



Tuki Delivery

Arquitectura, seguridad e integraciones de la plataforma de pedidos en línea

★ CLAUDE ASSISTED ANALYSIS | ● TUKI REWARDS INSIDE

Una instancia aislada por comercio, con la lealtad dentro del flujo de venta.

Tuki Delivery es la plataforma de pedidos en línea de Tuki: tienda web propia del comercio (domicilio y recoger), cobro con tarjeta, efectivo o puntos, tablero de operación en vivo y panel de administración — con el programa de lealtad Tuki operando de forma nativa en cada pedido y conectores a los puntos de venta del mercado (Soft Restaurant, InfoCaja, Odoo).

Dirigido a

Ingenieros y responsables de sistemas de comercios que evalúan o integran Tuki Delivery: arquitectura, modelo de datos, seguridad, integraciones y requisitos. No es un manual de operación — para eso existe el Manual de Uso de Tuki Delivery.

Objetivo del documento

Responder en pocas páginas las preguntas que un equipo técnico hace antes de adoptar la plataforma: dónde viven los datos, cómo se protegen, cómo se cobra, cómo se integra con el punto de venta y qué necesita el comercio para arrancar.

Lo que vas a encontrar.

01

Arquitectura y stack

Multi-instancia con aislamiento total por comercio, stack moderno y despliegue continuo.

02

Seguridad y pagos

Sesiones firmadas, credenciales cifradas, PCI del lado del procesador y montos que solo decide el servidor.

03

Integraciones

Lealtad Tuki nativa, adaptadores de punto de venta y aprovisionamiento de comercios.

04

Operación y requisitos

Los tres paneles, la observabilidad del negocio y lo que un comercio necesita para integrarse.

Aviso de alcance. Esta ficha describe la arquitectura y capacidades de Tuki Delivery vigentes a la fecha de publicación. Los componentes de infraestructura señalados son los actuales; la arquitectura es portable y puede desplegarse sobre infraestructura propia sin cambios de código. Algunos conectores se encuentran en fases distintas de liberación — se indica en cada caso.

Documento elaborado por Tuki — MARC (Mobile Ads and Rewards Corporation), septiembre de 2026. Alineado al Tuki Brand Book v1.1.

01

PARTE UNO

Arquitectura y stack.

Una instancia por comercio, un solo código base, y datos que jamás se mezclan.

Arquitectura multi-instancia.

Aislamiento por diseño: cada comercio corre su propia copia de la plataforma.

Tuki Delivery no es un SaaS de base compartida: cada comercio recibe una **instancia propia** — su aplicación, su base de datos PostgreSQL dedicada y su dominio — desplegada desde un **código base único y versionado**. El aislamiento de datos entre comercios es físico, no lógico: no existen consultas cross-tenant posibles porque no hay base compartida.

- **Dominio por comercio** — subdominio de plataforma (**{comercio}.tuki-delivery.com**) o dominio propio del comercio vía CNAME, con TLS automático.
- **Configuración por comercio** — identidad de marca, programa de lealtad, reglas comerciales, sucursales y credenciales viven en la base de la instancia y se administran desde su panel; el código no contiene nada de ningún comercio (white-label real).
- **Sucursales** — una instancia modela N sucursales, cada una con su punto de venta, su geolocalización, su tarifario de envío por distancia (ruteo por red vial) y sus horarios.
- **Escalamiento horizontal** — agregar un comercio es aprovisionar una instancia nueva (minutos), no migrar un esquema compartido.

Panel de plataforma

MARC opera un panel maestro independiente para el alta y supervisión de instancias, con emisión, rotación y revocación de credenciales de integración por comercio, y un servicio web de aprovisionamiento para el alta automática desde el ecosistema Tuki.

Stack tecnológico.

Componentes estándar de la industria, sin piezas exóticas que mantener.

- **Aplicación** — Next.js sobre Node.js 20 (App Router): páginas servidas con SSR ligero y lógica de negocio en API routes serverless. El cliente web es JavaScript nativo sin frameworks pesados — carga rápida en móviles de gama media, que es donde ordenan los clientes reales.
- **Persistencia** — PostgreSQL gestionado, accedido vía Prisma ORM (esquema versionado junto al código; las migraciones corren en el pipeline de despliegue).
- **Hosting** — infraestructura serverless gestionada (actualmente Vercel) con CDN global, TLS automático y autoescalado por demanda; base de datos gestionada (actualmente Neon) con respaldos automáticos.
- **CI/CD** — despliegue continuo desde Git: cada cambio integrado a la rama principal construye y publica de forma atómica en todas las instancias, con rollback inmediato a cualquier versión anterior.
- **Portabilidad** — la plataforma no depende de servicios propietarios del hosting: puede desplegarse en infraestructura propia (Node + PostgreSQL) sin cambios de código.

