

# DOCUMENTACIÓN COMPLETA - SISTEMA ERP CLOUD MULTI-NEGOCIO

**Versión:** 1.0

**Fecha:** Octubre 2025

**Sistema:** ERP Cloud SaaS Multi-tenant con TPV Local

---



## ÍNDICE COMPLETO

1. Visión General del Proyecto
  2. Arquitectura del Sistema
  3. Stack Tecnológico
  4. Sistema de Licencias y Planes
  5. Módulos del Sistema
    - 5.1 Autenticación y Usuarios
    - 5.2 Clientes y Proveedores
    - 5.3 Productos y Escandallos
    - 5.4 Ciclo Comercial (Ventas/Compras)
    - 5.5 TPV Local Offline
    - 5.6 Inventario Multi-almacén
    - 5.7 Módulo Restauración/Hostelería
    - 5.8 Módulo Taller Mecánico
    - 5.9 Módulo Informática
    - 5.10 Módulo Proyectos
    - 5.11 RRHH y Fichajes
    - 5.12 CRM y Marketing
    - 5.13 Reportes y BI
    - 5.14 Gestión Documental
    - 5.15 Comunicación
    - 5.16 Portal del Cliente
  6. Modelos de Base de Datos
  7. APIs y Endpoints
  8. Integraciones
  9. IA y Automatización
  10. Seguridad y Cumplimiento Legal
  11. Notificaciones
  12. Roadmap de Desarrollo
-

# 1. VISIÓN GENERAL DEL PROYECTO {#1-visión-general}









## 1.1 Descripción

Sistema ERP Cloud (SaaS Multi-tenant) modular, escalable y adaptable a múltiples verticales de negocio.  
Arquitectura híbrida: backend 100% cloud + TPV local offline-first.

## 1.2 Tipos de Negocio Soportados

Tipo	Módulos Principales
Retail/Comercio	Productos, Ventas, TPV, Inventario, eCommerce
Restauración	Mesas, KDS, Comandas, Escandallos, Carta Digital
Taller Mecánico	Vehículos, OT, Partes de Trabajo, Recambios
Informática	Tickets, Equipos, Contratos, Mantenimientos, Licencias
Servicios	Proyectos, Tareas, Certificaciones, Facturación por hitos
Producción	BOM, Producción, Control de calidad

## 1.3 Características Clave

-  **100% Cloud:** Backend centralizado en MongoDB Atlas
-  **TPV Offline:** Aplicación local con sincronización inteligente
-  **Modular:** Activar/desactivar módulos según negocio
-  **Multi-idioma:** ES, CA, EN (expandible)
-  **IA Integrada:** Asistente de voz, predicciones, OCR
-  **Responsive PWA:** Funciona en cualquier dispositivo
-  **Legal:** RGPD, Ley Antifraude, TicketBAI, Verifactu
-  **Sincronización:** Bidireccional cloud ↔ TPV local

---

# 2. ARQUITECTURA DEL SISTEMA {#2-arquitectura}

## 2.1 Diagrama de Arquitectura

## CLOUD BACKEND (SaaS)

Next.js 14+ App Router + React 18

- Server Components (SSR)

- API Routes (REST)

- Server Actions (mutations)

- WebSockets (real-time)

MongoDB Atlas (Multi-tenant)

- Segregación por empresaId

- Sharding para escalabilidad

- Backups automáticos cada 6h

Servicios Auxiliares

- Jobs (facturación recurrente, alertas, limpieza)

- IA (OpenAI GPT-4 / Claude)

- Email (Resend), SMS (Twilio), WhatsApp (Twilio API)

- Storage (S3 / Cloudinary)

- OCR (Tesseract.js / Google Vision)

↕ HTTPS + WSS

## WEB APP (Responsive PWA)

- Administración completa

- Acceso desde cualquier navegador/dispositivo

- Offline básico (Service Workers)

↕ Sincronización

## TPV LOCAL (Offline-first)

App Electron (Windows/Linux/Mac) o PWA Instalable

- Interfaz táctil optimizada

- Funciona 100% sin internet

- Pantalla dual cliente (opcional)

Base de Datos Local

- MongoDB Embedded o IndexedDB + Dexie.js

- Productos, clientes, configuración sincronizados

- Cola de ventas pendientes de subir

Hardware Integrado		
- Impresora térmica (tickets)		
- Lector código de barras / QR		
- Cajón de dinero / CashKeeper		
- Báscula		
- Pantalla dual (display cliente)		
- Terminal de pago (Redsys, SumUp, etc)		

## 2.2 Multi-tenant Strategy

- **Tenant = Empresa:** Cada empresa es un tenant independiente
- **Segregación de datos:** Todas las queries filtran por `empresaId`
- **Base de datos compartida:** Todas las empresas en MongoDB Atlas
- **Aislamiento total:** Middleware verifica tenant en cada request
- **Escalabilidad:** Sharding por `empresaId` cuando sea necesario

## 2.3 Sincronización TPV ↔ Cloud

### Datos que bajan del cloud al TPV:

- ☒ Productos completos (con variantes, precios, stock)
- ☒ Familias y categorías
- ☒ Clientes frecuentes (últimos 6 meses)
- ☒ Formas de pago
- ☒ Configuración del TPV (impuestos, impresora, etc)
- ☒ Promociones activas

### Datos que suben del TPV al cloud:

- ☒ Ventas realizadas
- ☒ Movimientos de caja
- ☒ Movimientos de stock
- ☒ Nuevos clientes creados en TPV

