



# Plan Detallado Fase 2: POS + Inventario Core

**Duración:** 4-6 semanas | **Prioridad:** Alta | **Estado:** Planificado



## Objetivos de la Fase 2

### Objetivo Principal

Desarrollar el corazón funcional de CRTLPyme: un sistema POS completo y gestión de inventario robusta que permita a las PYMEs chilenas operar eficientemente.

### Objetivos Específicos

- Sistema POS Completo:** Interface de venta intuitiva y rápida
- Gestión de Inventario:** Control de stock, productos y categorías
- Integración Productos Chilenos:** Base de datos con códigos de barras locales
- Sistema de Transacciones:** Procesamiento de ventas y métodos de pago
- Reportes Básicos:** Analytics esenciales para toma de decisiones



## Cronograma Detallado

### Semana 1-2: Sistema POS Core

#### Semana 1: Interface POS Base

- ☐ Días 1-2: Diseño y layout POS
  - ☐ Interface de venta principal
  - ☐ Carrito de compras dinámico
  - ☐ Búsqueda de productos
  - ☐ Calculadora de totales
- ☐ Días 3-4: Funcionalidad de productos
  - ☐ Catálogo de productos
  - ☐ Búsqueda por código de barras
  - ☐ Búsqueda por nombre/categoría
  - ☐ Información detallada de productos
- ☐ Días 5-7: Carrito y cálculos
  - ☐ Agregar/quitar productos
  - ☐ Modificar cantidades
  - ☐ Aplicar descuentos
  - ☐ Cálculo de impuestos (IVA)
  - ☐ Total **final** con desglose

#### Semana 2: Procesamiento de Ventas

- ☐ Días 1-2: Métodos de pago
  - ☐ Efectivo con cálculo de vuelto
  - ☐ Tarjeta de débito/crédito
  - ☐ Transferencia bancaria
  - ☐ Pago mixto (múltiples métodos)
- ☐ Días 3-4: Finalización de ventas
  - ☐ Confirmación de venta
  - ☐ Generación de recibo
  - ☐ Actualización de inventario
  - ☐ Registro en historial
- ☐ Días 5-7: Testing y optimización
  - ☐ Testing de flujos completos
  - ☐ Optimización de performance
  - ☐ Manejo de errores
  - ☐ Validaciones de negocio

## Semana 3-4: Gestión de Inventario

### Semana 3: CRUD Productos Avanzado

- └─ Días 1-2: Gestión de productos
  - └─ Crear productos con detalles completos
  - └─ Editar información de productos
  - └─ Eliminar productos (soft delete)
  - └─ Importar productos masivamente
- └─ Días 3-4: Categorías y organización
  - └─ Sistema de categorías jerárquico
  - └─ Etiquetas y filtros
  - └─ Búsqueda avanzada
  - └─ Organización por proveedor
- └─ Días 5-7: Códigos de barras
  - └─ Generación de códigos de barras
  - └─ Lectura de códigos existentes
  - └─ Validación de códigos únicos
  - └─ Integración con productos chilenos

### Semana 4: Control de Stock

- └─ Días 1-2: Gestión de stock
  - └─ Control de cantidades
  - └─ Stock mínimo y máximo
  - └─ Alertas de stock bajo
  - └─ Historial de movimientos
- └─ Días 3-4: Proveedores y compras
  - └─ Gestión de proveedores
  - └─ Órdenes de compra básicas
  - └─ Recepción de mercadería
  - └─ Actualización de costos
- └─ Días 5-7: Integración productos chilenos
  - └─ Base de datos productos locales
  - └─ Importación automática
  - └─ Validación de códigos chilenos
  - └─ Actualización de precios

