del prestador
- Promedio se actualiza automáticamente

Prestador de servicio:

- Puede responder a reseñas
- Acumula reputación
- Mayor rating = más visible en búsquedas

Modelo de Datos

```
interface Publicacion {
  id: string;
  publicadorId: string;
  tipo: 'empleo' | 'servicio';
  titulo: string;
  descripcion: string;
  ubicacion: Point;
  estado: 'activa' | 'pausada' | 'cerrada';
  createdAt: Date;
}

// Oferta de Empleo (Comercial publica)
interface OfertaEmpleo extends Publicacion {
  tipo: 'empleo';
  negocioId: string;
  tipoJornada: 'completo' | 'medio' | 'proyecto' | 'practicas' | 'freelance';
  salarioMin?: number;
  salarioMax?: number;
  horario: string;
  requisitos: string[];
  beneficios?: string[];
  aplicaciones: number;
}

// Oferta de Servicio (Personal publica)
interface OfertaServicio extends Publicacion {
  tipo: 'servicio';
  usuarioId: string;
  categoria: string; // Plomería, Diseño, etc.
  tarifaDesde: number;
  zonaAtencion: string[];
  disponibilidad: string;
  especialidades: string[];
  portafolio: string[]; // URLs de imágenes
  rating: number;
  totalReseñas: number;
  whatsappDisponible: boolean;
}

// Aplicación a Empleo
interface Aplicacion {
  id: string;
  empleoId: string;
  usuarioId: string;
  cv?: string; // URL
  mensaje: string;
  estado: 'enviada' | 'vista' | 'proceso' | 'aceptada' | 'rechazada';
  createdAt: Date;
}

// Calificación de Servicio
interface CalificacionServicio {
  id: string;
```

```

servicioId: string;
usuarioId: string; // Quien califica
estrellas: number; // 1-5
comentario?: string;
respuesta?: string; // Respuesta del prestador
createdAt: Date;
}

```

Filtros Disponibles

Vista Principal:

- Tipo: Empleos | Servicios | Todos
- Categoría (diferente por tipo)
- Rango de precio/salario
- Distancia
- Fecha de publicación
- Calificación (solo servicios)

6.7 Panel Admin ([/admin](#)) - Sistema Completo

Panel administrativo con gestión de vendedores, niveles de acceso, métricas globales y configuración dinámica de la plataforma.

Acceso: [/admin/*](#) (Requiere cuenta Admin/Vendedor/SuperAdmin)

Sistema de Usuarios Admin

Tipos de Cuenta Admin

Tipo	Nivel	Descripción	Acceso
Vendedor	1-3	Vende suscripciones	Limitado según nivel
Admin	4	Gestión general	Amplio pero no total
SuperAdmin	5	Control total (TÚ)	Sin restricciones

Sistema de Vendedores

Credenciales de Acceso

Login: /admin/login



PANEL ADMIN - LOGIN

Email:

[_____]

Password:

[_____]

[Iniciar sesión]

 Solo personal autorizado

Dashboard del Vendedor

PANEL DE VENTAS

Vendedor: Carlos Martínez - Nivel 2

[Cerrar sesión]

MIS MÉTRICAS DEL MES:

 Meta	 Ventas	 Clientes	 Com.
20 cuentas	15 vendidas	15 activos	\$6,735
75% <input checked="" type="checkbox"/>	\$6,735	12 renovó	(15%)

Gráfica de ventas (últimos 30 días)

[Gráfico de barras]

ALERTAS:

- 3 clientes por renovar esta semana
- 1 cliente no ha completado onboarding

Mi Cartera de Clientes

MI CARTERA

Total: 15 clientes activos

[+ Agregar cliente] [Filtros] [Exportar]

 Buscar... [Estado: Todos ▼] [Orden: Recientes ▼]

Pizzería Roma

Contacto: Juan Pérez • juan@email.com

Estado: ACTIVO • Renovación: 15 Ene 2025

Suscripción desde: 15 Dic 2024 (1 mes)

MRR: \$449 • Lifetime: \$449

📊 Actividad:

- Última sesión: Hoy a las 10:30 AM
- Productos: 25 • Clientes: 89
- Ventas del mes: \$12,450

[Ver detalle] [💬 Contactar] [📊 Reportes]

☕ Café Central

Contacto: Ana López • ana@email.com

Estado: ⚠ POR RENOVAR (5 días) • Vence: 27 Dic

Suscripción desde: 27 Nov 2024 (1 mes)

MRR: \$449 • Lifetime: \$449

⚠ ACCIONES REQUERIDAS:

- Contactar para renovación
- Verificar método de pago

[🔔 Recordar renovación] [💬 Contactar]

Detalle de Cliente

CLIENTE: Pizzería Roma

[← Volver] [Editar] [Historial completo]

📋 INFORMACIÓN GENERAL

Negocio: Pizzería Roma

Dueño: Juan Pérez

Email: juan@email.com

Teléfono: 55-1234-5678

Vendedor asignado: Carlos Martínez (yo)

📅 SUSCRIPCIÓN

Estado: ACTIVO

Plan: Comercial (\$449/mes)

Inicio: 15 Dic 2024

Próxima renovación: 15 Ene 2025 (24 días)

Método de pago: •••• 4242 (Stripe)

MRR: \$449

Lifetime Value: \$449

📊 ACTIVIDAD EN LA PLATAFORMA

Última sesión: Hoy 10:30 AM

Onboarding: Completado

Productos/Servicios: 25

Clientes registrados: 89

Ventas este mes: \$12,450

Empleados: 3

Cupones activos: 2

Ofertas publicadas: 1

HISTORIAL DE PAGOS

- 15 Dic 2024 - \$449 Exitoso

NOTAS DEL VENDEDOR

[Aregar nota...]

- 15 Dic - Cliente muy activo, usa la app diariamente
- 18 Dic - Pidió ayuda con cupones, resuelto

[Guardar nota]

ACCIONES:

- [Enviar mensaje] [Email] [Llamar]
 [Marcar para seguimiento] [Ver reportes detallados]

Estados de Cliente

Estado	Descripción	Color	Acción
<input checked="" type="checkbox"/> ACTIVO	Suscripción pagada	Verde	Monitorear
<input type="checkbox"/> POR RENOVAR	Vence en <7 días	Amarillo	Contactar
<input type="checkbox"/> VENCIDO	Suscripción expiró	Rojo	Urgente
<input type="checkbox"/> NUEVO	<30 días	Azul	Onboarding
<input type="checkbox"/> INACTIVO	No usa la app	Gris	Reactivar
<input checked="" type="checkbox"/> CANCELADO	Canceló suscripción	Rojo oscuro	Win-back

Filtros de Cartera

Filtrar por:

- Estado (Activo, Por renovar, Vencido, etc.)
- Fecha de renovación (Próximos 7/15/30 días)
- Actividad (Activos, Inactivos >7 días)
- Lifetime Value (Alto, Medio, Bajo)
- Onboarding (Completado, Pendiente)
- Categoría de negocio

Métricas del Vendedor

MIS MÉTRICAS
Período: [Este mes ▾]

VENTAS

- Nuevas suscripciones: 15
- MRR generado: \$6,735
- Meta del mes: 20 (75%)
- Tasa de conversión: 45% (15/33 demos)

RENOVACIONES

- Renovaciones exitosas: 12/13 (92%)
- Churn: 1 cliente (8%)
- Por renovar esta semana: 3

CARTERA

- Clientes totales: 15
- Clientes activos: 14
- Lifetime Value promedio: \$449

COMISIONES

- Este mes: \$1,010.25 (15% de \$6,735)
- Año acumulado: \$1,010.25
- Próximo pago: 1 Ene 2025

TENDENCIA
[Gráfico de líneas - últimos 6 meses]

Niveles de Acceso

Nivel 1 - Vendedor Junior

Permisos:

MI CARTERA:

- Ver mis clientes asignados
- Ver métricas de mis clientes
- Agregar notas
- Contactar clientes
- NO puede ver otros vendedores

REPORTES:

- Ver mis métricas personales
- Ver mis comisiones
- NO puede ver métricas globales

ACCIONES:

- NO puede editar datos de clientes

- └─ ✕ NO puede reasignar clientes
- └─ ✕ NO puede acceder a configuración

Nivel 2 - Vendedor Senior

Permisos:

Nivel 1 +

MI CARTERA:

- └─ ✓ Editar información de contacto
- └─ ✓ Ver historial completo de pagos
- └─ ✓ Enviar recordatorios automáticos

REPORTES:

- └─ ✓ Ver comparativa con otros vendedores
- └─ ✓ Exportar reportes de cartera

ACCIONES:

- └─ ✓ Solicitar reasignación de clientes
- └─ ✕ NO puede acceder a panel global

Nivel 3 - Líder de Ventas

Permisos:

Nivel 2 +

EQUIPO:

- └─ ✓ Ver todos los vendedores
- └─ ✓ Ver carteras de su equipo
- └─ ✓ Reasignar clientes entre vendedores
- └─ ✓ Ver métricas del equipo

REPORTES:

- └─ ✓ Métricas globales de ventas
- └─ ✓ Embudo de conversión
- └─ ✓ Proyecciones de MRR

ACCIONES:

- └─ ✓ Crear metas para vendedores
- └─ ✕ NO puede modificar configuración global

Nivel 4 - Admin General

Permisos:

Nivel 3 +

GESTIÓN:

- └─ Gestionar todos los negocios
- └─ Gestionar todos los usuarios
- └─ Ver y modificar suscripciones
- └─ Acceso a todos los módulos

REPORTES:

- └─ Métricas globales completas
- └─ Analytics avanzado
- └─ Exportar todo

ACCIONES:

- └─ Despublicar/publicar negocios
- └─ Suspender/activar cuentas
- └─ NO puede modificar permisos ni configuración

Nivel 5 - SuperAdmin (TÚ)

Permisos:

SIN RESTRICCIONES

GESTIÓN TOTAL:

- └─ TODO lo que pueden hacer los demás niveles
- └─ Crear/editar cuentas admin
- └─ Asignar niveles de acceso
- └─ Definir permisos por nivel
- └─ Ver actividad de todos los admins

CONFIGURACIÓN DINÁMICA:

- └─ Modificar parámetros de la app (sin código)
- └─ Activar/desactivar features
- └─ Configurar precios
- └─ Gestionar categorías
- └─ Modificar textos y labels

REPORTES AVANZADOS:

- └─ Dashboards personalizados
- └─ Métricas financieras completas
- └─ Proyecciones y forecasting
- └─ Exportar base de datos completa

AUDITORÍA:

- └─ Ver log de acciones de admins
- └─ Historial de cambios
- └─ Tracking de actividad sospechosa

Panel SuperAdmin

Dashboard Principal

SUPERADMIN - PANEL DE CONTROL
[Cerrar sesión] [Configuración] [Analytics]

MÉTRICAS GLOBALES (Diciembre 2024)

Negocios	MRR	Usuarios	Churn
234 activos	\$105,066	12,450	3.2%
+15 mes	+\$6,735	+890	-0.5%

EQUIPO DE VENTAS (5 vendedores)

- Top vendedor: María G. (23 cuentas, \$10,327)
- Promedio por vendedor: 47 cuentas
- Meta del mes: 80% alcanzada

ALERTAS CRÍTICAS:

- 12 suscripciones vencen en 24 horas
- 3 pagos rechazados requieren atención
- Sistema funcionando correctamente

[Ver reportes completos] [Gestionar vendedores]

Gestión de Vendedores

GESTIÓN DE VENDEDORES
[+ Nuevo vendedor] [Configurar comisiones]

María González - Nivel 3 (Líder)
maria@anunciaya.com • Activa desde: Ene 2024

Métricas:

- Cartera: 23 clientes
- MRR generado: \$10,327
- Tasa retención: 95%
- Comisiones del mes: \$1,549.05

[Ver detalle] [Editar nivel] [Ver actividad]

Configuración de Permisos por Nivel

CONFIGURAR NIVEL 2 - Vendedor Senior

CARTERA DE CLIENTES:

- Ver clientes asignados
- Ver métricas de clientes
- Editar información de contacto
- Reasignar clientes
- Ver clientes de otros vendedores

REPORTES:

- Ver métricas personales
- Ver comparativa con equipo
- Ver métricas globales
- Exportar base de datos

GESTIÓN:

- Gestionar negocios
- Gestionar usuarios finales
- Modificar suscripciones
- Acceder a configuración

[Guardar cambios] [Cancelar] [Restaurar defaults]

Configuración Dinámica de la App

CONFIGURACIÓN DINÁMICA - SIN CÓDIGO

PRECIOS Y PLANES

Plan Comercial: \$[449] MXN/mes
Plan Anual (próximamente): \$[____] MXN/año
Descuento anual: [__]%

FEATURES TOGGLS

- Marketplace habilitado
- Rifas habilitadas
- Bolsa de trabajo habilitada
- ChatYA habilitado
- Pagos en app (próximamente)
- Delivery integrado (beta)

CATEGORÍAS DE NEGOCIOS

[Gestionar categorías y subcategorías]

LÍMITES Y RESTRICCIONES

- Máx. productos por negocio: [____] (0 = ilimitado)

- Máx. servicios por negocio: [____] (0 = ilimitado)
- Máx. empleados por negocio: [____]
- Máx. sucursales por negocio: [____]
- Máx. imágenes por artículo: [5]
- Máx. imágenes en galería: [10]
- Máx. subcategorías por negocio: [3]

⌚ SISTEMA DE PUNTOS (CardYA)

- Umbral Bronce → Plata: [5,000] puntos lifetime
- Umbral Plata → Oro: [15,000] puntos lifetime
- Multiplicador Bronce: [1.0]x
- Multiplicador Plata: [1.25]x
- Multiplicador Oro: [1.5]x
- Expiración QR CardYA: [2] minutos

📝 TEXTOS Y LABELS

[Editar textos de la interfaz]

[Guardar cambios]

Reportes y Métricas SuperAdmin

📊 REPORTES AVANZADOS

💰 FINANCIEROS

- MRR actual: \$105,066
- Proyección 3 meses: \$127,500
- ARR (Annual Run Rate): \$1,260,792
- LTV promedio: \$2,245
- CAC promedio: \$125
- LTV/CAC ratio: 18x

👤 USUARIOS

- Total usuarios: 12,450
- Modo Personal: 12,216
- Modo Comercial: 234
- Tasa de conversión: 1.9%

📈 CRECIMIENTO

[Gráficos avanzados]

- MRR growth rate: 6.4%
- User growth rate: 7.2%
- Churn rate: 3.2%

⌚ POR CATEGORÍA

Top categorías por MRR:

1. Restaurantes: \$32,450 (65 negocios)
2. Servicios: \$18,780 (42 negocios)
3. Retail: \$15,230 (34 negocios)

SISTEMA DE LEALTAD (CardYA)

- Total puntos otorgados (global): X
- Usuarios por nivel: Bronce X / Plata X / Oro X
- Negocios con CardYA activo: X de Y
- Promedio puntos por transacción: X

CUPONES Y OFERTAS

- Cupones creados este mes: X
- Cupones canjeados: X
- Tasa de canje: X%
- Ofertas activas: X

DINÁMICAS

- Rifas activas: X
- Rifas completadas este mes: X
- Participantes totales: X

EMPLEOS

- Vacantes activas: X
- Aplicaciones este mes: X
- Vacantes cerradas: X

[Exportar Excel] [Exportar PDF] [Personalizar dashboard]

Log de Actividad de Admins

AUDITORÍA - ACTIVIDAD DE ADMINS

[Hoy] [Semana] [Mes] [Personalizado]

2:45 PM • María González (Nivel 3)

└ Reasignó cliente "Pizzería Roma" a Carlos M.