### Estrategia de sincronización:

javascript

```
// Sincronización inteligente
{
  // Automática cada X minutos (configurable: 5, 10, 15, 30 min)
  autoSync: true,
  intervalo: 10, // minutos

  // Sincronización al iniciar TPV
  syncOnOpen: true,

  // Sincronización al cerrar caja
  syncOnClose: true,

  // Sincronización manual on-demand
  manualSync: true,

  // Resolución de conflictos
  resolucion: "servidor-gana", // o "timestamp" o "manual"

  // Cola de operaciones
  cola: [
    { tipo: "venta", id: "...", timestamp: "...", data: {...} },
    { tipo: "cliente", id: "...", timestamp: "...", data: {...} }
  ],

  // Retry automático si falla
  maxReintentos: 3,
  tiempoEspera: 30000 // 30 segundos
}
```

---

## 3. STACK TECNOLÓGICO {#3-stack-tecnológico}

### 3.1 Frontend

Tecnología	Uso
Next.js 14+	Framework principal (App Router)
React 18+	UI Library
TypeScript	Tipado estático
Tailwind CSS	Estilos
shadcn/ui	Componentes UI base
Lucide React	Iconos
React Hook Form	Gestión formularios
Zod	Validación schemas
Zustand	State management
SWR	Data fetching y cache
Recharts	Gráficos y visualizaciones
React-PDF	Generación PDFs
date-fns	Manejo fechas
dnd-kit	Drag & drop (mesas, tareas)
Socket.io-client	Real-time

### 3.2 Backend

Tecnología	Uso
Node.js 20+	Runtime
Next.js API Routes	Endpoints REST
Next.js Server Actions	Mutations
MongoDB	Base de datos principal
Mongoose	ODM
NextAuth.js v5	Autenticación
Socket.io	WebSockets real-time
Bull / BullMQ	Cola de trabajos
Resend	Email transaccional
Twilio	SMS y WhatsApp
AWS S3 / Cloudinary	Almacenamiento archivos
Redis	Cache (opcional)

### 3.3 TPV Local

Tecnología	Uso
Electron	App desktop multiplataforma
MongoDB Embedded	BD local (alternativa: IndexedDB)
Dexie.js	Wrapper IndexedDB
node-thermal-printer	Impresión tickets
@zxing/library	Lectura códigos barras/QR
serialport	Hardware (cajón, báscula)

### 3.4 IA y Automatización

Tecnología	Uso
OpenAI GPT-4	Asistente IA, comandos de voz
Anthropic Claude	Análisis de documentos
Tesseract.js	OCR (escanear facturas)
Web Speech API	Reconocimiento de voz
TensorFlow.js	Predicciones de stock

### 3.5 DevOps

Tecnología	Uso
Vercel	Hosting Next.js
MongoDB Atlas	Database cloud
Cloudflare	CDN y protección DDoS
GitHub Actions	CI/CD
Sentry	Monitoreo de errores
Vercel Analytics	Analytics

## 4. SISTEMA DE LICENCIAS Y PLANES {#4-licencias}

### 4.1 Modelo de Negocio

SaaS (Software as a Service) con suscripción mensual/anual.


### 4.2 Planes Disponibles

 **PLAN DEMO (30 días gratis)**

**Precio:** €0

**Características:**



-  Todas las funcionalidades
  -  2 usuarios simultáneos
  -  100 documentos durante trial
  -  500 productos
  -  1 almacén
  -  1 TPV
  -  500 MB almacenamiento
  -  Marca de agua en documentos
- 

## **STARTER**

**Precio:** €29/mes o €290/año

**Límites:**

-  3 usuarios simultáneos (5 cuentas)
-  500 facturas/mes
-  1.000 productos
-  1 almacén
-  500 clientes
-  1 TPV
-  2 GB
-  500 emails/mes

**Módulos:** Ventas, Compras, Inventario básico, TPV, Reportes básicos

---

## **PROFESSIONAL**

**Precio:** €79/mes o €790/año

**Límites:**

- 👤 10 usuarios simultáneos (20 cuentas)
- 📄 2.000 facturas/mes
- 📦 10.000 productos
- 📦 3 almacenes
- 💻 3 TPVs
- 💾 10 GB
- 📧 2.000 emails/mes
- 📱 500 SMS/mes

**Módulos:** Todo Starter + Inventario avanzado, CRM, Proyectos, RRHH, Chat interno, API REST

---

## 💎 BUSINESS

**Precio:** €149/mes o €1.490/año

**Límites:**

- 👤 25 usuarios simultáneos (50 cuentas)
- 📄 Facturas ilimitadas
- 📦 Productos ilimitados
- 📦 Almacenes ilimitados
- 💻 10 TPVs
- 💾 50 GB
- 📧 10.000 emails/mes
- 📱 2.000 SMS/mes








**Módulos:** Todo Professional + Todos los módulos verticales (Taller, Informática, Restauración)

---

## 🏢 ENTERPRISE

**Precio:** Personalizado

**Características:**







-  Todo ilimitado
-  On-premise opcional
-  White-label (marca blanca)
-  Gestor de cuenta dedicado
-  SLA garantizado 99.9%
-  Desarrollos personalizados
-  Soporte 24/7