## Semana 5-6: Reportes y Optimización

### Semana 5: Sistema de Reportes

- Días 1-2: Reportes de ventas
  - Ventas por día/semana/mes
  - Productos más vendidos
  - Ventas por vendedor
  - Comparativas de períodos
- Días 3-4: Reportes de inventario
  - Estado actual de stock
  - Productos con stock bajo
  - Rotación de inventario
  - Valorización de inventario
- Días 5-7: Dashboard con métricas
  - KPIs principales en tiempo real
  - Gráficos interactivos
  - Alertas y notificaciones
  - Exportación de reportes

### Semana 6: Testing y Optimización Final

- Días 1-2: Testing integral
    - Testing de todos los flujos
    - Testing de performance
    - Testing de concurrencia
    - Testing de datos reales
  - Días 3-4: Optimización
    - Optimización de consultas DB
    - Caching de datos frecuentes
    - Optimización de UI/UX
    - Performance tuning
  - Días 5-7: Documentación y demo
    - Documentación técnica
    - Guías de usuario
    - Preparación demo Fase 2
    - Planificación Fase 3
-

## Arquitectura Técnica Detallada

### Base de Datos - Esquemas Principales

#### Productos

```
model Product {
  id          String    @id @default(cuid())
  name        String
  description  String?
  sku          String    @unique
  barcode      String?   @unique
  price        Decimal
  cost         Decimal?
  stock        Int       @default(0)
  minStock     Int       @default(0)
  maxStock     Int?
  categoryId   String?
  category     Category? @relation(fields: [categoryId], references: [id])
  supplierId   String?
  supplier     Supplier? @relation(fields: [supplierId], references: [id])
  isActive     Boolean   @default(true)
  createdAt    DateTime  @default(now())
  updatedAt    DateTime  @updatedAt

  // Relaciones
  saleItems    SaleItem[]
  stockMovements StockMovement[]
}
```

#### Ventas

```
model Sale {
  id          String    @id @default(cuid())
  saleNumber   String    @unique
  total        Decimal
  subtotal     Decimal
  tax          Decimal
  discount     Decimal   @default(0)
  paymentMethod String
  status       SaleStatus @default(COMPLETED)
  customerId   String?
  customer     Customer? @relation(fields: [customerId], references: [id])
  userId       String
  user         User       @relation(fields: [userId], references: [id])
  companyId    String
  company      Company    @relation(fields: [companyId], references: [id])
  createdAt    DateTime  @default(now())

  // Relaciones
  items        SaleItem[]
  payments     Payment[]
}
```

## Items de Venta

```
model SaleItem {
  id          String  @id @default(cuid())
  quantity    Int
  unitPrice   Decimal
  total       Decimal
  discount    Decimal @default(0)
  saleId      String
  sale        Sale    @relation(fields: [saleId], references: [id])
  productId   String
  product     Product @relation(fields: [productId], references: [id])
}
```

## APIs Principales

### POS APIs

- POST /api/pos/search-product - Búsqueda de productos
- POST /api/pos/add-to-cart - Agregar al carrito
- PUT /api/pos/update-cart-item - Actualizar item del carrito
- DELETE /api/pos/remove-from-cart - Remover del carrito
- POST /api/pos/process-sale - Procesar venta completa
- GET /api/pos/payment-methods - Métodos de pago disponibles

### Inventario APIs

- GET /api/inventory/products - Listar productos con paginación
- POST /api/inventory/products - Crear producto
- PUT /api/inventory/products/:id - Actualizar producto
- DELETE /api/inventory/products/:id - Eliminar producto
- POST /api/inventory/import - Importación masiva
- GET /api/inventory/stock-alerts - Alertas de stock bajo

### Reportes APIs

- GET /api/reports/sales-summary - Resumen de ventas
  - GET /api/reports/top-products - Productos más vendidos
  - GET /api/reports/inventory-status - Estado de inventario
  - GET /api/reports/sales-by-period - Ventas por período
-