Modelo de datos y portabilidad.

Todo lo del comercio vive en su base — y su base se lleva completa a donde sea.

El modelo gira alrededor de pocas entidades: **Comercio** (identidad, marca, reglas), **Sucursal** (POS, geografía, tarifas), **Producto** (menú con modificadores y precios por canal), **Pedido** (ciclo de vida completo con tiempos por etapa, descuentos y estado de cobro), **Promoción** y **Descuento General** (mecánicas comerciales), y los usuarios de los paneles.

- **Imágenes dentro de la base** — logos, portadas y fotos de producto se almacenan en la base de datos, no en un bucket externo: código + variables de entorno + dump = instancia completa. Migrar o respaldar un comercio es un solo movimiento.
- **Pedidos auditables** — cada pedido conserva su carrito, descuentos aplicados (promociones, cupones, descuento general, puntos), método y estado de cobro, folio del POS y marcas de tiempo de cada etapa operativa.
- **Zona horaria de negocio** — cortes, reportes y días operativos se calculan en la zona del comercio (sin sorpresas de UTC en el cierre de caja).

02

PARTE DOS

Seguridad y pagos.

La plataforma nunca ve una tarjeta, los montos solo los decide el servidor, y a los webhooks no se les cree nada.

Autenticación y roles.

Tres poblaciones, tres mecanismos, cero contraseñas compartidas con la plataforma de lealtad.

- **Cliente final** — inicia sesión vía **SSO del programa de lealtad Tuki**: el token de retorno se valida servidor-a-servidor contra la plataforma Tuki y se emite una cookie de sesión **firmada, HttpOnly y SameSite**. Tuki Delivery nunca conoce ni almacena la contraseña del afiliado.
- **Equipo del comercio** — roles separados: administrador (panel completo) y cajeros individuales para el tablero de pedidos, con altas, bajas y bloqueo inmediato por usuario.
- **Plataforma** — el panel maestro de MARC y el servicio de aprovisionamiento usan credenciales propias, independientes de las de cualquier comercio, con rotación y revocación en caliente.

Credenciales de integración en reposo

Las credenciales de los puntos de venta capturadas en el panel se almacenan **cifradas con AES-256-GCM**; los paneles jamás las devuelven en claro (escritura ciega: capturar, reemplazar o borrar). Los secretos de infraestructura viven exclusivamente en variables de entorno, nunca en el código ni en la base.

Cobros: PCI del lado del procesador.

Los datos de tarjeta nunca tocan la plataforma.

- **Stripe** — checkout alojado por Stripe con **cuenta conectada del comercio**: el cargo se procesa y deposita directo en la cuenta del comercio (el cumplimiento PCI DSS recae en Stripe). Reembolsos desde el panel del procesador.
- **Soft Restaurant Payments** — alternativa por **links de pago** del ecosistema National Soft, con onboarding del comercio vía API y modo sandbox para certificación; mismo principio: la tarjeta vive en el procesador.
- **Efectivo** — contra entrega, con el monto y el cambio indicados al repartidor; el corte del día concilia efectivo, pagos en línea y pagos a repartidores.
- **Puntos Tuki** — el saldo se verifica contra la plataforma de lealtad en el momento del cobro, considerando puntos ya comprometidos en pedidos vivos.

Integridad de montos y webhooks.

El navegador propone; el servidor dispone.

- **Recalculo total en servidor** — precios, modificadores, promociones, cupones, descuento general, puntos y costo de envío se recalculan y validan en el servidor en cada cobro. Un cliente manipulado no puede alterar un monto: la interfaz solo muestra lo que el servidor ya decidió.
- **Canjes verificados** — recompensas y cupones se revalidan contra la lista real del afiliado en la plataforma Tuki antes de cobrar; un canje que no existe se rechaza.
- **Webhooks con desconfianza por diseño** — un aviso de pago entrante jamás marca un pedido como pagado por sí mismo: el estado se **verifica contra el API autenticado del procesador** antes de tocar la orden.
- **Conciliación por sondeo** — si un webhook se pierde, un proceso de reconciliación consulta los cobros pendientes y auto-sana el estado; los pedidos no pagados expiran solos y liberan los puntos reservados.

03

PARTE TRES

Integraciones.

Lealtad nativa, adaptadores de punto de venta por sucursal, y una regla de oro: la venta nunca se bloquea.

Lealtad Tuki nativa.

No es un plugin de puntos: es el programa del comercio operando dentro del flujo de venta.

