**Incremento del Producto**

**Sprint #: 1**

**Versión entregable: 1.0.0**

**Funcionalidades incluidas:**

- Sistema CRUD completo para todas las tablas del modelo de datos (Site, SAPEquipment, CECO, VehicleType, VehicleStatus, Vehicle, Role, UserStatus, FlotaUser, Route, ServiceType, MaintenanceSchedule, Ingreso, WorkOrderStatus, WorkOrder, Task, TaskAssignment, Pause, Document, Repuesto, Notification, Report, WorkOrderMechanic, SparePartUsage, Incident, Diagnostics, IncidentImage, IngresoImage)

- Formularios de creación, edición y eliminación para cada entidad

- Filtros y búsquedas avanzadas en listados de datos

- Validaciones básicas en formularios

- Interfaz responsiva básica con Bootstrap

- Navegación entre módulos

**ID: SB-011 (Subida y gestión de documentos - implementado como base CRUD sin subida de archivos multimedia)**

**Pruebas realizadas:**

- Pruebas de usabilidad básica con usuarios de prueba

- Validación de datos en formularios

**Mejoras identificadas:**

- Implementar subida real de documentos e imágenes con firma digital

- Agregar validación de archivos (tipos, tamaños, seguridad)

- Crear control de versiones de documentos

- Implementar firma digital electrónica

- Crear consulta de históricos con límite de 45 días

- Agregar metadatos completos a archivos (usuario, fecha, tipo)

- Integrar validación documental por recepcionista

- Desarrollar notificaciones de cambios en documentos

**Sprint Goal**

**Sprint #1**

**Objetivo:**

Implementar la base funcional completa del sistema de gestión de flota mediante operaciones CRUD para todas las entidades del modelo de datos, incluyendo formularios, filtros, navegación y autenticación básica, sentando las bases para las funcionalidades avanzadas de subida de documentos y firma digital en sprints posteriores.

**Medición de éxito:**

- 100% de las entidades del modelo (28 tablas) con operaciones CRUD funcionales (crear, leer, actualizar, eliminar)

- Formularios con validaciones básicas implementados para todas las entidades críticas

- Sistema de filtros y búsquedas avanzadas operativo en al menos 80% de los listados

- No hay bugs críticos que impidan el uso básico del sistema

- Documentación técnica actualizada con la implementación realizada

- Base de datos poblada con datos de prueba para validación

**Product Goal**

**Objetivo a largo plazo:**

Centralizar y automatizar completamente la gestión del ciclo de vida de mantenimiento de la flota vehicular de PepsiCo, desde el ingreso hasta la salida de talleres, reduciendo errores operativos en al menos 40%, eliminando dependencias de procesos manuales (Excel, WhatsApp) y proporcionando una plataforma web responsiva que soporte la trazabilidad total de todas las operaciones de mantenimiento, con integración SAP y firma digital para máxima eficiencia y cumplimiento normativo.

**Fecha revisión**:

20 de Noviembre de 2025

**Progreso actual:**

- Sprint 1 completado: Base CRUD implementada para todas las tablas del modelo de datos, con formularios, filtros y navegación básica.

- 11 historias de usuario restantes en Product Backlog, organizadas en 4 sprints adicionales.

- Arquitectura del sistema definida con diagramas UML (clases, casos de uso, actividad).

- Requerimientos documentados según estándar IEEE 830.

- Equipo formado y metodologías ágiles implementadas (DoR, DoD, Sprint Backlog, Roadmap).

- Prototipo funcional desplegado en entorno de desarrollo.

- Próximo hito: Implementación de Calendario, Ingreso, Incidentes, Diagnósticos, Órdenes de Trabajo, Tareas, Repuestos, Pausas y Subida de documentos con firma digital.

**Definition of Done (DoD) Criterios de completitud:**

El Definition of Done establece los criterios que deben cumplirse para considerar que una historia de usuario, tarea o sprint está completamente terminada y lista para ser entregada al usuario final.