## Componentes Frontend Principales

### 1. Sistema POS

#### 1.1 Componente POS Principal

```
// components/pos/POSInterface.tsx
interface POSInterfaceProps {
  user: User;
  company: Company;
}

const POSInterface = ({ user, company }: POSInterfaceProps) => {
  // Estado del carrito, productos, etc.
  // Lógica de búsqueda y selección
  // Procesamiento de pagos
  // Generación de recibos
}
```

#### 1.2 Carrito de Compras

```
// components/pos/ShoppingCart.tsx
interface CartItem {
  product: Product;
  quantity: number;
  unitPrice: number;
  discount: number;
  total: number;
}

const ShoppingCart = ({ items, onUpdateItem, onRemoveItem }: CartProps) => {
  // Renderizado de items
  // Cálculos de totales
  // Aplicación de descuentos
}
```

#### 1.3 Búsqueda de Productos

```
// components/pos/ProductSearch.tsx
const ProductSearch = ({ onProductSelect }: ProductSearchProps) => {
  // Búsqueda por nombre, SKU, código de barras
  // Filtros por categoría
  // Resultados en tiempo real
  // Selección rápida
}
```

#### 1.4 Procesamiento de Pagos

```
// components/pos/PaymentProcessor.tsx
const PaymentProcessor = ({ total, onPaymentComplete }: PaymentProps) => {
  // Selección de método de pago
  // Cálculo de vuelto para efectivo
  // Validación de montos
  // Confirmación de pago
}
```

## 2. Gestión de Inventario

### 2.1 Lista de Productos

```
// components/inventory/ProductList.tsx
const ProductList = ({ filters, onProductEdit }: ProductListProps) => {
  // Tabla con paginación
  // Filtros y búsqueda
  // Acciones rápidas (editar, eliminar)
  // Indicadores de stock
}
```

### 2.2 Formulario de Producto

```
// components/inventory/ProductForm.tsx
const ProductForm = ({ product, onSave }: ProductFormProps) => {
  // Formulario completo de producto
  // Validaciones
  // Subida de imágenes
  // Generación de códigos de barras
}
```

### 2.3 Gestión de Categorías

```
// components/inventory/CategoryManager.tsx
const CategoryManager = ({ categories, onCategoryChange }: CategoryProps) => {
  // Árbol de categorías
  // CRUD de categorías
  // Drag & drop para reorganizar
}
```

## 3. Sistema de Reportes

### 3.1 Dashboard de Métricas

```
// components/reports/MetricsDashboard.tsx
const MetricsDashboard = ({ dateRange }: DashboardProps) => {
  // KPIs principales
  // Gráficos interactivos
  // Comparativas de períodos
  // Alertas y notificaciones
}
```

### 3.2 Reportes de Ventas

```
// components/reports/SalesReports.tsx
const SalesReports = ({ filters }: SalesReportsProps) => {
  // Tabla de ventas
  // Filtros avanzados
  // Exportación a Excel/PDF
  // Gráficos de tendencias
}
```

---





## Funcionalidades Específicas

---

### 1. Sistema POS Completo

#### 1.1 Interface de Venta

- [ ] **Layout Intuitivo:** Diseño optimizado para uso rápido
- [ ] **Búsqueda Rápida:** Por nombre, SKU, código de barras
- [ ] **Carrito Dinámico:** Agregar/quitar productos fácilmente
- [ ] **Cálculos Automáticos:** Subtotal, impuestos, descuentos, total
- [ ] **Shortcuts de Teclado:** Navegación rápida con teclado

#### 1.2 Gestión de Productos en POS

- [ ] **Catálogo Visual:** Imágenes de productos
- [ ] **Información Detallada:** Precio, stock, descripción
- [ ] **Filtros Rápidos:** Por categoría, precio, disponibilidad
- [ ] **Productos Favoritos:** Acceso rápido a productos frecuentes
- [ ] **Sugerencias:** Productos relacionados o complementarios