### 4.3 Add-ons (Módulos Extra)




Add-on	Precio/mes
Módulo Taller Mecánico	+€20
Módulo Informática/Tickets	+€20
Módulo Restauración	+€30
Módulo Proyectos Avanzado	+€20
TPV adicional	+€10/TPV
Usuario adicional	+€5/usuario
Almacenamiento extra (10GB)	+€5
API avanzada	+€30
White-label	+€100

### 4.4 Control de Límites

El sistema controla en tiempo real:

-  Usuarios simultáneos conectados
-  Documentos creados en el período
-  Productos en catálogo
-  Almacenamiento usado
-  Llamadas a API
-  Emails/SMS enviados

**Comportamiento al alcanzar límite:**

-  80% del límite → Alerta amarilla
  -  90% del límite → Alerta naranja
  -  100% del límite → Bloqueo de creación + sugerencia upgrade
- 










## 5. MÓDULOS DEL SISTEMA {#5-módulos}

### 5.1 Autenticación y Usuarios

#### Funcionalidades:

- Login con email/password
- Registro de empresas
- Recuperación de contraseña
- Autenticación 2FA (opcional)
- Sesiones con JWT
- Roles y permisos granulares

#### Roles predefinidos:

-  Super Admin (multi-empresa, solo para plataforma)
-  Administrador
-  Gerente
-  Vendedor
-  Camarero (restauración)
-  Técnico/Mecánico
-  Soporte IT
-  Almacenero
-  Visualizador (solo lectura)

#### Permisos granulares por módulo:

```
javascript
```

```
{  
  modulo: "ventas",  
  permisos: {  
    presupuestos: { ver: true, crear: true, editar: true, eliminar: false, aprobar: false },  
    facturas: { ver: true, crear: false, editar: false, eliminar: false }  
  }  
}
```

---

## 5.2 Clientes y Proveedores

### Clientes

#### Campos:

- Tipo: Particular, Empresa, Autónomo
- Nombre/Razón Social, NIF/CIF/DNI
- Emails (múltiples), Teléfonos (múltiples)
- Direcciones (facturación, envío, otras)
- Persona de contacto
- Categoría (Minorista, Mayorista, VIP, etc)
- Forma de pago predeterminada
- Plazo de pago (días)
- Límite de crédito
- Descuento general (%)
- Tarifa de precios asignada
- Observaciones
- Documentos adjuntos
- Estado (Activo/Inactivo/Bloqueado)

#### Campos específicos por negocio:

- **Taller:** Vehículos asociados, preferencias contacto
- **Informática:** Equipos, contratos, licencias
- **Restauración:** Alergias, preferencias mesa

## Funcionalidades:

- CRUD completo
- Importación masiva (Excel, CSV)
- Exportación
- Fusión de duplicados
- Historial completo de interacciones
- Cuentas por cobrar (aging report)
- Portal del cliente (autoservicio)

## Proveedores

Similar a clientes pero para compras.

---

## 5.3 Productos y Escandallos/BOM

### 5.3.1 Productos Base

#### Tipos de producto:

1. **Producto simple:** Artículo individual vendible
2. **Producto con variantes:** Ej: Camiseta (tallas y colores)
3. **Producto compuesto (BOM/Escandallo):** Se fabrica/ensambla a partir de otros productos
4. **Servicio:** No tiene stock físico
5. **Materia prima:** Solo para comprar, no para vender directo

#### Campos principales:

- SKU único
- EAN/Código de barras
- Referencia proveedor
- Nombre
- Descripción corta/larga
- Familia/Categoría (jerárquica)
- Marca
- Proveedor habitual
- Unidad de medida (ud, kg, litros, m, etc)
- Tipo de IVA
- Control de stock (Sí/No)
- Estado (Activo, Descatalogado, Próximamente)

#### **Precios:**

- Precio de compra (coste)
- Precio de venta (PVP)
- Tarifas personalizadas (por cliente o categoría)
- Margen de beneficio (%)
- Historial de precios

#### **Stock:**

- Stock actual (por almacén)
- Stock mínimo
- Stock máximo
- Punto de pedido
- Stock reservado (pedidos pendientes)
- Stock disponible (calculado)
- Ubicación en almacén

### **5.3.2 Sistema de Variantes**

#### **Configuración de atributos:**

javascript

```
{
  producto: "Camiseta Básica",
  tieneVariantes: true,
  atributos: [
    {
      nombre: "Talla",
      valores: ["XS", "S", "M", "L", "XL", "XXL"],
      tipoVisualizacion: "botones" // o "dropdown"
    },
    {
      nombre: "Color",
      valores: [
        { valor: "Blanco", hexColor: "#FFFFFF" },
        { valor: "Negro", hexColor: "#000000" },
        { valor: "Azul", hexColor: "#0000FF" },
        { valor: "Rojo", hexColor: "#FF0000" }
      ],
      tipoVisualizacion: "colores"
    }
  ],
  // Generación automática de todas las combinaciones
  variantes: [
    {
      sku: "CAM-BAS-XS-BLC",
      combinacion: { talla: "XS", color: "Blanco" },
      stock: 15,
      precioExtra: 0,
      imagenes: ["url1.jpg"]
    },
    {
      sku: "CAM-BAS-S-BLC",
      combinacion: { talla: "S", color: "Blanco" },
      stock: 25,
      precioExtra: 0,
      imagenes: ["url1.jpg"]
    }
    // ... resto de combinaciones (24 variantes en total)
  ]
}
```