1:30 PM • Carlos Martínez (Nivel 2)

└ Agregó nota a cliente "Café Central"

10:15 AM • Admin General (Nivel 4)

└ Suspenderá negocio "Tienda XYZ" (falta de pago)

9:00 AM • TÚ (SuperAdmin)

└ Modificó precio de Plan Comercial a \$449

[Exportar log] [Filtrar por usuario] [Filtrar por acción]

Módulos del Panel Admin

1. Negocios ([/admin/negocios](#))

Funciones: |— Ver todos los negocios (tabla paginada) |— Filtrar por estado (activo, suspendido, etc.) |— Buscar por nombre/categoría |— Ver detalle completo del negocio |— Ver sucursales del negocio |— Estadísticas por sucursal |— Activar/desactivar sucursal específica |— Editar información |— Suspender/reactivar negocio |— Ver actividad (logins, ventas, etc.) |— Reasignar vendedor |— Exportar listado

2. Usuarios (`/admin/usuarios`)

Funciones: |— Ver todos los usuarios |— Filtrar por modo (Personal/Comercial) |— Ver perfil completo |— Cambiar modo de cuenta |— Ver historial de suscripciones |— Ver actividad en la app |— Suspender/reactivar cuenta |— Exportar base de usuarios

3. Suscripciones y Pagos (`/admin/suscripciones`)

Funciones: |— Ver todas las suscripciones |— Estado (activas, vencidas, canceladas) |— Próximas a vencer (alertas) |— Pagos rechazados |— Gráfica de MRR histórico |— Proyecciones de ingresos |— Procesar reembolso manual |— Exportar reporte financiero |— Reintentar pagos fallidos |— Webhook logs (eventos de Stripe) Δ Nota: Reembolsos y ajustes complejos se procesan en el Dashboard de Stripe directamente.

4. Vendedores (`/admin/vendedores`)

Solo SuperAdmin

Funciones: |— Ver todos los vendedores |— Crear nuevo vendedor |— Asignar nivel de acceso |— Ver métricas por vendedor |— Reasignar carteras |— Configurar comisiones |— Ver log de actividad |— Suspender/activar cuenta

5. Configuración (`/admin/config`)

Solo SuperAdmin

Funciones: ┌── Configurar precios ┌── Activar/desactivar features ┌── Gestionar categorías ┌──
Configurar límites ┌── Editar textos de la app ┌── Configurar integraciones ┌── Gestionar permisos por
nivel └── Backup de configuración

6. Reportes (`/admin/reportes`)

Acceso según nivel: ┌── Vendedores: Solo sus métricas ┌── Líderes: Métricas del equipo ┌── Admin:
Métricas globales └── SuperAdmin: Todo + exportación completa

📄 Modelo de Datos

```
```typescript
interface AdminUser {
 id: string;
 email: string;
 password: string; // Hasheado
 nombre: string;
 nivel: 1 | 2 | 3 | 4 | 5; // 1-3: Vendedores, 4: Admin, 5: SuperAdmin
 activo: boolean;

 // Solo vendedores
 carteraClientes?: string[]; // IDs de negocios asignados
 metasMensuales?: {
 nuevasCuentas: number;
 renovaciones: number;
 };
 comisiones?: {
 porcentaje: number; // 15% por defecto
 acumulado: number;
 };
}

createdAt: Date;
ultimoAcceso?: Date;
}

interface ClienteVendedor {
 negocioId: string;
 vendedorId: string;
 fechaAsignacion: Date;
 estado: 'activo' | 'por_renovar' | 'vencido' | 'cancelado';
 notas: Array<{
 fecha: Date;
 vendedorId: string;
 texto: string;
 }>;
}
```

```
recordatorios: Array<{
 fecha: Date;
 tipo: 'renovacion' | 'seguimiento' | 'otro';
 enviado: boolean;
}>;
}

interface ConfiguracionApp {
 precios: {
 planComercial: number;
 planAnual?: number;
 descuentoAnual?: number;
 };
 features: {
 marketplace: boolean;
 rifas: boolean;
 bolsaTrabajo: boolean;
 pagosEnApp: boolean;
 chatYA: boolean;
 };
 limites: {
 maxProductos: number;
 maxServicios: number;
 maxEmpleados: number;
 maxSucursales: number;
 maxImagenesPorArticulo: number;
 maxImagenesGaleria: number;
 maxSubcategorias: number;
 };
 cardYA: {
 umbralPlata: number; // 5000
 umbralOro: number; // 15000
 multiplicadorBronce: number; // 1.0
 multiplicadorPlata: number; // 1.25
 multiplicadorOro: number; // 1.5
 expiracionQRMinutos: number; // 2
 };
 textos: {
 [key: string]: string;
 };
}