#### 1.3 Procesamiento de Ventas

- [ ] **Múltiples Métodos de Pago:** Efectivo, tarjeta, transferencia
- [ ] **Pago Mixto:** Combinación de métodos de pago
- [ ] **Cálculo de Vuelto:** Automático para pagos en efectivo
- [ ] **Validaciones:** Verificación de stock, precios, descuentos
- [ ] **Confirmación:** Resumen antes de finalizar venta

#### 1.4 Generación de Recibos

- [ ] **Recibo Digital:** PDF generado automáticamente
- [ ] **Recibo Impreso:** Compatible con impresoras térmicas
- [ ] **Información Completa:** Productos, precios, impuestos, total
- [ ] **Branding:** Logo y datos de la empresa
- [ ] **Numeración:** Correlativo automático de recibos

### 2. Gestión de Inventario Avanzada

#### 2.1 CRUD de Productos

- [ ] **Crear Productos:** Formulario completo con validaciones
- [ ] **Editar Productos:** Actualización de información y precios
- [ ] **Eliminar Productos:** Soft delete para mantener historial
- [ ] **Duplicar Productos:** Crear productos similares rápidamente
- [ ] **Importación Masiva:** Excel/CSV para carga de productos

#### 2.2 Información de Productos

- [ ] **Datos Básicos:** Nombre, descripción, SKU, código de barras
- [ ] **Precios:** Precio de venta, costo, margen de ganancia
- [ ] **Stock:** Cantidad actual, mínimo, máximo
- [ ] **Categorización:** Categorías, subcategorías, etiquetas
- [ ] **Proveedor:** Información del proveedor principal

#### 2.3 Control de Stock

- [ ] **Seguimiento en Tiempo Real:** Actualización automática con ventas

- [ ] **Alertas de Stock Bajo:** Notificaciones cuando stock < mínimo
- [ ] **Historial de Movimientos:** Log de entradas y salidas
- [ ] **Ajustes de Inventario:** Correcciones manuales con justificación
- [ ] **Valorización:** Cálculo del valor total del inventario

## 2.4 Categorías y Organización

- [ ] **Sistema Jerárquico:** Categorías y subcategorías
- [ ] **Filtros Avanzados:** Múltiples criterios de búsqueda
- [ ] **Etiquetas:** Sistema de tags para organización flexible
- [ ] **Búsqueda Inteligente:** Búsqueda por múltiples campos
- [ ] **Ordenamiento:** Por nombre, precio, stock, ventas

## 3. Integración Productos Chilenos

### 3.1 Base de Datos Local

- [ ] **Productos Comunes:** Base de datos con productos chilenos típicos
- [ ] **Códigos de Barras:** Códigos EAN-13 válidos para Chile
- [ ] **Precios Referenciales:** Precios promedio del mercado chileno
- [ ] **Categorías Locales:** Categorización adaptada al mercado local
- [ ] **Proveedores:** Base de datos de proveedores chilenos

### 3.2 Importación Automática

- [ ] **Búsqueda por Código:** Buscar producto por código de barras
- [ ] **Importación Rápida:** Un clic para agregar producto conocido
- [ ] **Actualización de Precios:** Sincronización con precios de mercado
- [ ] **Validación de Códigos:** Verificación de códigos de barras válidos
- [ ] **Sugerencias:** Productos similares o relacionados

## 4. Sistema de Reportes Básicos

### 4.1 Reportes de Ventas

- [ ] **Ventas por Período:** Día, semana, mes, año
- [ ] **Ventas por Producto:** Productos más y menos vendidos
- [ ] **Ventas por Vendedor:** Performance individual del equipo
- [ ] **Ventas por Método de Pago:** Distribución de métodos de pago
- [ ] **Comparativas:** Comparación entre períodos

### 4.2 Reportes de Inventario

- [ ] **Estado de Stock:** Productos con stock bajo, alto, sin stock
- [ ] **Rotación de Inventario:** Productos de alta y baja rotación
- [ ] **Valorización:** Valor total del inventario por categoría
- [ ] **Movimientos:** Historial de entradas y salidas
- [ ] **Productos Inactivos:** Productos sin movimiento