**Generador automático:** El sistema genera todas las combinaciones posibles automáticamente.

### 5.3.3 Sistema Universal de Escandallos/BOM



**Bill of Materials (BOM)** = Lista de materiales/componentes necesarios para fabricar o ensamblar un producto final.

**Estructura:**

javascript

```
{
  productoFinal: {
    id: ObjectId,
    nombre: "Hamburguesa Clásica",
    tipo: "producto_compuesto",
    precioVenta: 8.50
  },

  // Lista de componentes
  componentes: [
    {
      productoId: ObjectId,
      nombre: "Pan de hamburguesa",
      cantidad: 1,
      unidad: "ud",
      costoUnitario: 0.40,
      costoTotal: 0.40,
      merma: 0 // % de desperdicio
    },
    {
      productoId: ObjectId,
      nombre: "Carne de ternera",
      cantidad: 150,
      unidad: "g",
      costoUnitario: 0.012, // por gramo
      costoTotal: 1.80,
      merma: 5 // 5% de merma en cocción
    },
    {
      productoId: ObjectId,
      nombre: "Queso cheddar",
      cantidad: 2,
      unidad: "lonchas",
      costoUnitario: 0.25,
      costoTotal: 0.50,
      merma: 0
    },
    {
      productoId: ObjectId,
      nombre: "Lechuga",
      cantidad: 30,
      unidad: "g",
      costoUnitario: 0.003,
      costoTotal: 0.09,
      merma: 10 // 10% de desperdicio en lavado
```

```
merma: 10 // 10% de hojas no aprovechables
},
{
  productoId: ObjectId,
  nombre: "Tomate",
  cantidad: 50,
  unidad: "g",
  costoUnitario: 0.004,
  costoTotal: 0.20,
  merma: 15
},
{
  productoId: ObjectId,
  nombre: "Salsa especial",
  cantidad: 30,
  unidad: "ml",
  costoUnitario: 0.015,
  costoTotal: 0.45,
  merma: 5
}
],

// Cálculos automáticos
costoTotalComponentes: 3.44,
mermaTotal: 0.21, // Suma de mermas
costoReal: 3.65, // Componentes + merma
precioVenta: 8.50,
margenBruto: 4.85,
porcentajeMargen: 57.06,

// Información adicional
tiempoPreparacion: 8, // minutos
dificultad: "media",
alergenos: ["gluten", "lactosa"],

// Para restauración
categoria: "Hamburguesas",
disponibleEnCarta: true,

// Control de stock automático
descontarStock: true, // Al vender 1 hamburguesa, descuenta todos los componentes

// Notas
instruccionesPreparacion: "Cocinar carne a término medio...",
observaciones: "Incluye patatas fritas como guarnición"
}
```

## Restauración negocio:

javascript

*// Ejemplo: Mojito*

```
{
  productoFinal: "Mojito",
  componentes: [
    { producto: "Ron blanco", cantidad: 50, unidad: "ml", costo: 0.75 },
    { producto: "Lima", cantidad: 1, unidad: "ud", costo: 0.30 },
    { producto: "Hierbabuena", cantidad: 10, unidad: "hojas", costo: 0.10 },
    { producto: "Azúcar", cantidad: 15, unidad: "g", costo: 0.02 },
    { producto: "Soda", cantidad: 100, unidad: "ml", costo: 0.15 },
    { producto: "Hielo", cantidad: 150, unidad: "g", costo: 0.05 }
  ],
  costoTotal: 1.37,
  precioVenta: 7.00,
  margen: 5.63
}
```

## INFORMÁTICA:

javascript

*// Ejemplo: PC Gaming Custom*

```
{
  productoFinal: "PC Gaming Pro 2025",
  componentes: [
    { producto: "Placa Base ASUS ROG", cantidad: 1, unidad: "ud", costo: 180 },
    { producto: "Intel i7-14700K", cantidad: 1, unidad: "ud", costo: 380 },
    { producto: "RAM DDR5 32GB", cantidad: 2, unidad: "ud", costo: 120 },
    { producto: "RTX 4070", cantidad: 1, unidad: "ud", costo: 650 },
    { producto: "SSD 1TB NVMe", cantidad: 1, unidad: "ud", costo: 90 },
    { producto: "Fuente 850W", cantidad: 1, unidad: "ud", costo: 120 },
    { producto: "Torre ATX", cantidad: 1, unidad: "ud", costo: 80 },
    { producto: "Ventiladores RGB", cantidad: 3, unidad: "ud", costo: 15 }
  ],
  manoDeObra: { descripcion: "Ensamblaje y testeo", horas: 2, tarifaHora: 30, costo: 60 },
  costoTotal: 1755,
  precioVenta: 2299,
  margen: 544
}
```

## TALLER MECÁNICO:

javascript

*// Ejemplo: Cambio de aceite completo*

```
{
  productoFinal: "Cambio de aceite 5W30",
  componentes: [
    { producto: "Aceite 5W30", cantidad: 5, unidad: "litros", costo: 6.50 },
    { producto: "Filtro de aceite", cantidad: 1, unidad: "ud", costo: 8.00 },
    { producto: "Junta carter", cantidad: 1, unidad: "ud", costo: 2.50 }
  ],
  manoDeObra: { descripcion: "Cambio de aceite", horas: 0.5, tarifaHora: 45, costo: 22.50 },
  costoTotal: 55.50,
  precioVenta: 89.90,
  margen: 34.40
}
```

