

# FastOps — Documento Maestro de Frontend V1.1

**Referencia Backend:** FastOps\_Documento\_Maestro\_V4.0\_Definitivo.md **Tecnología:** React 18 + TypeScript + TailwindCSS + Redux Toolkit **Enfoque:** PWA (Progressive Web App) Mobile-First para Clientes y Desktop-Optimized para Caja.

## 1. ARQUITECTURA DEL FRONTEND

### 1.1 Estructura Multi-Portal

El frontend manejará dos experiencias distintas basadas en el subdominio o la ruta:

1. **Portal Cliente (PWA Pública):** [https://\[slug\].fastops.com](https://[slug].fastops.com)
  - **Usuarios:** Clientes finales.
  - **UX:** Estilo UberEats/Rappi. Simplificada, visual, enfocada en conversión.
  - **Auth:** Opcional (Guest checkout) o Login simple.
2. **Portal Operativo (Staff PWA):** <https://app.fastops.com>
  - **Usuarios:** Admin, Cajero, Cocina, Domiciliario.
  - **UX:** Alta densidad de información, controles rápidos, modo oscuro opcional.
  - **Auth:** JWT Estricto + Selección de Company/Branch.

### 1.2 State Management (Redux Toolkit)

- **authSlice** : Token, User Roles, Company ID, Branch ID.
- **cartSlice** : Carrito del cliente (Persistencia local).
- **orderSlice** : Gestión de pedidos en tiempo real (Caja/Cocina).
- **uiSlice** : Modales, Sidebars, Toasts de notificación.
- **RTK Query:** Para caching de API (Productos, Menú, Historial).

### 1.3 Sistema de Impresión (NUEVO)

- **Componente ThermalTicket** : Un componente React invisible en la UI normal, que solo se renderiza al invocar `window.print()`.
- **CSS Print Media:** Estilos específicos para ocultar la interfaz (botones, menús) y solo mostrar el ticket formateado en blanco y negro, sin márgenes, optimizado para rollos de 58mm y 80mm.

## 2. REQUISITOS FUNCIONALES Y WIREFRAMES

## MÓDULO 1: PWA CLIENTE (PEDIDOS)

**Requisito:** Permitir armar pedidos complejos (con modificadores) y enviarlos con geolocalización.

### Pantalla 1.1: Catálogo (Home)

- **Header:** Logo del Negocio, Botón Login, Icono Carrito (con badge de cantidad).
- **Buscador:** Input de texto.
- **Categorías (Sticky):** Scroll horizontal (Hamburguesas, Bebidas, Adiciones).
- **Grid Productos:** Tarjeta con Foto, Nombre, Precio Base, Botón "Agregar".

### Pantalla 1.2: Builder de Producto (Modal/Page)

- **Hero:** Foto grande del producto.
- **Selector Cantidad:** - 1 +
- **Sección Modificadores (Crucial V4.0):**
  - **Adiciones (Checkbox):** "Extra Queso (+\$3.000)", "Tocineta (+\$4.000)".
  - **Exclusiones (Checkbox):** "Sin Cebolla", "Sin Salsas".
- **Notas:** Textarea "Instrucciones especiales".
- **Footer:** Botón "Agregar al Pedido - \$Total Calculado".

### Pantalla 1.3: Checkout

- **Resumen:** Lista de ítems.
- **Tipo Entrega (Tabs):** [Mesa] | [Llevar] | [Domicilio].
  - *Si es Domicilio:* Formulario Dirección + Botón "Usar mi ubicación GPS".
- **Pago:** Selección método (Efectivo/Transferencia).
- **Action:** Botón "Enviar Pedido por WhatsApp" (Fallback) o "Enviar al Sistema".

## MÓDULO 2: POS & CAJA (OPERATIVO)

**Requisito:** Velocidad. Recepción de pedidos online y toma de pedidos locales. Asignación de drivers. **Impresión térmica.**

### Pantalla 2.1: Terminal de Venta (POS)

**Layout:** 2 Columnas (70% Catálogo / 30% Ticket).

- **Izquierda (Catálogo):** Grid de productos con buscador rápido.
- **Derecha (Ticket Virtual):**

- Lista de ítems agregados.
- Selector de Cliente (Búsqueda por tel).
- Botones Acción:
  - "Cobrar e Imprimir" (Cierra venta y lanza print dialog).
  - "Solo Guardar".
  - "Imprimir Pre-cuenta" (Para llevar a la mesa).

### Pantalla 2.2: Dispatcher (Control de Despachos)

**Layout:** Kanban Board (Columnas).

- **Columna 1: En Cocina:** Tarjetas de pedidos en preparación (Timer en verde/rojo).
- **Columna 2: Listos:** Pedidos terminados esperando driver.
  - **Acción:** Botón "Re-imprimir Ticket" (por si se perdió o para pegar en la bolsa).
- **Columna 3: En Ruta:** Pedidos asignados.
- **Sidebar Drivers:** Lista de Domiciliarios (Online/Offline).

### Pantalla 2.3: Arqueo de Caja (Ciego)

- **Input:** Campos numéricos para ingresar cantidad de billetes/monedas o total contado.
- **NO MOSTRAR:** El total esperado por el sistema (Seguridad V4.0).
- **Action:** "Cerrar Turno e Imprimir Reporte" (Ticket térmico con resumen del cierre).