interface LogAdmin {
 id: string;
 adminId: string;
 accion: string;
 modulo: string;
 detalles: any;
 timestamp: Date;
}
```

## 🚀 FASE 7: Lanzamiento + Operaciones

### 7.1 Pre-Lanzamiento

- Testing E2E
- Performance
- SEO
- Analytics
- Sentry

### 7.2 Infraestructura Producción

- Vercel (Frontend)
- Railway (Backend + PostgreSQL)
- MongoDB Atlas
- Upstash (Redis)
- Stripe → **Modo LIVE**

### 7.3 Beta (50 negocios)

1. Embajadores registran pilotos
2. Trial de 7 días (modo comercial gratis)
3. Capacitación
4. Feedback
5. Iteración

---

## 📋 PARTE 3: ORDEN DE DESARROLLO

---

### FASE 4 COMPLETADA

└─ ColumnaDerecha (rediseño)

---

### FASE 5 - Bloque 1: Fundamentos COMPLETADO

- └─ 5.0 Sistema de Modos - Backend  COMPLETADO
- └─ 5.1 Onboarding Backend + Sucursales  COMPLETADO
  - └─ 15 endpoints REST
  - └─ Sistema de sucursales
  - └─ Migración BD
- └─ 5.1.0 Estandarización Nomenclatura  COMPLETADO
  - └─ Drizzle casing automático
  - └─ API responses en inglés (439 cambios)
- └─ 5.1.1 Onboarding Frontend  COMPLETADO (26/12/2024)
  - └─ Layout + componentes base

- └─ Paso 1-8 completos ✓
  - └─ Sistema de Finalización ✓
  - └─ Cloudinary (imágenes) ✓
- 
- └─ 5.2 Toggle UI + Protección Rutas ✓ COMPLETADO (26/12/2024)
    - └─ Componente ToggleModoUsuario ✓
    - └─ ModalCambiarModo ✓
    - └─ ColumnaIzquierda dinámica ✓
    - └─ MenuDrawer dinámico ✓
    - └─ BottomNav dinámico ✓
    - └─ ModoGuard (protección rutas) ✓
    - └─ Datos dinámicos del negocio ✓
- 
- └─ 5.3 Negocios Directorio ✓ COMPLETADO (02/01/2026)
    - └─ Lista con geolocalización (PostGIS) ✓
    - └─ Filtros por categoría/subcategoría ✓
    - └─ Búsqueda por nombre ✓
    - └─ Vista mapa con marcadores ✓
    - └─ Perfil de negocio (galería, horarios, catálogo) ✓
    - └─ Sistema de votos y métricas ✓
- 
- └─ 5.3.1 Sistema Compartir (base) ✓ PARCIAL (02/01/2026)
    - └─ DropdownCompartir.tsx ✓
    - └─ BannerRegistro.tsx ✓
    - └─ LayoutPublico.tsx ✓
    - └─ useOpenGraph.ts ✓
    - └─ /p/negocio/:id ✓
    - └─ /p/articulo/:id ✓
    - └─ /p/oferta/:id ✓
- 
- └─ 5.3.2 Auth Opcional + ModalAuthRequerido | ✓ 100% (16/01/2026)

---

## FASE 5 - Bloque 2: Business Studio 📈 EN PROGRESO (15%)

- └─ 5.4 Business Studio - Layout y Base ✓ COMPLETADO (02/01/2026)
    - └─ DrawerBusinessStudio.tsx ✓
    - └─ MenuBusinessStudio.tsx ✓
    - └─ PanelPreviewNegocio.tsx ✓
    - └─ Router con 14 rutas ✓
    - └─ Interceptor Axios (sucursalId automático) ✓
    - └─ negocioManagement.service.ts (15 funciones CRUD) ✓
- 
- └─ 5.4 Dashboard ✓ COMPLETADO (02/01/2026)
    - └─ KPIs principales y secundarios ✓
    - └─ Gráfica de ventas ✓
    - └─ Actividad reciente ✓
    - └─ 7 endpoints backend ✓
- 
- └─ 5.4 Mi Perfil ✓ COMPLETADO (06/01/2026)

- └─ Tab Datos del Negocio + Panel CardYA ✓
- └─ Tab Contacto (teléfono, WhatsApp, redes) ✓
- └─ Tab Ubicación (mapa Leaflet) ✓
- └─ Tab Horarios (con break/comida) ✓
- └─ Tab Imágenes (logo, portada, galería) ✓
- └─ Tab Operación (pagos, envío, servicio) ✓
  
- └─ 5.4.1 Catálogo CRUD ✓ COMPLETADO (10/01/2026)
  - └─ Lista de productos/servicios
  - └─ Modal crear/editar artículo
  - └─ Upload imágenes (Cloudinary)
  - └─ Filtros por tipo y categoría
  - └─ Toggle activo/inactivo
  
- └─ 5.4.2 Ofertas CRUD ✓ COMPLETADO 90% (16/01/2026)
  - └─ Lista de ofertas del negocio
  - └─ Modal crear/editar oferta
  - └─ Tipos: 2x1, %, \$, combo, happy hour
  - └─ Configurar días y horarios
  - └─ Vista previa pública

## FASE 5 - Bloque 3: Sistema de Lealtad (~1-2 semanas)

- └─ 5.5 ScanYA + PWA ✎ PENDIENTE (~2-3 días) | └─ Escanear QR de clientes | └─ Registrar ventas |
- └─ Otorgar puntos automáticos | └─ Validar cupones | └─ Login empleados (nick + PIN) | └─ PWA widget standalone |
- └─ 5.6 CardYA + PWA ✎ PENDIENTE (~2-3 días) → depende de 5.5 | └─ Tarjeta de lealtad digital |
- └─ QR dinámico (expira 2 min) | └─ Puntos por negocio | └─ Niveles Bronce/Plata/Oro |
- └─ Multiplicadores (1x, 1.25x, 1.5x) | └─ PWA widget standalone | └─ 5.7 Clientes + Transacciones ✎ PENDIENTE (~2-3 días) → depende de 5.5 | └─ Lista de clientes que han comprado | └─ Historial de transacciones |
- └─ Filtros por fecha/cliente | └─ Detalle por cliente (visitas, puntos, nivel) | └─ 5.8 Opiniones (BS + ScanYA) ✎ PENDIENTE (~2-3 días) → depende de 5.5 + 5.7 | └─ Ver reseñas con calificación (☆ 1-5) | └─ Dashboard de métricas (promedio, total, tasa respuesta) | └─ Filtros (calificación, estado, sucursal, fecha) | └─ Responder reseñas desde BS (web) | └─ Responder reseñas desde ScanYA (móvil) | └─ Templates de respuesta pre-escritos | └─ Editar/eliminar respuestas propias |