## PRODUCCIÓN/MANUFACTURA:

javascript

*// Ejemplo: Mesa de madera artesanal*

```
{
  productoFinal: "Mesa comedor roble 180x90cm",
  componentes: [
    { producto: "Tablero roble macizo", cantidad: 1, unidad: "ud", costo: 180 },
    { producto: "Pata torneada roble", cantidad: 4, unidad: "ud", costo: 35 },
    { producto: "Tornillos 6x60", cantidad: 20, unidad: "ud", costo: 0.15 },
    { producto: "Cola de carpintero", cantidad: 200, unidad: "ml", costo: 0.05 },
    { producto: "Barniz satinado", cantidad: 250, unidad: "ml", costo: 0.12 },
    { producto: "Lija grano 120", cantidad: 3, unidad: "hojas", costo: 1.20 }
  ],
  manoDeObra: { descripcion: "Corte, ensamblaje, lijado y barnizado", horas: 6, tarifaHora: 25, costo: 150 },
  costoTotal: 514.50,
  precioVenta: 899.00,
  margen: 384.50
}
```

## Funcionalidades del sistema BOM/Escandallos:

- ☒ Creación de recetas/componentes ilimitados
- ☒ Cálculo automático de costes con mermas
- ☒ Descuento automático de stock al vender producto compuesto
- ☒ Alertas si falta algún componente
- ☒ Histórico de cambios de receta
- ☒ Cálculo de rentabilidad en tiempo real
- ☒ Simulador de precios (ajustar PVP según coste)
- ☒ Exportación de recetas a PDF
- ☒ Control de alérgenos (restauración)

### 5.3.4 Familias y Categorías

Estructura **jerárquica ilimitada**:

Electrónica



**Configuración por familia:**

- Imagen representativa
- Atributos personalizados
- Descuentos automáticos
- Comisiones específicas
- Plantilla de producto (campos predefinidos)

## 5.4 Ciclo Comercial Completo (Ventas y Compras)






**Flujo estándar:** Presupuesto → Pedido → Albarán → Factura

### 5.4.1 PRESUPUESTOS

#### **Funcionalidades:**

- Creación rápida desde plantillas
- Añadir líneas de productos (con variantes)
- Añadir líneas de servicios
- Añadir líneas de texto libre
- Descuentos por línea
- Descuento general sobre total
- Validez del presupuesto (días)
- Conversión a pedido con 1 clic
- Envío por email automático
- Versiones (v1, v2, v3...)
- Duplicar presupuesto
- Comparador de presupuestos (en compras)

#### **Estados:**

-  Borrador
-  Enviado
-  Aceptado
-  Rechazado
-  Caducado

#### **Documento PDF generado:**








- Logo empresa
- Datos fiscales
- Datos cliente
- Líneas detalladas
- Totales con IVA desglosado
- Condiciones de pago
- Validez
- Firma digital (opcional)

### 5.4.2 PEDIDOS

#### Características adicionales:

- Fecha de entrega estimada
- Prioridad (Baja, Normal, Alta, Urgente)
- Reserva automática de stock
- Estados de preparación
- División en varios albaranes (entregas parciales)
- Control de cantidad servida vs pendiente
- Notificaciones automáticas al cliente
- Integración con transportistas

#### Estados:

-  Pendiente
-  Confirmado
-  En preparación
-  Preparado
-  Enviado
-  Entregado
-  Cancelado

### 5.4.3 ALBARANES (Entregas)

#### Funcionalidades:



- Generación desde pedido
- Entregas parciales (múltiples albaranes por pedido)
- Control de cantidades
- Geolocalización de entrega
- Firma digital del receptor
- Fotos de la entrega
- Registro de incidencias
- Integración con rutas de reparto

**Información adicional:**

- Transportista
- Número de seguimiento
- Número de bultos
- Peso total
- Fecha y hora de entrega
- Receptor (nombre, DNI, firma)

#### **5.4.4 FACTURAS**






**Tipos de factura:**

1. **Factura normal**
2. **Factura simplificada** (< 400€)
3. **Factura rectificativa**
4. **Factura proforma**
5. **Abono / Nota de crédito**

**Características:**

- Generación desde albaranes o pedidos
- Facturación masiva (seleccionar múltiples albaranes)
- Series de numeración configurables
- Vencimientos múltiples
- Desglose de IVA por tipos
- Recargo de equivalencia
- Retención IRPF
- Estados de pago
- Integración contabilidad
- Envío automático SII (Suministro Inmediato de Información)
- Cumplimiento TicketBAI / Verifactu

#### Estados financieros:

-  Emitida
-  Pagada parcialmente
-  Pagada
-  Vencida
-  Impagada

#### Control de vencimientos:

```
javascript

{
  factura: "FAC-2025-00123",
  total: 1200,
  vencimientos: [
    { fecha: "2025-11-03", importe: 400, pagado: true, fechaPago: "2025-11-01", metodo: "transferencia" },
    { fecha: "2025-12-03", importe: 400, pagado: false },
    { fecha: "2026-01-03", importe: 400, pagado: false }
  ],
  totalPagado: 400,
  totalPendiente: 800
}
```

---

## 5.5 TPV LOCAL OFFLINE-FIRST

### 5.5.1 Características Principales

#### Aplicación independiente:

- Desktop: Electron (Windows, Mac, Linux)
- Alternativa: PWA instalable avanzada
- Interfaz táctil optimizada
- Funciona 100% sin conexión a internet
- Sincronización automática al reconectar