### Pantalla 2.4: Diseño del Ticket Térmico (Invisible/Print-Only)

**Formato:** Estrecho (58mm/80mm), Fuente Monospace o Sans-Serif condensada.

- **Cabecera:** Logo (B/N), Nombre Negocio, NIT, Dirección, #Orden, Fecha/Hora.
- **Cuerpo (Tabla Simple):**
  - Cantidad | Producto | Total
  - *Detalle:* Modificadores debajo del producto (ej: "+ Extra Queso").
- **Totalles:** Subtotal, Descuentos, Total a Pagar, Método de Pago.
- **Pie:** Info Wifi, "Gracias por su compra", Espacio para propina (si aplica).
- **Corte:** Espacio en blanco al final.

## MÓDULO 3: COCINA (KDS)

**Requisito:** Visualización clara, alertas sonoras, gestión de tiempos.

## Pantalla 3.1: Comanderas Digitales

**Layout:** Grid Responsivo (Masonry).

- **Tarjeta de Pedido:**
  - Header: #Orden, Hora, Canal (Mesa/App), Timer (Minutos transcurridos).
  - Body: Lista de ítems.
    - *Resaltado:* Modificadores de Exclusión en ROJO ("SIN CEBOLLA").
    - *Resaltado:* Adiciones en AZUL ("EXTRA TOCINETA").
  - Footer:
    - Botón grande "LISTO".
    - Botón secundario "IMPRIMIR COMANDA" (Ticket sin precios, solo ítems grandes para pegar en el plato).
- **Alertas:** Flash visual y sonido cuando entra nueva orden (Socket).

## MÓDULO 4: INVENTARIO (AUDITORÍA)

**Requisito:** Wizard paso a paso para conteo ciego.

### Pantalla 4.1: Setup Auditoría

- **Selector de Modo (Tarjetas):**
  - [ Flash Top 10]
  - [ Personalizado]
  - [ Full Cierre]
- **Action:** "Iniciar Conteo".

### Pantalla 4.2: Wizard de Conteo (Mobile Focus)

- **Header:** Ítem 1 de 10.
- **Card Central:**
  - Nombre: "Queso Mozzarella".
  - Unidad: "Bloques".
- **Input Gigante:** [ 0.00 ] (Teclado numérico en pantalla).
- **Navegación:** Botón "Siguiente".

## 3. ESTRATEGIA DE DESARROLLO INCREMENTAL

Para evitar el caos, el desarrollo se divide en 3 capas estrictas. No pasar a la siguiente sin aprobar la anterior.

### ● FASE A: ESTRUCTURA Y RUTAS (Skeleton)

Objetivo: Tener la navegación funcionando sin diseño visual ni lógica compleja.

1. **Setup:** Vite + React + TS . Configuración de ESLint/Prettier.
2. **Routing:** Implementar react-router-dom .
  - Detectar subdominio. Si es `app` , ir a rutas Admin. Si es `slug` , ir a rutas Cliente.
3. **Componentes Cascarón:** Crear las páginas vacías con solo un `<h1>Título</h1>` .
  - `src/pages/client/Catalog.tsx`
  - `src/pages/admin/POS.tsx`
  - `src/pages/admin/Kitchen.tsx`
4. **Navegación:** Crear los Layouts (Sidebar para admin, Navbar para cliente).

### ● FASE B: ESTILOS Y COMPONENTES UI (Static)

Objetivo: Que la app se vea profesional y responsive (Tailwind).

1. **Design System:** Configurar `tailwind.config.js` con los colores de marca.
2. **Componentes UI (Átomos):** Crear botones, inputs, cards, modales y badges.
3. **Maquetación de Vistas:**
  - Construir el **Grid del Catálogo** (Responsive).
  - Construir el **Tablero Kanban** del Dispatcher.
  - Diseñar el **Ticket de Comanda** (KDS) con colores de alerta.
4. **Diseño de Tickets Térmicos (NUEVO):**
  - Crear `components/print/Receipt.tsx` .
  - Configurar `@media print` en CSS para ocultar todo `body` excepto el contenedor del ticket.
  - Probar estilos con anchos de 58mm y 80mm.
5. **Responsive Check:** Verificar que el KDS y el POS funcionen en Tablet y el Cliente en Móvil.

### ● FASE C: INTEGRACIÓN Y LÓGICA (Dynamic)

Objetivo: Dar vida a la app conectando con el Backend V4.0.

1. **API Client:** Configurar `axios` con interceptors para JWT.
2. **Auth Integration:** Login form, guardar token, redirección por rol.

3. **Data Fetching:** Conectar Catálogo con GET /products .
4. **Lógica de Carrito:** Implementar cartSlice (Aregar, sumar, quitar, calcular total con modifiers).
5. **WebSockets:**
  - Conectar socket.io-client .
  - Escuchar evento new\_order -> Actualizar KDS.
6. **Impresión (NUEVO):**
  - Conectar el botón "Imprimir" del POS para inyectar datos en Receipt.tsx y llamar window.print() .

## 4. DIRECTORIOS SUGERIDOS

```

src/
├── assets/           # Imágenes estáticas
├── components/
│   ├── ui/            # Botones, Inputs, Modales (Genéricos)
│   ├── client/         # ProductCard, CartDrawer (Específicos Cliente)
│   ├── admin/          # KanbanBoard, TicketCard (Específicos Admin)
│   └── print/          # ThermalReceipt (NUEVO: Componente de impresión)
├── hooks/            # useAuth, useSocket, useCart
├── layouts/          # ClientLayout, AdminLayout
├── pages/
│   ├── auth/           # Login
│   ├── client/          # Menu, Checkout, Tracking
│   └── admin/           # POS, Kitchen, Inventory, Settings
├── store/             # Redux Slices (auth, cart, ui)
├── services/          # API calls (axios instances)
├── types/              # Interfaces TypeScript (Product, Order, User)
└── utils/              # FormatCurrency, DateFormatters

```