# Plan SCRUM Definitivo - Sistema

#### **Tortillería**

- *Objetivo:* Crear un sistema **offline-first**, instalable en Windows, con arquitectura sólida y soporte a largo plazo.
- **Duración total:** 7 sprints (1 sprint 0 + 6 sprints de desarrollo)
- Metodología: SCRUM mejorado con buffer QA/retrabajo y flujo de valor IA.

### SPRINT 0: Análisis & Diseño Base (5 días)

- of "Cimentar arquitectura y backlog con IA y Codex"
- Actividades clave:
  - Refinar Casos de Uso e Historias de Usuario
    - o Revisar todo lo que ya tenemos.
    - Validación con IA:
      - GPT-4o → Flujos alternos.
      - Gemini → Gaps funcionales.
      - DeepSeek → Sugerencias sobre modelo de datos.
  - Crear Blueprint Arquitectónico:
    - o Diagrama C4 (Contexto, Contenedores, Componentes).
    - o Diagrama de base de datos relacional (PostgreSQL).
  - Backlog inicial ordenado por prioridad (MoSCoW).
  - Configuración inicial del repositorio Git + estructura carpetas Clean Architecture.
- **©** Entregables:
- ADR (Architecture Decision Record).
- Diagrama UML preliminar.
- Repo Git con estructura inicial.

### SPRINT 1: Setup Avanzado + Curso Node.js (7 días)

- 🎯 "Entorno listo y base técnica sólida"
- Actividades clave:
  - Curso Node.js + Express (10h)
    - o Día 1-3: CRUD básico con Postman (Usuarios).
    - o Día 4-5: JWT y autenticación roles.
    - o Día 6-7: Conexión PostgreSQL y testing básico.
  - Dev Container con Docker (Node.js + PostgreSQL + PgAdmin).
  - Configuración de ESLint + Prettier + Husky hooks.
- **f** Entregables:
- CRUD funcional (Usuarios).
- Entorno de desarrollo con Docker Compose.
- Manual rápido de despliegue local.

#### SPRINT 2: Backend Modular & Seguridad (7 días)

- o "Backend robusto con roles y auditoría"
- Actividades clave:
  - Implementar Clean Architecture:

#### src/

- -- core/ # Lógica negocio
- Infrastructure/ # DB/APIs externas
- application/ # Casos de uso
- presentation/ # Controllers REST
- Roles y permisos (RBAC).
- Seguridad:
  - Hashing con bcrypt (salt rounds=12).
  - o Rate limiting (express-rate-limit).
  - o Pruebas de penetración básica con ZAP Proxy.
- API documentada con Swagger.

- **©** Entregables:
- API REST completa con seguridad básica.
- Reporte inicial de seguridad.

### SPRINT 3: Frontend React + Sync Offline (7 días)

o "Interfaz role-centric y almacenamiento local"

#### Actividades clave:

- Frontend modular: React + TailwindCSS.
- Offline Engine con PouchDB para guardar datos localmente.
- Sincronización automática cuando haya conexión.
- Dashboards multi-rol (Dueño, Despachador, Motociclista).

#### **f** Entregables:

- Dashboard React conectado al backend.
- Sincronización de datos local ↔ servidor.

### SPRINT 4: Reporting & Packaging (7 días)

🎯 "Business Intelligence y empaquetado instalable"

#### Actividades clave:

- Reportes con Chart.js:
  - Ventas por hora/día.
  - o Merma materia prima.
  - Utilidad neta diaria.
- Empaquetado con Electron Builder:
  - o Generar instalador .exe + .msi.
  - o Configurar auto-updater.
- Testing en VM (Windows 10/11).

- **f** Entregables:
- App instalable para Windows.
- Reportes exportables a PDF.

### SPRINT 5: Testing Avanzado + IA Prompts (7 días)

- 🌀 "Pulir y automatizar el futuro con IA"
- Actividades clave:
  - Testing:
    - o 60% unitarias (Jest).
    - o 30% integración.
    - o 10% E2E (Cypress).
  - Crear **Prompt Library** para Codex y GPT:

##Prompt CRUD Inventario

- "Genera módulo Node.js con schema PostgreSQL y validación Joi"
- Validación de prompts con GPT-4o, DeepSeek, Gemini.
- **f** Entregables:
- Suite de pruebas con reporte cobertura.
- Librería de Prompts para mantenimiento futuro.

### SPRINT 6: Go-Live & Transferencia (7 días)

- 🌀 "Entrega cliente-céntrica y soporte extendido"
- Actividades clave:
  - Manual técnico (MkDocs).
  - Manual de usuario final (PDF pictográfico).

- Instalador final firmado digitalmente.
- Backup automático (DB → carpeta local + S3).

#### **©** Entregables:

- App auto-contenida en USB.
- ✓ Docker image para réplicas rápidas.
- Kit de documentación y capacitación.

#### Plan de Mitigación de Riesgos

Riesgo	Mitigación
Cambios de requisitos	Feature flags + Sprint buffer
Problemas de conexión	Offline-first con PouchDB
Baja adopción usuarios	Onboarding guiado (manual pictográfico)
Vulnerabilidades seguridad	Pentest mensual + Honeypot local

#### ■ KPI por Sprint

Sprint	KPI Clave
Sprint 1	CRUD usuarios en <3 días
Sprint 2	API completa con RBAC y <5% bugs críticos
Sprint 3	Dashboard offline con <2 seg respuesta
Sprint 4	Instalador Windows probado en VM
Sprint 5	95% cobertura pruebas unitarias
Sprint 6	Go-live exitoso + respaldo automático

## **Entrega Final**

- ✓ Sistema offline-first instalable en Windows.
- ✓ Librería de Prompts Codex validada.
- ✓ Docker image para despliegues rápidos.
- √ Soporte extendido con manuales técnicos y de usuario.