



# 1. Planning

## ▼ 1. Análisis de viabilidad y alcance

### Viabilidad Técnica

- **Tecnologías propuestas:** Java (aplicación desktop) + Python (API) + Base de datos
- **Arquitectura viable:** Aplicación desktop Java con API REST en Python
- **Compatibilidad:** Framework Scrum adecuado para desarrollo iterativo

### Alcance Definido

#### INCLUIDO:

- Sistema de autenticación y roles
- Gestión de dispositivos y clientes
- Control de inventario de piezas
- Seguimiento de reparaciones
- Sistema de facturación e informes
- Sistema de backup

#### EXCLUIDO:

- Aplicación móvil
- Integración con sistemas externos
- Página web para clientes
- Sistema de pagos online

---

## ▼ 2. Cronograma con Framework Scrum (Agosto - Diciembre)

- Definición completa de requisitos
- Diseño de arquitectura
- Configuración de entornos
- Capacitación en tecnologías

### **Sprint 1: Módulo básico**

- Sistema de autenticación
- CRUD de clientes
- CRUD de dispositivos
- Base de datos inicial

### **Sprint 2: Gestión de reparaciones**

- Flujo de trabajo de reparaciones
- Estados del dispositivo
- Asignación de técnicos
- Sistema de notificaciones básico

### **Sprint 3: Inventario y Piezas**

- Gestión de inventario
- Asociación piezas-reparaciones
- Alertas de stock bajo
- Precios y costos

### **Sprint 4: Facturación y reportes**

- Sistema de facturación
- Generación de reportes
- Estadísticas y dashboard
- Sistema de backup

## **Sprint 5: Integración y Pruebas**

- Pruebas integrales
- Depuración de errores
- Optimización de rendimiento
- Documentación final

### **▼ Cronograma**

#### **SEMANA 1 (26 Oct - 1 Nov): Planificación, Análisis y Diseño**

- Definición de alcance revisado
- Asignación de tareas específicas
- Setup completo de entornos de desarrollo
- Estructura de base de datos final

#### **Días 3-4 (28-29 Oct): Análisis Detallado**

- Especificación técnica de requisitos
- Diagramas de casos de uso
- Modelo de datos entidad-relación
- Definición de APIs Java-Pytho

#### **Días 5-7 (30 Oct - 1 Nov): Diseño Técnico**

- Arquitectura de la aplicación
- Diseño de interfaces de usuario
- Esquema de base de datos
- Plan de integración Java-Python

#### **SEMANA 2 (2-8 Nov): Implementación - Módulos Base**

##### **Backend (Python API)**

- Sistema de autenticación y usuarios

- CRUD de clientes y dispositivos
- Modelos de base de datos
- API endpoints básicos

### **Frontend (Java Desktop)**

- Login y pantallas principales
- Formularios de clientes
- Listado de dispositivos
- Navegación básica

## **SEMANA 3 (9-15 Nov): Implementación - Núcleo del Sistema**

### **Módulos Críticos**

- Gestión completa de reparaciones
- Sistema de inventario de piezas
- Estados y flujos de trabajo
- Asignación de técnicos

### **Integraciones**

- Conexión frontend-backend
- Manejo de imágenes
- Sistema de notificaciones básico

## **SEMANA 4 (16-22 Nov): Implementación - Funcionalidades Avanzadas & Pruebas Iniciales**

### **Funcionalidades Completas**

- Sistema de facturación
- Generación de reportes
- Alertas y notificaciones

- Sistema de backup

### **Pruebas Iniciales (en paralelo)**

- Pruebas unitarias
- Pruebas de integración
- Pruebas de base de datos
- Depuración de errores críticos

## **SEMANA 5 (23-29 Nov): Pruebas Intensivas, Implementación y Entrega**

### **Días 23-26 Nov: Pruebas Exhaustivas**

- Pruebas de sistema completas
- Pruebas de usabilidad
- Pruebas de rendimiento
- Validación de todos los requisitos

### **Días 27-28 Nov: Implementación Final**

- Instalación y configuración
- Migración de datos de prueba
- Preparación de documentación
- Manual de usuario rápido

### **29 Nov: Entrega y Presentación**

- Demo final del sistema
- Entrega de documentación
- Presentación al cliente/profesor
- Plan de mantenimiento básico

## ▼ 3. Recursos y roles del equipo

### Roles Scrum

- **Product Owner:** Representante del cliente
- **Scrum master:** Karla
- **Team Members:** 2 desarrolladores

### Recursos Técnicos

- Java JDK 11+
  - Python 3.8+
  - Base de datos (MySQL/PostgreSQL)
  - Git para control de versiones
  - IDE (IntelliJ/Eclipse + PyCharm/VSCode)
- 

## ▼ 4. Entregables por fase

### Fase de Planificación (Actual)

- Documento de requisitos detallado
  - Plan de proyecto con cronograma
  - Diseño de arquitectura
  - Definición de criterios de aceptación
- 

## ▼ 5. Gestión de riesgos

### Riesgos identificados

1. Complejidad de integración Java-Python
2. Gestión de imágenes y almacenamiento
3. Performance con grandes volúmenes de datos
4. Curva de aprendizaje del equipo

## Mitigación

- Prototipos tempranos de integración
- Definición clara de APIs
- Pruebas de carga tempranas
- Sesiones de capacitación técnica

## ▼ 6. Métricas de éxito

- Cumplimiento de fechas de entrega por sprint
- Cobertura de pruebas > 80%
- Menos de 5 bugs críticos en producción
- Satisfacción del cliente en revisión de sprint

## ▼ Distribución de esfuerzo por fase

Fase	Semanas	% Esfuerzo	Actividades Clave
Planificación	0.5	10%	Alcance, cronograma, recursos
Análisis	0.5	10%	Requisitos técnicos, modelos
Diseño	0.5	10%	Arquitectura, interfaces, BD
Implementación	2.5	50%	Codificación, integración
Pruebas	1.0	15%	Unitarias, integración, sistema
Implementación	0.5	5%	Instalación, configuración

## ▼ Entregables por semana

### Fin Semana 1 (1 Nov)

- Documento de diseño completo
- Prototipos de interfaz
- Esquema de BD finalizado

### **Fin Semana 2 (8 Nov)**

- Sistema de login funcionando
- CRUD de clientes y dispositivos
- API básica operativa

### **Fin Semana 3 (15 Nov)**

- Flujo completo de reparaciones
- Inventario de piezas funcional
- Integración frontend-backend

### **Fin Semana 4 (22 Nov)**

- Todas las funcionalidades implementadas
- Sistema estable y testeado
- Documentación técnica

### **Fin Semana 5 (29 Nov)**

- Sistema completo y depurado
- Manual de usuario
- Presentación final

---

## ▼ **Enfoque de desarrollo por módulos prioritarios**

1. **Módulo CRÍTICO:** Gestión de reparaciones + inventario
2. **Módulo IMPORTANTE:** Facturación + reportes
3. **Módulo COMPLEMENTARIO:** Backup + estadísticas avanzadas