### 4.3 Dashboard Ejecutivo

- [ ] **KPIs Principales:** Ventas del día, semana, mes
- [ ] **Gráficos Interactivos:** Tendencias de ventas y stock
- [ ] **Alertas:** Stock bajo, metas no cumplidas
- [ ] **Resumen Financiero:** Ingresos, costos, márgenes
- [ ] **Métricas de Equipo:** Performance de vendedores

---

## Plan de Testing Fase 2

---

### Testing Funcional

#### POS Testing

- [ ] **Flujo Completo de Venta:** Desde búsqueda hasta recibo
- [ ] **Métodos de Pago:** Cada método individualmente y combinados
- [ ] **Cálculos:** Verificar precisión en totales, impuestos, descuentos
- [ ] **Stock:** Verificar actualización correcta después de ventas
- [ ] **Errores:** Manejo de productos sin stock, precios inválidos

#### Inventario Testing

- [ ] **CRUD Productos:** Crear, leer, actualizar, eliminar productos
- [ ] **Importación:** Carga masiva de productos desde Excel/CSV
- [ ] **Búsquedas:** Verificar filtros y búsquedas avanzadas
- [ ] **Categorías:** Gestión de categorías y subcategorías
- [ ] **Stock:** Alertas de stock bajo, ajustes de inventario

#### Reportes Testing

- [ ] **Precisión de Datos:** Verificar que los reportes reflejen datos reales
- [ ] **Filtros:** Testing de filtros por fecha, producto, vendedor
- [ ] **Exportación:** Verificar exportación a Excel/PDF
- [ ] **Performance:** Tiempo de generación de reportes grandes
- [ ] **Visualización:** Gráficos y tablas se muestran correctamente

### Testing de Performance

#### Carga de Datos

- [ ] **1,000 Productos:** Performance con catálogo mediano
- [ ] **10,000 Productos:** Performance con catálogo grande
- [ ] **100 Ventas Simultáneas:** Testing de concurrencia
- [ ] **Reportes Grandes:** Reportes con miles de registros
- [ ] **Búsquedas:** Tiempo de respuesta en búsquedas complejas

#### Optimización

- [ ] **Consultas DB:** Optimización de queries complejas
- [ ] **Caching:** Implementar cache para datos frecuentes
- [ ] **Paginación:** Implementar paginación eficiente
- [ ] **Lazy Loading:** Carga diferida de componentes pesados
- [ ] **Compresión:** Compresión de respuestas API

### Testing de Integración

#### Base de Datos

- [ ] **Transacciones:** Verificar integridad en operaciones complejas
- [ ] **Relaciones:** Verificar integridad referencial
- [ ] **Migraciones:** Testing de migraciones de esquema
- [ ] **Backup/Restore:** Verificar procesos de respaldo
- [ ] **Concurrencia:** Testing de acceso simultáneo

## APIs

- [ ] **Endpoints:** Testing de todos los endpoints
  - [ ] **Validaciones:** Verificar validaciones server-side
  - [ ] **Errores:** Manejo correcto de errores y excepciones
  - [ ] **Autenticación:** Verificar permisos por rol
  - [ ] **Rate Limiting:** Testing de límites de requests
- 



## Métricas de Éxito Fase 2

---

### Métricas Técnicas

- [ ] **Performance POS:** Procesamiento de venta < 2 segundos
- [ ] **Búsqueda Productos:** Resultados en < 500ms
- [ ] **Carga de Inventario:** Lista de 1000 productos < 1 segundo
- [ ] **Generación Reportes:** Reportes básicos < 3 segundos
- [ ] **Uptime:** 99.9% disponibilidad durante testing