#### Pantalla TPV:

BÚSQUEDA Y PRODUCTOS				CARRITO / TICKET			
[ 🔍 Buscar producto/código ]				Cliente: Juan Pérez			
[ Bebidas ]		[ Comidas ]		[ Postres ]		Hamburguesa x2 17.00€	
				Coca-Cola x1 2.50€			
				Café x1 1.80€			
[ Producto 1 ]		[ Producto 2 ]		Subtotal:		21.30€	
[ Producto 3 ]		[ Producto 4 ]		Descuento:		-2.13€	
[ Producto 5 ]		[ Producto 6 ]		IVA (10%):		1.92€	
[ Más productos... ]				TOTAL:		21.09€	
				[ 🗑️ ] [ 👤 ] [ 💾 ] [ 💳 ]			
				[ COBRAR (F12) ]			

#### Funcionalidades del TPV:

- ☒ Búsqueda rápida (código, nombre, EAN)
- ☒ Escaneo de código de barras
- ☒ Selección de variantes (talla, color)
- ☒ Modificadores de productos (restauración)
- ☒ Calculadora integrada
- ☒ Cambio de precio (con permisos)
- ☒ Descuentos por línea
- ☒ Descuento general
- ☒ Selección de cliente
- ☒ Múltiples formas de pago en una venta
- ☒ División de pago (pagar entre varios)
- ☒ Tickets en espera (aparcar venta)
- ☒ Devoluciones
- ☒ Apertura y cierre de caja
- ☒ Movimientos de efectivo (entrada/salida)
- ☒ Impresión de ticket
- ☒ Envío de ticket por email
- ☒ Pantalla dual para cliente (opcional)

#### **Atajos de teclado:**

- **(F1)** - Ayuda
- **(F2)** - Buscar producto
- **(F3)** - Buscar cliente
- **(F4)** - Descuento
- **(F5)** - Aparcar ticket
- **(F6)** - Recuperar ticket
- **(F12)** - Cobrar
- **(ESC)** - Cancelar
- **(+)** / **(-)** - Cantidad

## 5.5.2 Base de Datos Local

### Opción 1: MongoDB Embedded

```
javascript

// Colecciones locales
{
  productos: [...], // Sincronizados desde cloud
  clientes: [...], // Clientes frecuentes
  ventas_pendientes: [], // Cola para subir
  configuracion: {...}, // Config del TPV
  caja_actual: {...} // Estado de caja abierta
}
```

### Opción 2: IndexedDB + Dexie.js

```
javascript

const db = new Dexie("TPV_Database");
db.version(1).stores({
  productos: 'id, sku, nombre, ean',
  clientes: 'id, nombre, nif',
  ventas: '++id, fecha, total, sincronizado',
  configuracion: 'clave'
});
```

## 5.5.3 Gestión de Caja

### Apertura de caja:

```
javascript

{
  fecha: "2025-10-03T08:00:00Z",
  usuario: "María García",
  tpv: "TPV-001",
  efectivoInicial: 200.00,
  estado: "abierta"
}
```

### Movimientos durante el día:

- Ventas (con detalle forma de pago)
- Entradas de efectivo (reposición)
- Salidas de efectivo (gastos, cambio)

### Cierre de caja:

```
javascript

{
  fechaCierre: "2025-10-03T22:00:00Z",
  resumenVentas: {
    numVentas: 87,
    totalBruto: 2847.50
  },
  resumenFormasPago: [
    { formaPago: "Efectivo", numOperaciones: 45, importe: 1234.50 },
    { formaPago: "Tarjeta", numOperaciones: 38, importe: 1523.00 },
    { formaPago: "Bizum", numOperaciones: 4, importe: 90.00 }
  ],
  efectivoTeoricoFinal: 1434.50,
  efectivoRealFinal: 1430.00,
  diferencia: -4.50,
  observaciones: "Falta 4.50€ por error en cambio"
}
```

### 5.5.4 Hardware Integrado

#### Impresora térmica de tickets:

- Conexión: USB, Serial, Ethernet, Bluetooth
- Tamaño: 58mm o 80mm
- Velocidad: Alta para no esperas
- Auto-corte de papel

#### Lector de código de barras:

- 1D y 2D (QR)
- Conexión: USB (emula teclado)
- Escaneo automático

#### Cajón de dinero:

- Apertura automática al cobrar efectivo
- Apertura manual con permiso
- Conexión vía impresora térmica

#### **CashKeeper / Gestoras de efectivo:**

- Integración para contar billetes/monedas
- Validación automática
- Reducción de errores

#### **Báscula:**

- Para productos a peso (frutería, charcutería)
- Conexión Serial/USB

#### **Pantalla dual:**

- Monitor secundario mirando al cliente
- Muestra productos, precios, total
- Publicidad entre ventas

#### **Terminal de pago (TPV físico):**

- Redsys, SumUp, iZettle
  - Conexión por API o manual
- 

## **5.6 Inventario Multi-almacén**

### **5.6.1 Almacenes**

#### **Configuración:**

javascript

```
{
  codigo: "ALM-001",
  nombre: "Almacén Central",
  tipo: "principal", // principal | secundario | transito | tienda
  direccion: {...},
  responsable: "Juan López",
  esTPV: false,
  zonas: [
    { codigo: "A1", nombre: "Estantería A - Nivel 1" },
    { codigo: "A2", nombre: "Estantería A - Nivel 2" },
    { codigo: "B1", nombre: "Cámara frigorífica" }
  ],
  activo: true
}
```