- └─ Reportar reseñas inapropiadas | └─ Badge "Compra verificada" (valida transacciones) | └─ Sistema de permisos por empleado | └─ Notificaciones push de nuevas reseñas | └─ Atribución de respuestas (dueño/gerente/empleado) | └─ Vista móvil simplificada en ScanYA | └─ Límite: 1 reseña cada 30 días por cliente | └─ Estadísticas: distribución calificaciones + tiempo respuesta | └─ 5.9 Puntos (Config BS) ✎ PENDIENTE (~1 día) → depende de 5.6 | └─ Valor del punto (\$X = 1 punto) | └─ Activar/desactivar CardYA | └─ Simulador de acumulación └─ Estadísticas de puntos otorgados

### **Nota importante sobre 5.8 Opiniones:**

- **Requiere obligatoriamente:** Tabla **transacciones** de 5.5 ScanYA para validar compras
- **Requiere obligatoriamente:** Sistema de clientes de 5.7 para verificar historial
- **Doble acceso:** Business Studio (web) + ScanYA (PWA móvil)
- **Validación:** Solo clientes con compras verificadas pueden reseñar
- **Multi-usuario:** Dueños, gerentes y empleados autorizados pueden responder

- **Trazabilidad:** Sistema registra quién respondió y desde dónde (BS/ScanYA)
- 

## FASE 5 - Bloque 4: Comunicación y Cupones (~1 semana)

- 5.10 ChatYA + PWA PENDIENTE (~3-4 días)
  - Socket.io + MongoDB
  - Identidad según modo (personal/comercial)
  - Overlay persistente (no ruta dedicada)
  - PWA widget standalone
- 5.11 Cupones PENDIENTE (~2-3 días) → depende de 5.5
  - Vista pública: /p/cupon/:codigo
  - CRUD cupones en BS
  - Tipos: %, \$, 2x1, regalo
  - Límites de uso
  - Validación en ScanYA

---

## FASE 6 - Secciones Públicas + BS Restantes (~2-3 semanas)

- 6.0 Ofertas Públicas PENDIENTE (~1-2 días) → depende de 5.4.2
  - Vista pública lista (ruta ya existe)
- 6.1 MarketPlace PENDIENTE (~3-4 días)
  - Vista pública: /p/marketplace/:id
  - Compra-venta entre usuarios
  - Publicaciones modo Personal
- 6.2 Dinámicas PENDIENTE (~3-4 días)
  - Vista pública: /p/rifa/:id
  - Rifas y sorteos
  - Participación con puntos
  - CRUD Rifas en BS
- 6.3 Empleos PENDIENTE (~2-3 días)
  - Vista pública: /p/empleo/:id
  - Bolsa de trabajo
  - CRUD Vacantes en BS
- 6.4 Empleados (BS) PENDIENTE (~1-2 días) → depende de 5.5
  - Gestión de empleados
  - Nick + PIN para ScanYA
  - Permisos configurables
- 6.5 Sucursales (BS) PENDIENTE (~2 días)
  - Ver todas las sucursales
  - Agregar/editar sucursal
  - Configurar sucursal principal

- └── 6.6 Reportes + Alertas (BS) PENDIENTE (~2-3 días) → depende de 5.5
  - ├── Estadísticas de ventas
  - ├── Exportar reportes
  - └── Notificaciones del sistema
  
- └── 6.7 Panel Admin PENDIENTE (~1-2 semanas)
  - ├── Gestión de vendedores (niveles 1-5)
  - ├── Métricas globales
  - ├── Configuración dinámica (sin código)
  - ├── Gestión de negocios/usuarios
  - ├── Suscripciones y pagos (Stripe)
  - └── Log de auditoría

## FASE 7 (~1 semana)

- └── Testing y optimización
- └── Infraestructura producción
- └── Variables de entorno (Railway/Vercel)
- └── Stripe modo live
- └── Dominio personalizado + SSL
- └── Beta 50 negocios

## SISTEMA DE COMPARTIR CONTENIDO

Cada sección incluye su ruta pública correspondiente:

Fase	Sección	Ruta Pública	Estado
5.3	Negocios	/p/negocio/:id	Implementado
5.4.1	Artículos	/p/articulo/:id	Implementado
5.4.2	Ofertas	/p/oferta/:id	Implementado
5.10	Cupones	/p/cupon/:codigo	Pendiente
6.1	MarketPlace	/p/marketplace/:id	Pendiente
6.2	Dinámicas	/p/rifa/:id	Pendiente
6.3	Empleos	/p/empleo/:id	Pendiente

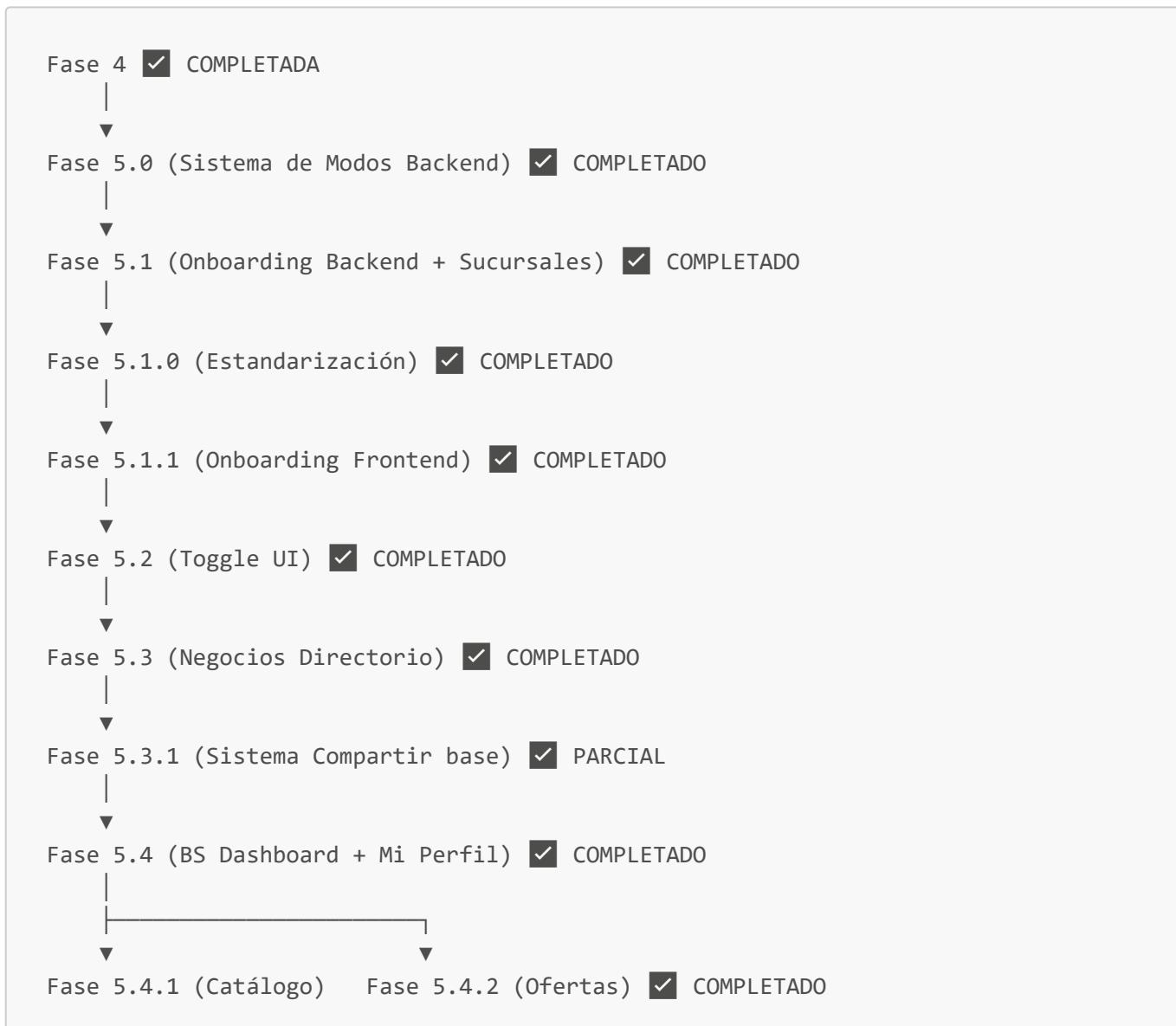
Cada vista pública incluye:

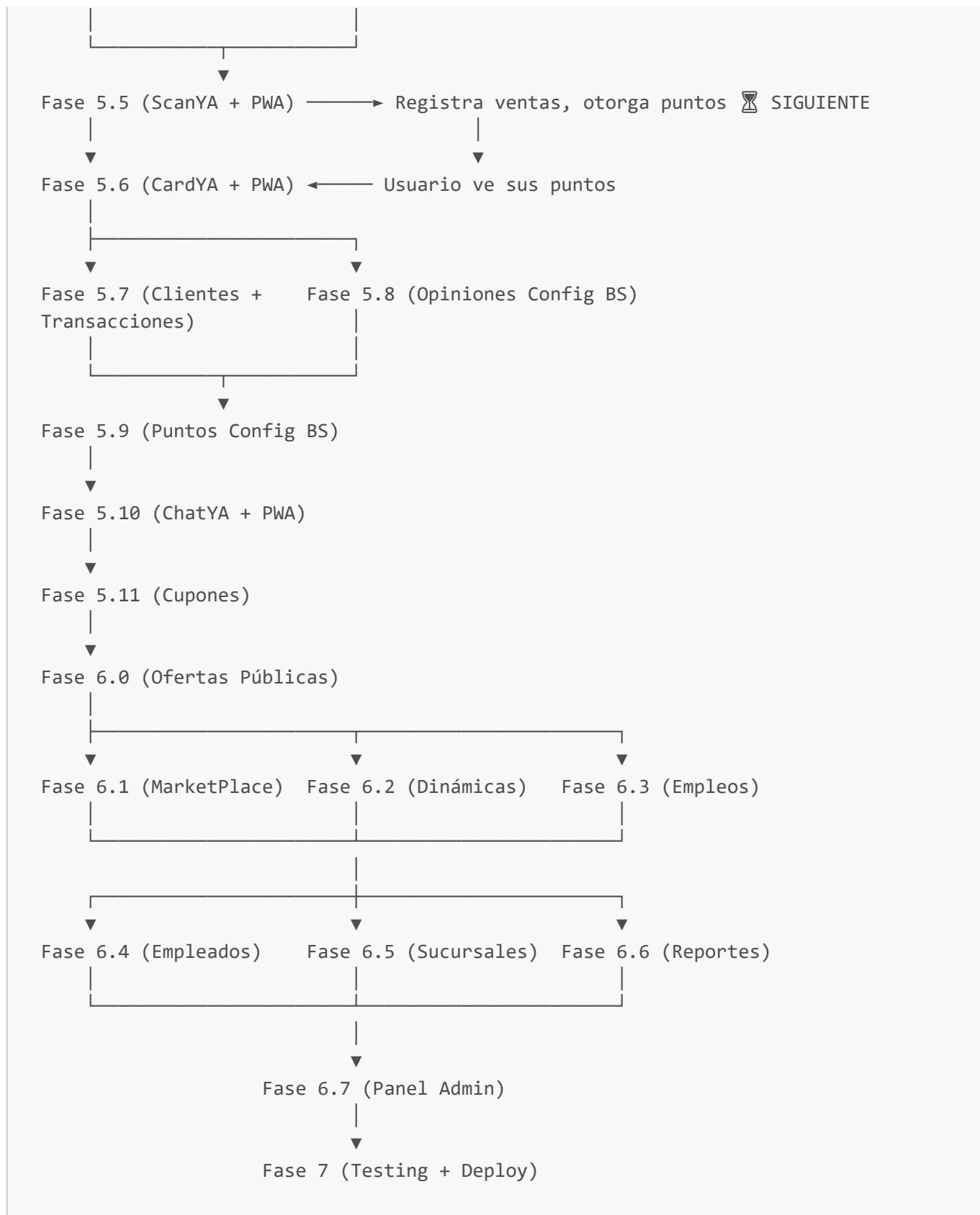
- Contenido completo visible
- Imágenes y galería
- Información del publicador
- Botón de registro/descarga de app
- Metadatos Open Graph (preview en redes sociales)

## ⌚ TIEMPO TOTAL RESTANTE ESTIMADO

Bloque	Tiempo
5.4.1 + 5.4.2 (Catálogo + Ofertas)	~2-4 días
5.5 - 5.8 (Sistema de Lealtad)	~7-10 días
5.9 - 5.10 (ChatYA + Cupones)	~5-7 días
6.0 - 6.3 (Secciones Públicas)	~8-11 días
6.4 - 6.6 (BS Restantes)	~5-7 días
6.7 (Panel Admin)	~1-2 semanas
Fase 7 (Testing + Deploy)	~1 semana
<b>TOTAL</b>	<b>~8-10 semanas</b>

## ⌚ PARTE 4: DEPENDENCIAS





## Resumen de Dependencias Críticas

Fase	Depende de	Bloquea a
5.5 ScanYA	5.4.1, 5.4.2	5.6, 5.7, 5.10, 6.4, 6.6
5.6 CardYA	5.5 ScanYA	5.9 Puntos
5.7 Clientes	5.5 ScanYA	-

Fase	Depende de	Bloquea a
5.8 Opiniones	5.5 ScanYA	-
5.9 Puntos	5.6 CardYA	-
5.11 Cupones	5.5 ScanYA	-
6.0 Ofertas Públicas	5.4.2 Ofertas	-
6.4 Empleados	5.5 ScanYA	-
6.6 Reportes	5.5 ScanYA	-

### ⚠ Ruta Crítica