- **SSO de afiliados** — el cliente entra con su cuenta Tuki del comercio; la tienda conoce su nivel, saldo, recompensas y cupones vigentes.
- **Acumulación por SKU** — cada venta reporta a Tuki el detalle por producto, alimentando las reglas del programa (incluyendo exclusiones, p. ej. productos de tasa cero).
- **Canjes y pago con puntos** — recompensas, cupones porcentuales, 2×1/3×2 y pago parcial o total con puntos, con los puntos comprometidos protegidos entre pedidos concurrentes.
- **Tolerancia a fallas** — si la plataforma de lealtad no responde, la venta continúa y la acumulación queda en cola de reintento. El cliente no pierde ni el pedido ni sus puntos.

Puntos de venta: adaptadores por sucursal.

Cada sucursal declara su POS; la plataforma habla el idioma de cada uno.

- **Soft Restaurant (National Soft)** — consulta del catálogo del restaurante vía Catalog API (importación del menú a la tienda con un clic) y cobro con Soft Restaurant Payments; envío de ventas al POS en liberación conjunta con National Soft.
- **InfoCaja** — envío de pedidos por servicio web SOAP con detalle completo: paquetes con comandas, modificadores, renglones de descuento con criterio fiscal (precio de lista + beneficio como renglón negativo nombrado) y referencia de folio; en producción con el piloto.
- **Odoo** — integración por XML-RPC: carga del menú (productos, categorías, atributos, imágenes) y creación de órdenes de venta confirmadas e idempotentes; probado de punta a punta.
- **Sin POS** — el tablero de pedidos opera de forma autónoma; el POS es un complemento, nunca un requisito.

La regla de oro

El puente al POS **jamás bloquea una venta**: si el punto de venta no responde, el pedido se cobra y se opera desde el tablero, y el reenvío al POS es idempotente (un pedido ya recibido no se duplica).

Servicios de plataforma.

Lo que rodea a la tienda: correo, analítica y aprovisionamiento.

- **Correo transaccional** — por el SMTP del propio comercio (confirmaciones, cancelaciones con estado de reembolso, reportes); un correo fallido jamás afecta un pedido.
- **Analítica** — Google Analytics propio por comercio (configurable desde el panel), además de los reportes internos de la plataforma.
- **Aprovisionamiento** — servicio web autenticado para el alta automática de comercios desde el ecosistema Tuki, con credenciales por comercio emitibles, rotables y revocables desde el panel de plataforma.

04

PARTE CUATRO

Operación y requisitos.

Tres paneles, observabilidad del negocio, y una lista corta de requisitos para arrancar.

Los tres paneles.

Tienda para el cliente, tablero para la operación, administración para el dueño.

- **La tienda** — web móvil-primero con menú, modificadores, canjes de lealtad, seguimiento del pedido en vivo (con enlace tokenizado) y pedidos programados con liberación automática a cocina.
- **Tablero de pedidos (/ops)** — columnas por etapa con actualización continua, alarma sonora persistente, mensaje de WhatsApp listo para el repartidor, interruptor de contingencia de la tienda y **Corte del día** (efectivo, pagado en línea, descuentos y pagos a repartidores para el cierre de caja).
- **Administración (/admin)** — catálogo y modificadores, promociones modelo POS, reglas del descuento general, marca (con paleta propuesta desde el logo), sucursales y credenciales, cajeros, y reportes siempre comparativos (periodo anterior y año previo) con resumen diario por correo.

Requisitos para integrar un comercio.

Nada que instalar, nada que hospedar.

1. **Programa Tuki activo** — Tuki Delivery se monta sobre el programa de lealtad existente del comercio.
2. **Dominio** — un subdominio del comercio (CNAME) o el subdominio de plataforma asignado.
3. **Cuenta de cobro** — Stripe (alta con datos fiscales y bancarios del comercio) o Soft Restaurant Payments.
4. **Menú** — catálogo con precios por canal y fotos; o importación directa desde el POS cuando el conector lo soporta.
5. **Opcionales** — credenciales del POS por sucursal, SMTP del comercio para correos transaccionales, propiedad de Google Analytics.

Del lado del cliente final

Cualquier navegador moderno en móvil o escritorio. No hay app que instalar ni cuenta obligatoria: la sesión de lealtad es opcional y aporta beneficios, nunca fricción.

Ciclo de vida y soporte.

Despliegue continuo con instancias siempre alineadas.

- **Actualizaciones** — todas las instancias corren la misma versión; las mejoras llegan a todos los comercios en el mismo despliegue, sin ventanas de mantenimiento.
- **Observabilidad** — logs de ejecución por instancia en la plataforma de hosting y auditoría funcional en la propia base (pedidos, cobros, envíos al POS con su estado).
- **Soporte** — soporte@tukicard.com, con niveles de atención por severidad (crítico 30 min, medio 2 h, bajo 24 h en horario de operación).