**Criterios de completitud:**

1. **Código implementado y funcional**
   * El código está escrito siguiendo las mejores prácticas de Django
   * Todas las funcionalidades requeridas están implementadas
   * El código compila sin errores
2. **Documentación actualizada**
   * Documentación técnica actualizada (README, docstrings)
   * Documentación de usuario si aplica
   * Diagramas UML actualizados si hay cambios en modelos
3. **Aceptación del Product Owner**
   * Demostración funcional al Product Owner
   * Aprobación explícita de que cumple con los requerimientos
   * Feedback incorporado
4. **Pruebas de calidad**
   * Pruebas de usabilidad básica realizadas
5. **Compatibilidad**
   * Funciona en navegadores soportados (Chrome, Firefox, Safari, Edge)
   * Interfaz responsiva en desktop, tablet y móvil
   * Compatibilidad con versiones de Django/Python especificadas
6. **Sin defectos críticos**
   * No hay bugs críticos o bloqueantes
   * Issues menores documentados y planificados para corrección
   * Logging y manejo de errores implementado
7. **Trazabilidad**
   * Historia ligada a requerimientos del IEEE 830
   * Criterios de aceptación cumplidos
   * Cambios versionados en Git con commits descriptivos

Estos criterios se aplican a cada historia de usuario completada. En caso de que algún criterio no sea aplicable (ej. integración SAP en historias iniciales), debe documentarse explícitamente por qué se omite.

**Definition of Ready (DoR) Criterios de preparación:**

El Definition of Ready establece los criterios que debe cumplir una historia de usuario o tarea antes de ser considerada "ready" para ser incluida en un sprint y comenzada por el equipo de desarrollo.

Criterios de preparación:

1. **Historia bien definida (INVEST)**
   * Independent: La historia puede desarrollarse sin depender excesivamente de otras
   * Negotiable: No es un contrato detallado, permite discusión
   * Valuable: Proporciona valor claro al usuario o negocio
   * Estimable: El equipo puede estimar el esfuerzo requerido
   * Small: Puede completarse dentro de un sprint
   * Testable: Tiene criterios de aceptación verificables
2. **Criterios de aceptación claros**
   * Definidos y específicos (Given-When-Then)
   * Verificables objetivamente
   * Aprobados por el Product Owner
3. **Estimación realizada**
   * Estimación de esfuerzo en horas o puntos de historia
   * Acordada por todo el equipo de desarrollo
   * Considera complejidad técnica y riesgos
4. **Prioridad definida**
   * Prioridad clara en el Product Backlog
   * Alineada con objetivos del sprint
   * Aprobada por Product Owner
5. **Dependencias identificadas**
   * Dependencias técnicas o de negocio documentadas
   * Plan de mitigación si existen bloqueos
   * No hay dependencias críticas sin resolver
6. **Aprobación del Product Owner**
   * Revisada y aprobada por el PO
   * Alcance y expectativas claras
   * Alineada con visión del producto
7. **Información técnica suficiente**
   * Requerimientos no funcionales claros (rendimiento, seguridad)
   * Restricciones técnicas identificadas
   * Diseño de alto nivel si es complejo
8. **Pruebas preliminares consideradas**
   * Estrategia de testing definida
   * Entornos de prueba identificados
   * Criterios de calidad claros
9. **Tamaño apropiado**
   * Puede completarse en 1-3 días de trabajo
   * Si es más grande, debe descomponerse
   * Equilibrio entre valor y esfuerzo
10. **Impacto en arquitectura evaluado**
    * Cambios en modelos de datos identificados
    * Impacto en otras funcionalidades considerado
    * Compatibilidad con sistemas existentes verificada
11. **Riesgos identificados**
    * Riesgos técnicos documentados
    * Plan de contingencia si aplica
    * Complejidad estimada correctamente
12. **Documentación preliminar**
    * Historia documentada en el backlog
    * Referencias a requerimientos IEEE 830
    * Mockups o wireframes si aplica para UI

Estos criterios se verifican durante el Sprint Planning o Refinement meetings. Si una historia no cumple con el DoR, se devuelve al Product Backlog para refinamiento antes de ser considerada para el próximo sprint.