```
5.4.1/5.4.2 → 5.5 ScanYA → 5.6 CardYA → 5.8 Opiniones → 5.9 Puntos
↓
5.7 Clientes
5.11 Cupones
6.4 Empleados
6.6 Reportes
```

**ScanYA (5.5) es el cuello de botella principal** - bloquea 6 fases posteriores.

---

#### ## 📋 PARTE 5: CHECKLISTS DE VALIDACIÓN

---

##### ### Estandarización ✓ COMPLETADO

- [x] Drizzle `casing: 'snake\_case'` configurado
- [x] Transformación automática snake\_case ↔ camelCase
- [x] API responses en inglés (success, data, message)
- [x] 439 cambios aplicados
- [x] 0 errores TypeScript
- [x] Backend compila correctamente
- [x] Frontend compila correctamente

---

##### ### Sistema de Modos ✓ COMPLETADO

- [x] Migración BD ejecutada
- [x] Campo `tieneModoComercial` funciona
- [x] Campo `modoActivo` funciona
- [x] Endpoint PATCH `/api/auth/modo` funciona
- [x] Endpoint GET `/api/auth/modo-info` funciona
- [x] Middleware `requiereModoPersonal` funciona
- [x] Middleware `requiereModoComercial` funciona

- [x] Middleware `requiereAccesoComercial` funciona
- [x] Token JWT incluye `modoActivo`
- [x] Webhook cancelación Stripe implementado
- [x] useAuthStore tiene `cambiarModo()` ✓
- [x] Toggle UI funciona ✓
- [x] ColumnaIzquierda cambia según modo ✓
- [x] MenuDrawer cambia según modo ✓
- [x] BottomNav cambia según modo ✓
- [x] Navbar oculta Market en modo Comercial ✓
- [x] ModoGuard protege rutas ✓
- [x] Datos dinámicos del negocio (nombre, correo, foto) ✓

---

### ### Toggle UI + Protección de Rutas ✓ COMPLETADO (26/12/2024)

#### \*\*Frontend:\*\*

- [x] ToggleModoUsuario.tsx - Componente reutilizable
- [x] ModalCambiarModo.tsx - Confirmación de cambio
- [x] ModoGuard.tsx - Guard de protección de rutas
- [x] Navbar.tsx - Toggle + NAV\_ITEMS dinámicos
- [x] MenuDrawer.tsx - Toggle + avatar dinámico
- [x] ColumnaIzquierda.tsx - Contenido por modo
- [x] BottomNav.tsx - Market ↔ Business según modo
- [x] Avatar dinámico según modo (usuario o negocio)

#### \*\*Backend:\*\*

- [x] Campo `foto\_perfil` en negocio\_sucursales
- [x] Función `obtenerDatosNegocio()` reutilizable
- [x] Campos de negocio en respuestas de auth
- [x] PATCH /api/auth/modo devuelve nuevo token

#### \*\*Decisiones Arquitectónicas Implementadas:\*\*

- Login respeta último modo usado (guardado en BD)
- Multi-dispositivo: Sesiones independientes
- Notificaciones: Solo del modo activo
- JWT: Nuevo token al cambiar modo
- Persistencia: localStorage + backend siempre gana

---

### ### Onboarding Wizard ✓ COMPLETADO (26/12/2024)

#### \*\*Backend:\*\*

- [x] 15 endpoints REST implementados
- [x] Sistema de sucursales
- [x] Migración BD ejecutada
- [x] Validaciones Zod
- [x] TypeScript compila sin errores
- [x] Endpoints de imágenes (logo, portada, galería)
- [x] Endpoint finalizar onboarding

#### \*\*Frontend:\*\*

- [x] Layout estilo Stripe

- [x] Indicador dinámico de pasos (8 pasos)
- [x] Botones Navegación con validación
- [x] Modal Pausar con guardado
- [x] useOnboardingStore completo
- [x] Paso 1 (Categorías) completo
- [x] Paso 2 (Ubicación) completo
- [x] Paso 3 (Contacto + Lada Editable) completo
- [x] Paso 4 (Horarios) completo
- [x] Paso 5 (Imágenes + Cloudinary) completo
- [x] Paso 6 (Métodos de Pago) completo
- [x] Paso 7 (Sistema de Puntos) completo
- [x] Paso 8 (Productos/Servicios) completo
- [x] Redirección según onboardingCompletado
- [x] JWT incluye onboardingCompletado
- [x] Cloudinary upload/delete optimista
- [x] Optimización automática a .webp
- [x] Upload diferido (sin huérfanos)
- [x] Sistema de finalización funcional
- [x] Botón Anterior guarda cambios
- [x] Validación flexible (1 guardar, 3 publicar)

**\*\*Decisiones Implementadas:\*\***

- Todos los negocios son físicos (requieren ubicación)
- Campos `tiene\_servicio\_domicilio` y `tiene\_envio\_domicilio` en sucursales
- Optimización client-side antes de Cloudinary
- Upload diferido para evitar huérfanos

---

### ### Negocios Directorio COMPLETADO (02/01/2026)

**\*\*Frontend:\*\***

- [x] PaginaNegocios.tsx - Lista principal
- [x] PaginaPerfilNegocio.tsx - Detalle de negocio
- [x] PanelFiltros.tsx - Filtros por categoría
- [x] ModalDetalleItem.tsx - Detalle de producto/servicio
- [x] ModalHorarios.tsx - Horarios del negocio
- [x] SeccionCatalogo.tsx - Productos y servicios
- [x] useListaNegocios.ts - Hook de listado
- [x] usePerfilNegocio.ts - Hook de perfil
- [x] useHorariosNegocio.ts - Hook de horarios
- [x] useVotos.ts - Hook de votación

**\*\*Backend:\*\***

- [x] Endpoints de negocios con PostGIS
- [x] Sistema de votos y métricas
- [x] Filtros por categoría/subcategoría
- [x] Búsqueda por nombre

---

### ### Sistema Compartir PARCIAL (02/01/2026)

**\*\*Implementado:\*\***

- [x] DropdownCompartir.tsx - Botón universal
- [x] BannerRegistro.tsx - CTA para registro
- [x] LayoutPublico.tsx - Layout sin login
- [x] useOpenGraph.ts - Metadatos sociales
- [x] /p/negocio/:id - Vista pública negocio
- [x] /p/articulo/:id - Vista pública artículo
- [x] /p/oferta/:id - Vista pública oferta

\*\*Pendiente:\*\*

- [ ] /p/cupon/:codigo (Fase 5.10)
- [ ] /p/marketplace/:id (Fase 6.1)
- [ ] /p/rifa/:id (Fase 6.2)
- [ ] /p/empleo/:id (Fase 6.3)
- [ ] Tracking de vistas
- [ ] Deep linking a la app

---

### ### Business Studio EN PROGRESO (27%)

\*\*Dashboard  COMPLETADO (02/01/2026):\*\*

- [x] PaginaDashboard.tsx
- [x] HeaderDashboard.tsx
- [x] KPIPrincipal.tsx + KPISecundario.tsx
- [x] GraficaVentas.tsx
- [x] ActividadReciente.tsx
- [x] 7 endpoints backend

\*\*Mi Perfil  COMPLETADO (06/01/2026):\*\*

- [x] Tab Datos del Negocio + Panel CardYA
- [x] Tab Contacto (teléfono, WhatsApp, redes)
- [x] Tab Ubicación (mapa Leaflet)
- [x] Tab Horarios (con break/comida)
- [x] Tab Imágenes (logo, portada, galería 10 máx)
- [x] Tab Operación (pagos, envío, servicio)

\*\*Catálogo  COMPLETADO (10/01/2026):\*\*

- [x] Lista de productos/servicios
- [x] Modal crear/editar artículo