## 5.6.2 Movimientos de Inventario

### Tipos de movimiento:

1. **Entrada por compra:** Al recibir albarán de proveedor
2. **Salida por venta:** Al generar albarán de venta
3. **Traspaso entre almacenes:** Mover stock
4. **Ajuste de inventario:** Correcciones manuales
5. **Producción/Transformación:** Usar BOM (consumir componentes, crear producto final)
6. **Merma:** Pérdidas, roturas, caducidad
7. **Devolución cliente:** Entrada
8. **Devolución a proveedor:** Salida

### Registro de movimiento:



javascript

```
{
  fecha: "2025-10-03T14:30:00Z",
  tipo: "salida_venta",
  productoId: ObjectId,
  varianteId: "...",
  almacenOrigenId: ObjectId,
  almacenDestinoId: null,
  cantidad: 2,
  unidad: "ud",
  valorUnitario: 45.50,
  valorTotal: 91.00,
  metodoValoracion: "pmp", // precio medio ponderado
  lote: "L2025-034",
  numeroSerie: null,
  fechaCaducidad: null,
  documentoTipo: "albaran_venta",
  documentoId: ObjectId,
  documentoNumero: "ALB-2025-00456",
  ubicacion: "A1-B3",
  motivo: "Venta a cliente",
  usuarioId: ObjectId,
  aprobado: true
}
```

### 5.6.3 Control de Stock

**Stock por almacén:**

javascript

```
{
  productoId: ObjectId,
  stockPorAlmacen: [
    {
      almacenId: ObjectId,
      almacen: "Almacén Central",
      stock: 150,
      stockReservado: 12,
      stockDisponible: 138,
      ubicacion: "A1-B3",
      stockMinimo: 20,
      stockMaximo: 200
    },
    {
      almacenId: ObjectId,
      almacen: "Tienda Centro",
      stock: 25,
      stockReservado: 3,
      stockDisponible: 22,
      ubicacion: "Estante 5"
    }
  ],
  stockTotalGlobal: 175,
  stockDisponibleGlobal: 160
}
```

### Alertas automáticas:

- 🟡 Stock por debajo del mínimo
- 🔴 Stock a cero
- ⚠️ Stock negativo (error)
- 📅 Productos próximos a caducar (7, 15, 30 días)

### 5.6.4 Inventario Físico (Recuento)

#### Proceso:

1. Crear recuento (seleccionar almacén y productos)
2. Congelar movimientos durante recuento (opcional)
3. Ingresar cantidades físicas reales
4. Comparar con stock teórico
5. Generar diferencias
6. Aprobar y ajustar stock

**Tipos de recuento:**

- **Completo:** Todos los productos del almacén
  - **Parcial:** Solo una familia o zona
  - **Rotativo:** Un grupo de productos cada día
- 

## **5.7 MÓDULO RESTAURACIÓN / HOSTELERÍA**

### **5.7.1 Gestión de Salones y Mesas**

**Diseñador visual de mesas (drag & drop):**

javascript

```
{
  salonId: ObjectId,
  nombre: "Salón Principal",
  layout: {
    ancho: 1200, // px
    alto: 800,
    mesas: [
      {
        id: "mesa-01",
        numero: 1,
        tipo: "rectangular", // rectangular | redonda | cuadrada
        capacidad: 4,
        posX: 100,
        posY: 100,
        ancho: 120,
        alto: 80,
        estado: "libre", // libre | ocupada | reservada | sucia
        camarero: null,
        horaOcupacion: null
      },
      {
        id: "mesa-02",
        numero: 2,
        tipo: "redonda",
        capacidad: 6,
        posX: 300,
        posY: 150,
        radio: 70,
        estado: "ocupada",
        camarero: "María",
        horaOcupacion: "2025-10-03T14:30:00Z",
        comensales: 5,
        ticketId: ObjectId
      }
    ],
    // ... más mesas
  },
  elementos: [
    { tipo: "barra", posX: 50, posY: 50, ancho: 300, alto: 80 },
    { tipo: "puerta", posX: 600, posY: 0 },
    { tipo: "ventana", posX: 0, posY: 200 }
  ]
}
```

## Funcionalidades de mesas:

- ☒ Crear layout visual (arrastrar/soltar)
- ☒ Cambiar estado de mesa con un clic
- ☒ Unir mesas (para grupos grandes)
- ☒ Transferir mesa a otro camarero
- ☒ Ver tiempo de ocupación
- ☒ Códigos de color por estado
- ☒ Mapa de calor (mesas más rentables)

## Reservas de mesas:

javascript

```
{  
  fecha: "2025-10-05",  
  hora: "21:00",  
  comensales: 4,  
  cliente: "Laura Martínez",  
  telefono: "+34666777888",  
  email: "laura@email.com",  
  mesaAsignada: "mesa-05",  
  observaciones: "Cumpleaños, preparar postre",  
  estado: "confirmada", // pendiente | confirmada | cancelada | completada  
  recordatorioEnviado: true  
}
```

## 5.7.2 Sistema de Comandas

### Flujo de comanda:

1. Camarero toma pedido en mesa
2. Envía comanda a cocina
3. Cocina recibe en impresora o KDS
4. Prepara platos
5. Marca como listo
6. Camarero sirve
7. Cliente paga