### Métricas de Negocio

- [ ] **Precisión de Stock:** 100% precisión en cálculos de stock
- [ ] **Integridad de Ventas:** 0 discrepancias en totales de ventas
- [ ] **Compleitud de Datos:** 100% de campos requeridos validados
- [ ] **Usabilidad POS:** Venta completa en < 1 minuto
- [ ] **Reportes Precisos:** 100% precisión en reportes vs datos reales

### Métricas de Usuario

- [ ] **Facilidad de Uso:** POS intuitivo para usuarios no técnicos
  - [ ] **Tiempo de Aprendizaje:** Usuario nuevo operativo en < 30 minutos
  - [ ] **Eficiencia:** Reducción 50% tiempo vs proceso manual
  - [ ] **Satisfacción:** Feedback positivo en usabilidad
  - [ ] **Adopción:** 100% de funciones core utilizadas
-

## Riesgos y Mitigaciones Fase 2

### Riesgos Técnicos

Riesgo	Probabilidad	Impacto	Mitigación
Performance con muchos productos	Alta	Alto	Paginación, indexación DB, caching
Concurrencia en ventas	Media	Alto	Transacciones DB, locking optimista
Precisión en cálculos	Baja	Crítico	Testing exhaustivo, validaciones múltiples
Integración códigos de barras	Media	Medio	Base de datos robusta, validaciones

### Riesgos de Negocio

Riesgo	Probabilidad	Impacto	Mitigación
Complejidad del POS	Media	Alto	Diseño simple, testing con usuarios reales
Datos de productos chilenos	Alta	Medio	Investigación de mercado, validación con PYMEs
Reportes no útiles	Media	Medio	Feedback temprano, iteración rápida
Curva de aprendizaje	Media	Medio	Documentación clara, tutoriales

### Riesgos de Tiempo

Riesgo	Probabilidad	Impacto	Mitigación
Subestimación de complejidad POS	Alta	Alto	Buffer time, desarrollo iterativo
Testing toma más tiempo	Alta	Medio	Testing paralelo, automatización
Integración productos chilenos	Media	Medio	Comenzar temprano, datos mock iniciales

---

## Checklist Final Fase 2

---

### Pre-Demo Técnico

- ☐ Sistema POS completamente funcional
- ☐ Gestión de inventario operativa
- ☐ Reportes básicos generándose correctamente
- ☐ Base de datos con productos chilenos cargada
- ☐ Testing completo realizado sin errores críticos
- ☐ Performance optimizada según métricas
- ☐ Documentación técnica actualizada

### Pre-Demo Negocio

- ☐ Datos de demo realistas cargados
- ☐ Flujos de negocio validados
- ☐ Casos de uso típicos de PYMEs probados
- ☐ Reportes con datos significativos
- ☐ Feedback de usuarios beta incorporado
- ☐ Comparación con competencia documentada

### Demo Preparation

- ☐ Presentación de 20-30 minutos preparada
- ☐ Demo script con casos de uso reales
- ☐ Datos de prueba representativos
- ☐ Backup plan para problemas técnicos
- ☐ Métricas de performance documentadas
- ☐ Feedback form para evaluadores

### Post-Demo

- ☐ Feedback recolectado y categorizado
- ☐ Issues críticos identificados y priorizados
- ☐ Plan de correcciones pre-Fase 3 definido
- ☐ Lecciones aprendidas documentadas
- ☐ Preparación Fase 3 iniciada
- ☐ Celebración del hito alcanzado 🎉

---

**Fecha de inicio objetivo:** Octubre 4, 2024

**Fecha objetivo de finalización:** Noviembre 15, 2024

**Responsable:** [Tu nombre]

**Stakeholders:** Profesor guía, evaluadores de tesis

**Próxima revisión:** Octubre 11, 2024

---

Este plan detallado será la guía principal para el desarrollo de la funcionalidad core de CRTLPyme. Se actualiza semanalmente basado en progreso real y feedback recibido.