- [x] Upload imágenes (Cloudinary)
- [x] Filtros por tipo y categoría
- [x] Toggle activo/inactivo

\*\*Ofertas  COMPLETADO 90/100% (16/01/2026):\*\*-5.4.2

- [x] Lista de ofertas del negocio
- [x] Modal crear/editar oferta
- [x] Tipos: 2x1, %, \$, combo, happy hour
- [x] Configurar días y horarios
- [x] Vista previa pública

---

### ### ScanYA PENDIENTE (5.5)

- [ ] Escanear QR de clientes

- [ ] Registrar ventas
- [ ] Otorgar puntos automáticos
- [ ] Validar cupones
- [ ] Login empleados (nick + PIN)
- [ ] Historial de ventas del día
- [ ] PWA widget standalone

---

### ### CardYA con Sistema de Niveles PENDIENTE (5.6)

- [ ] Tarjeta de lealtad digital
- [ ] QR dinámico (expira cada 2 min)
- [ ] Puntos por negocio (no transferibles)
- [ ] Bronce (0-4,999) → 1.0x
- [ ] Plata (5,000-14,999) → 1.25x
- [ ] Oro (15,000+) → 1.5x
- [ ] Solo visible en modo Personal
- [ ] Notificación al subir de nivel
- [ ] PWA widget standalone

---

### ### Clientes + Transacciones PENDIENTE (5.7)

- [ ] Lista de clientes que han comprado
- [ ] Historial de transacciones
- [ ] Filtros por fecha/cliente
- [ ] Detalle por cliente (visitas, puntos, nivel)
- [ ] Exportar reportes

---

### ### Opiniones Config BS PENDIENTE (5.9)

### ### Puntos Config BS PENDIENTE (5.9)

- [ ] Valor del punto (\$X = 1 punto)
- [ ] Activar/desactivar CardYA
- [ ] Simulador de acumulación
- [ ] Estadísticas de puntos otorgados

---

### ### ChatYA PENDIENTE (5.10)

- [ ] Socket.io + MongoDB
- [ ] No se cierra al navegar (overlay persistente)
- [ ] Socket.io permanece conectado
- [ ] Muestra nombre/avatar según modo activo
- [ ] PWA widget standalone

---

- ### Cupones ☰ PENDIENTE (5.11)
- [ ] Vista pública: /p/cupon/:codigo
  - [ ] CRUD cupones en BS
  - [ ] Lista de cupones guardados (usuario)
  - [ ] Filtros: Vigentes / Usados / Expirados
  - [ ] Countdown de expiración
  - [ ] Solo visible en modo Personal
  - [ ] Badge con cantidad en menú
  - [ ] Validación en ScanYA

---

### ### Secciones Públicas ☰ PENDIENTE

#### \*\*6.0 Ofertas Públicas:\*\*

- [ ] Vista pública (ruta ya existe)

#### \*\*6.1 MarketPlace:\*\*

- [ ] Vista pública: /p/marketplace/:id
- [ ] Compra-venta entre usuarios
- [ ] Publicaciones modo Personal

#### \*\*6.2 Dinámicas (Rifas):\*\*

- [ ] Vista pública: /p/rifa/:id
- [ ] Rifas y sorteos
- [ ] Participación con puntos
- [ ] CRUD Rifas en BS

#### \*\*6.3 Empleos:\*\*

- [ ] Vista pública: /p/empleo/:id
- [ ] Bolsa de trabajo
- [ ] CRUD Vacantes en BS

---

### ### BS Restantes ☰ PENDIENTE

#### \*\*6.4 Empleados:\*\*

- [ ] Gestión de empleados
- [ ] Nick + PIN para ScanYA
- [ ] Permisos configurables

#### \*\*6.5 Sucursales:\*\*

- [ ] Ver todas las sucursales
- [ ] Agregar/editar sucursal
- [ ] Configurar sucursal principal

#### \*\*6.6 Reportes + Alertas:\*\*

- [ ] Estadísticas de ventas
- [ ] Exportar reportes
- [ ] Notificaciones del sistema

---

### Panel Admin PENDIENTE (6.7)  
- [ ] Gestión de vendedores (niveles 1-5)  
- [ ] Métricas globales  
- [ ] Configuración dinámica (sin código)  
- [ ] Gestión de negocios/usuarios  
- [ ] Suscripciones y pagos (Stripe)  
- [ ] Log de auditoría

---

### Geolocalización Global  
- [x] Auto-detección GPS con fallback IP/WiFi  
- [ ] Cambio de ciudad actualiza TODAS las secciones

---

## CHANGELOG

---

### 06 Enero 2026 (v5.0)

\*\*Fase 5.4 Mi Perfil COMPLETADO\*\*

\*\*Componentes Creados:\*\*

- PaginaPerfil.tsx - Vista principal con tabs
- TabInformacion.tsx - Nombre, descripción, categorías + Panel CardYA
- TabContacto.tsx - Teléfono, WhatsApp, email, web, redes sociales
- TabUbicacion.tsx - Dirección, ciudad, mapa Leaflet interactivo
- TabHorarios.tsx - Horarios por día, break/comida, 24/7, duplicar
- TabImagenes.tsx - Logo, foto perfil, portada, galería (máx 10)
- TabOperacion.tsx - Métodos pago, envío domicilio, servicio domicilio

\*\*Decisión Arquitectónica - Negocios Solo Físicos:\*\*

- Eliminado tipo "Online" - Todos los negocios requieren ubicación
- Eliminada columna `requiere\_direccion`
- Agregado `tiene\_servicio\_domicilio` (tú vas al cliente)
- Agregado `tiene\_envio\_domicilio` (envías productos)

\*\*Actualización Roadmap:\*\*

- Nueva numeración de fases (5.4.1 - 6.7)
- Panel Admin movido a 6.7
- Testing + Deploy como Fase 7
- Tiempo estimado restante: ~8-10 semanas

---

### 02-03 Enero 2026 (v4.9)

\*\*Fase 5.3 Negocios Directorio COMPLETADO\*\*

\*\*Frontend Creado:\*\*

- PaginaNegocios.tsx - Lista con geolocalización
- PaginaPerfilNegocio.tsx - Detalle de negocio

- PanelFiltros.tsx - Filtros por categoría/subcategoría
- ModalDetalleItem.tsx - Detalle de producto/servicio
- ModalHorarios.tsx - Horarios del negocio
- SeccionCatalogo.tsx - Productos y servicios
- useListaNegocios.ts, usePerfilNegocio.ts, useVotos.ts

\*\*Backend:\*\*

- Endpoints de negocios con PostGIS (ST\_DWithin)
- Sistema de votos y métricas
- Triggers automáticos para consistencia

---

\*\*Fase 5.3.1 Sistema Compartir PARCIAL\*\*

\*\*Componentes Creados:\*\*

- DropdownCompartir.tsx - Botón universal
- BannerRegistro.tsx - CTA para registro
- LayoutPublico.tsx - Layout sin login
- useOpenGraph.ts - Metadatos sociales

\*\*Rutas Públicas Implementadas:\*\*

- /p/negocio/:id
- /p/articulo/:id
- /p/oferta/:id

---

\*\*Fase 5.4 Business Studio - Dashboard COMPLETADO\*\*

\*\*Frontend Creado:\*\*

- DrawerBusinessStudio.tsx - Navegación lateral
- MenuBusinessStudio.tsx - Menú móvil
- PanelPreviewNegocio.tsx - Preview del negocio
- PaginaDashboard.tsx + 10 componentes KPI

\*\*Backend:\*\*

- dashboard.controller.ts + dashboard.service.ts
- 7 endpoints (kpis, grafica-ventas, actividad-reciente, etc.)

---

\*\*Sistema de Sucursales Múltiples COMPLETADO\*\*

\*\*Implementado:\*\*

- Campo `sucursal\_asignada` en usuarios
- Middleware `verificarNegocio` corregido
- Middleware `validarAccesoSucursal`
