Plan de Trabajo con Sprints – Gestión de Botiquines (MVP)

Fecha de elaboración: Jueves 11 de septiembre de 2025

Fecha de término: Viernes 10 de octubre de 2025

Duración: 4 semanas (3 sprints de 7 días + 1 sprint final de 9 días con buffer)

# 1. Introducción y fundamento

Este plan organiza el desarrollo del MVP de Gestión de Botiquines en sprints semanales, con entregables claros y revisiones frecuentes. Se basa en el documento de requerimientos aprobado (alcance, funcionalidades y stack: Flask, Bootstrap, PostgreSQL, DigitalOcean).

Razonamiento de la cadencia: sprints semanales permiten ciclos cortos de retroalimentación, visibilidad temprana del progreso y un cierre seguro al 10 de octubre. El último sprint se extiende 9 días para incluir pruebas, ajustes y buffer de riesgo.

# 2. Alcance de referencia del MVP

Incluye: Dashboard funcional con visualización de inventario y alertas (caducidad y stock), gestión básica de medicamentos y autenticación de un solo rol. No incluye: integración hardware-software, múltiples roles, reportes avanzados o notificaciones externas.

# 3. Estructura de sprints y entregables

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Fechas | Objetivo | Entregables (Definición de Hecho) |
| Sprint 1 | 11–17 Sep 2025 | Base técnica y cimientos del sistema. | - Proyecto Flask creado con estructura modular. - PostgreSQL provisionado y esquema inicial (medicamentos). - Autenticación básica (rol único) funcionando. - README con instrucciones de ejecución local. - Despliegue inicial en DO (staging) funcionando. |
| Sprint 2 | 18–24 Sep 2025 | Inventario y pantalla inicial del dashboard. | - CRUD de medicamentos (crear/editar/eliminar/listar). - Dashboard inicial con listado de inventario. - Validaciones de campos obligatorios y fechas de caducidad. - Campos obligatorios: nombre comercial, genérico, concentración, fecha caducidad, cantidad. - Seed de datos realistas (Paracetamol, Ibuprofeno, etc.). |
| Sprint 3 | 25 Sep – 1 Oct 2025 | Alertas y visualizaciones claras. | - Alertas por caducidad: crítica (≤ 7 días), preventiva (8–30 días). - Alertas por stock: crítico (cantidad = 0), bajo (≤ umbral configurado). - Indicadores visuales (colores, etiquetas, barras simples). - Filtros por estado (disponible, bajo, agotado, próximos a vencer, vencidos). - Configuración básica de umbrales por medicamento. |
| Sprint 4 | 2–10 Oct 2025 | Cierre, pruebas y preparación de entrega. | - UI con Bootstrap (responsive y consistente). - Pruebas funcionales y corrección de errores. - Usuario demo + dataset de ejemplo robusto. - Documentación breve de uso y despliegue. - Demo final lista con casos de uso completos. |

# 4. Hitos y calendario

|  |  |  |
| --- | --- | --- |
| Fecha | Hito | Criterio de aceptación |
| 11–12 Sep | Entorno + repositorio + estructura Flask | App corre localmente y conecta a BD + deploy staging. |
| 17 Sep | Cierre Sprint 1 - Review/Retro | Login funcionando + esquema BD creado + staging operativo. |
| 24 Sep | Cierre Sprint 2 - Review/Retro | Dashboard lista inventario realista + validaciones funcionando. |
| 1 Oct | Cierre Sprint 3 - Review/Retro | Filtros por estado + alertas estables + umbrales configurables. |
| 10 Oct | Entrega MVP + demo + Revisión final con stakeholders | Guía de uso, dataset demo, demo en vivo y feedback validado. |

# 5. Riesgos y mitigación

|  |  |  |
| --- | --- | --- |
| Riesgo | Impacto | Mitigación |
| Retraso en CRUD | Demora en alertas (Sprint 3) | Priorizar campos esenciales; postergar extras. Validaciones mínimas pero funcionales. |
| Problemas de despliegue | Bloquea demo | Automatizar pasos básicos desde Sprint 1 y usar DO con imagen estándar. |
| Cambios de alcance | Retrabajo y retrasos | Definir alcance fijo y control de cambios estricto. |
| Datos insuficientes para pruebas | Alertas poco realistas | Seed con casos límite: medicamentos vencidos, agotados, próximos a caducar. |
| Configuración de umbrales compleja | Retraso Sprint 3 | Implementar configuración simple (valores por defecto) en Sprint 2, personalización en Sprint 3. |

# 6. Consideraciones de equipo

Estimación: Plan diseñado para 2-3 desarrolladores trabajando tiempo completo o 1 desarrollador con dedicación al proyecto.  
  
Distribución sugerida:  
- Developer 1: Backend (Flask, BD, autenticación, CRUD)  
- Developer 2: Frontend (Bootstrap, dashboard, alertas visuales)  
- Developer 3 (opcional): DevOps, testing, documentación  
  
Ajustes por tamaño de equipo:  
- 1 desarrollador: Considerar reducir scope de Sprint 3 (alertas básicas sin configuración)  
- 3+ desarrolladores: Agregar testing automatizado y CI/CD básico