**Estructura de comanda:**

javascript

```
{
  numero: "CMD-2025-00523",
  fecha: "2025-10-03T14:35:00Z",
  mesa: "Mesa 7",
  salon: "Terraza",
  camarero: "Pedro García",
  comensales: 3,

  lineas: [
    {
      producto: "Ensalada César",
      cantidad: 2,
      precio: 8.50,
      modificadores: ["Sin anchoas", "Extra parmesano"],
      alergenos: ["gluten", "lactosa", "pescado"],
      notasCocina: "Cliente alérgico a frutos secos",
      zona: "cocina_fria",
      estado: "pendiente", // pendiente | preparando | listo | servido
      prioridad: "normal",
      horaEnvio: "2025-10-03T14:35:10Z",
      horaInicio: null,
      horaFin: null
    },
    {
      producto: "Solomillo al punto",
      cantidad: 1,
      precio: 22.00,
      modificadores: ["Al punto", "Sin guarnición"],
      zona: "cocina_caliente",
      estado: "preparando",
      cocinero: "Chef Antonio"
    },
    {
      producto: "Cerveza Estrella",
      cantidad: 3,
      precio: 2.50,
      zona: "barra",
      estado: "servido"
    }
  ],

  estado: "en_curso", // abierta | en_curso | completada | pagada
  total: 32.50
}
```

## Modificadores de productos: Configurables por producto:

javascript

```
{
  producto: "Hamburguesa",
  modificadoresDisponibles: [
    {
      grupo: "Punto de carne",
      opciones: ["Poco hecha", "Al punto", "Muy hecha"],
      obligatorio: true,
      precioExtra: 0
    },
    {
      grupo: "Extras",
      opciones: [
        { nombre: "Extra queso", precio: 1.00 },
        { nombre: "Extra bacon", precio: 1.50 },
        { nombre: "Aguacate", precio: 2.00 }
      ],
      obligatorio: false,
      multiple: true
    },
    {
      grupo: "Sin ingredientes",
      opciones: ["Sin cebolla", "Sin tomate", "Sin lechuga"],
      precioExtra: 0,
      multiple: true
    }
  ]
}
```

### 5.7.3 Kitchen Display System (KDS)

**Pantalla táctil en cocina** que reemplaza tickets impresos:



COCINA CALIENTE

Pedidos: 8

Tiempo promedio: 12m

Mesa 3

15:32

2x Solomillo

AL PUNTO

1x Lubina

a la sal

Nota: Mesa

celebración

[ INICIAR ]

Mesa 7

15:35

1x Entrecot

MUY HECHO

1x Merluza

[ EN PROCESO ]

Mesa 12

15:37

3x Pollo

asado

[ LISTO ]

Mesa 15

15:33

Llevar #34

15:36

Código de colores por tiempo:

- Verde: < 10 minutos
- Amarillo: 10-15 minutos
- Rojo: > 15 minutos (urgente)
- Negro: > 30 minutos (crítico)

Funcionalidades KDS:

- ☒ Visualización en tiempo real
- ☒ Ordenación automática por prioridad/tiempo
- ☒ Táctil (iniciar, finalizar platos)
- ☒ Avisos sonoros en platos urgentes
- ☒ Filtros por zona (caliente, fría, postres)
- ☒ Estadísticas de tiempos
- ☒ Bump bar (botón físico para marcar listo)

**Impresoras por zona:** Alternativa o complemento al KDS:

- Cocina caliente: Carnes, pescados, arroces
- Cocina fría: Ensaladas, entrantes fríos
- Barra: Bebidas
- Postres: Postres y cafés

Cada zona recibe solo sus comandas.

#### 5.7.4 Tipos de Servicio

##### 1. Servicio en mesa tradicional:

- Camarero asignado por sección
- Comandas desde tablet o TPV
- Pago en mesa o en caja

##### 2. Fast Food / Para llevar:

- TPV de mostrador
- Número de pedido
- Preparación rápida
- Entrega en ventanilla

##### 3. Delivery:

- Pedidos online (web/app)
- Integración con Glovo, Uber Eats, Just Eat
- Asignación de repartidor
- Tracking en tiempo real

#### **4. Autoservicio / Buffet:**

- Cliente se sirve
- Pago por peso o precio fijo
- Control de salidas

#### **5.7.5 Menús y Carta Digital**

**Menú del día:**

javascript

```
{
  nombre: "Menú del día",
  precio: 12.50,
  disponible: ["lunes", "martes", "miercoles", "jueves", "viernes"],
  horario: { inicio: "13:00", fin: "16:00" },

  estructura: [
    {
      seccion: "Primeros",
      elige: 1,
      opciones: [
        "Ensalada mixta",
        "Lentejas estofadas",
        "Sopa del día",
        "Pasta carbonara"
      ]
    },
    {
      seccion: "Segundos",
      elige: 1,
      opciones: [
        "Pollo asado",
        "Merluza a la plancha",
        "Solomillo de cerdo",
        "Tortilla española"
      ]
    },
    {
      seccion: "Postres",
      elige: 1,
      opciones: [
        "Fruta del tiempo",
        "Flan casero",
        "Helado",
        "Yogur"
      ]
    },
    {
      seccion: "Bebida",
      incluido: true,
      opciones: ["Agua", "Vino", "Cerveza", "Refresco"]
    }
  ],
  i: 1,
  nombre: "Domingo 14 de mayo"
```

```
incluye: "Pan y postre",
observaciones: "IVA incluido"
}
```

### Carta digital con QR:

- Cliente escanea QR en mesa
- Ve carta actualizada en tiempo real
- Filtros por alérgenos
- Fotos de platos
-