- Interceptor Axios (sucursalId automático)
- negocioManagement.service.ts (15 funciones CRUD)
- Tabla `articulo\_sucursales` N:N
- Alertas con `sucursal\_id`

\*\*Decisiones:\*\*

- sucursalActiva = temporal (UI) vs sucursalAsignada = permanente (BD)
- Con 1 sucursal: todo unificado
- Con 2+ sucursales: selector visible + Tab Sucursales

---

### 26 Diciembre 2024 (v4.6)

\*\*Fase 5.2 Toggle UI + Protección de Rutas COMPLETADA\*\*

\*\*Componentes Creados:\*\*

- ToggleModoUsuario.tsx - Toggle reutilizable
- ModalCambiarModo.tsx - Confirmación de cambio
- ModoGuard.tsx - Guard de protección

\*\*Archivos Modificados:\*\*

- useAuthStore.ts - Función `cambiarModo()` + campos de negocio
- Navbar.tsx - Toggle + items dinámicos + avatar dinámico
- MenuDrawer.tsx - Toggle + avatar dinámico
- ColumnaIzquierda.tsx - Contenido por modo
- BottomNav.tsx - Market ↔ Business según modo
- router/index.tsx - Guards aplicados

\*\*Backend:\*\*

- Migración: Campo `foto\_perfil` en negocio\_sucursales
- Función `obtenerDatosNegocio()` en negocios.service.ts
- Datos del negocio incluidos en respuestas de auth

\*\*Decisiones Arquitectónicas:\*\*

- Login respeta último modo usado
- Multi-dispositivo independiente
- Notificaciones solo modo activo
- Nuevo token JWT al cambiar modo

---

\*\*Fase 5.1.1 Frontend Onboarding COMPLETADA\*\*

- Paso 1 (Categorías) completo
- Paso 2 (Ubicación) completo
- Paso 3 (Contacto) - Lada editable internacional
- Paso 4 (Horarios) - 24/7, cerrado, comida
- Paso 5 (Imágenes) - Cloudinary + optimización .webp
- Paso 6 (Métodos de Pago) - Efectivo, Tarjeta, Transferencia
- Paso 7 (Puntos) - Toggle CardYA
- Paso 8 (Productos) - CRUD completo con imágenes
- Sistema de Finalización funcional
- Botón Anterior ahora guarda cambios
- 12 bugs resueltos
- 8 endpoints nuevos creados
- ~5,000 líneas de código

\*\*Bugs Resueltos:\*\*

- PostGIS retornaba WKB binario → Usar ST\_X/ST\_Y
- Lada mostraba 3 dígitos → Función específica por país

- Imágenes huérfanas → Upload diferido
- Error 400 snake\_case → Usar camelCase
- Duplicación productos → DELETE + INSERT
- Finalizar no funcionaba → Lógica completa implementada
- /auth/yo devolvía false → Consultar tabla negocios
- Loop infinito redirección → Flag sessionStorage

\*\*Pendientes Documentados (Fase 6.7 Panel Admin):\*\*

- Endpoint DELETE negocios + limpieza Cloudinary
- Endpoint DELETE usuarios

---

### 21 Diciembre 2024 (v4.4)

- Estandarización Parte 1: Drizzle `casing: 'snake\_case'`
- Estandarización Parte 2: API responses en inglés (439 cambios)
- Onboarding Paso 1 (Categorías) implementado
- Bugs corregidos: redirección según onboardingCompletado
- JWT ahora incluye `onboardingCompletado`
- Rate Limiter ajustado (1000 dev, 100 prod)
- Análisis de arquitectura de sucursales completado

---

### 20 Diciembre 2024 (v4.3)

- Sistema de Modos Backend completado
- Onboarding Backend completado (15 endpoints)
- Sistema de sucursales implementado
- Migración BD ejecutada

---

### 18-19 Diciembre 2024

- Cloudinary upload/delete optimista
- GPS con fallback IP/WiFi
- Actualización BD (42 tablas en 9 esquemas)

---

#  NOTAS IMPORTANTES

---

## Flujo de Registro

\*\*Flujo Actual:\*\*

Registro → Usuario elige tipo (Personal o Comercial) | ↗ Personal → Acceso gratuito → Modo Personal activo | ↘ Comercial → Pago (\$449 MXN/mes) → Onboarding → Business Studio

---

## ## Modelo de Cuenta Dual

Un usuario puede tener **\*\*ambos modos\*\*** con el mismo correo:

Modo	Acceso	Costo
Personal	Siempre disponible	Gratis
Comercial	Requiere suscripción	\$449 MXN/mes

---

## ## Flujo Comercial Completo

Usuario selecciona "Comercial" en registro ↓ Se muestra modal de planes/pago ↓ Usuario paga suscripción (Stripe) ↓ Se crea cuenta con tieneModoComercial: true ↓ Se crea negocio en estado borrador ↓ Redirige a /business/onboarding ↓ Usuario completa 8 pasos del wizard ↓ Al finalizar: onboardingCompletado: true, esBorrador: false ↓ Acceso completo a Business Studio

---

## ## CTA "¿Tienes un negocio?"

Para usuarios que ya tienen cuenta Personal y quieren agregar Comercial:

Usuario hace clic en CTA ↓ Se muestra modal de planes ↓ Usuario paga suscripción ↓ Se actualiza: tieneModoComercial: true ↓ Se crea negocio en estado borrador ↓ Redirige a /business/onboarding ↓ Completa wizard → Acceso a Business Studio

---

## ## Alternancia de Modos

Usuarios con ambos modos pueden alternar:

[ Personal] ↔ [ Comercial]

- Toggle disponible en Navbar/Sidebar
- Cambia contenido y opciones del menú
- ChatYA unificado (mismo historial)

---

## ## Cancelación de Suscripción

Usuario cancela suscripción comercial ↓ Stripe webhook notifica ↓ Backend actualiza: |—— tieneModoComercial: false |—— modoActivo: 'personal' (forzado) |—— negocio: se oculta del directorio (no se borra) ↓ Usuario solo puede usar modo Personal ↓ Si reactiva suscripción → negocio vuelve a aparecer

---

## ## Decisión: Negocios Solo Físicos (06 Enero 2026)

✗ ELIMINADO: Tipo de negocio "Online"  TODOS los negocios requieren ubicación física

Campos en sucursales: |—— tiene\_servicio\_domicilio: boolean (tú vas al cliente: plomero, electricista) |—— tiene\_envio\_domicilio: boolean (envías productos: restaurante, farmacia)

---

## ## Nomenclatura Estandarizada (21 Dic 2024)

PostgreSQL: snake\_case (negocio\_id, created\_at) ↓ Drizzle (automático) TypeScript: camelCase (negocioid, createdAt) ↓ API Response: { success, data, message }

---

## ## Resumen del Proyecto

Área	Estado
Fases 1-4	<input checked="" type="checkbox"/> 100% Completadas
Fase 5	 ~40% (Bloque 1-2 completos)
Fase 6	 0% Pendiente
Business Studio	 15% (2/14 módulos)

---

\*Roadmap Maestro: 16 Enero 2026\*

\*Proyecto: AnunciaYA v3.0\*

\*Versión: 5.1\*

\*Desarrollador: Juan Manuel Valenzuela\*

\*Tiempo Fases 1-4: 4 semanas\*

\*Tiempo Total Estimado Restante: ~8-9 semanas\*

