



# PLAN DE MANTENIMIENTO Y SOPORTE

GESCOMPH

Jesús David Fierro Rivera  
Brayan Santiago Guerrero Mendez

|           |                                      |
|-----------|--------------------------------------|
| Contenido |                                      |
| 1.        | 2                                    |
| 2.        | 2                                    |
| 3.        | 4                                    |
| 4.        | 5                                    |
| 5.1       | Mantenimiento Correctivo             |
| 5.2       | Mantenimiento Preventivo             |
| 5.3       | Mantenimiento Adaptativo             |
| 5.4       | Mantenimiento Perfectivo             |
| 5.        | 85                                   |
| 6.        | <b>¡Error! Marcador no definido.</b> |
| 7.        | <b>¡Error! Marcador no definido.</b> |
| 8.        | <b>¡Error! Marcador no definido.</b> |
| 9.        | <b>¡Error! Marcador no definido.</b> |
|           | 8                                    |

## 1. Introducción

El presente plan define las estrategias y procedimientos para asegurar la continuidad operativa, la calidad del servicio y la evolución técnica del sistema **GESCOMPH**. Se establecen políticas de mantenimiento preventivo, correctivo, adaptativo y perfectivo, así como los mecanismos de soporte a usuarios internos y externos.

## 2. Objetivos

### Objetivo General

Garantizar la operatividad continua, la disponibilidad y la evolución sostenible del sistema **GESCOMPAH** mediante la implementación de políticas de mantenimiento (correctivo, preventivo, adaptativo y perfectivo) y un plan de soporte estructurado, que asegure la atención eficiente a los usuarios, la prevención de fallos, la adaptación a cambios normativos o tecnológicos y la mejora progresiva del software en el entorno de la Alcaldía de Palermo – Huila.

Este objetivo general busca que el sistema no solo se mantenga funcionando, sino que permanezca alineado con las necesidades institucionales, técnicas y legales, minimizando riesgos de interrupción y maximizando el valor entregado a la comunidad.

### Objetivos Específicos

#### 1. Establecer un marco de mantenimiento correctivo eficiente

- Asegurar la rápida identificación, diagnóstico y solución de fallos en producción.
- Reducir el impacto de los errores críticos que afecten la gestión de arriendos, contratos y pagos.
- Documentar cada incidencia para evitar recurrencias y mantener la trazabilidad.

#### 2. Implementar prácticas de mantenimiento preventivo periódico

- Definir rutinas técnicas (limpieza de logs, optimización de consultas SQL, verificación de copias de seguridad).
- Evitar degradación del sistema y prevenir fallos antes de que afecten a los usuarios.
- Mantener las versiones de Angular, .NET y SQL Server actualizadas para reducir vulnerabilidades de seguridad.

### 3. Definir estrategias de mantenimiento adaptativo

- Adaptar el sistema a nuevas condiciones externas como:
  - Cambios en normativa local de arriendos.
  - Nuevas versiones de navegadores y dispositivos móviles.
  - Migraciones a versiones más recientes de librerías y frameworks.
- Garantizar que el sistema evolucione de acuerdo con el entorno tecnológico y legal de la Alcaldía.

### 4. Promover la mejora continua mediante mantenimiento perfectivo

- Recoger y analizar sugerencias de los usuarios (administradores, arrendatarios, ciudadanos).
- Optimizar procesos de gestión de contratos, reportes y notificaciones.
- Mejorar la experiencia de usuario (UI/UX) con interfaces más intuitivas, rápidas y accesibles.
- Incorporar métricas y estadísticas avanzadas para apoyar la toma de decisiones.

### 5. Diseñar e implementar un plan de soporte a usuarios estructurado

- Establecer niveles de soporte (N1, N2, N3) con roles y responsabilidades claras.
- Definir canales de comunicación accesibles (formulario interno, correo institucional, línea telefónica).
- Estandarizar tiempos de respuesta y resolución de incidencias según su criticidad.
- Garantizar la capacitación básica a funcionarios administrativos para un uso correcto del sistema.

## 6. Asegurar la calidad del servicio mediante indicadores medibles

- Definir métricas clave: disponibilidad  $\geq 95\%$ , tiempo de respuesta  $< 5$  segundos, resolución de incidencias críticas en  $\leq 24$  horas.
- Realizar seguimiento continuo con reportes de mantenimiento y soporte.
- Aplicar encuestas de satisfacción para evaluar la percepción de los usuarios.

## 3. Alcance del Sistema

GESCOMPH estará orientado a la administración de los establecimientos comerciales en plazas públicas de Palermo, Huila, y contará con módulos funcionales para tres perfiles de usuarios: administrador, arrendatario y ciudadano no autenticado. El administrador tendrá acceso a operaciones CRUD sobre plazas y establecimientos, gestión de contratos y consulta del estado de ocupación. También podrá administrar citas y confirmar solicitudes agendadas por ciudadanos. Los arrendatarios, tras autenticarse, podrán consultar sus contratos vigentes, aunque el sistema no incluirá funcionalidades de pago en línea ni integración con pasarelas de recaudo electrónico. Los ciudadanos no autenticados podrán explorar locales disponibles, visualizar información básica y registrar solicitudes de cita para conocerlos, sujetas a validación por el administrador.

El sistema excluye funcionalidades de gestión presupuestal, control financiero interno, integración con sistemas contables, validación automática de pagos, envío de SMS, módulos de firma digital y validación jurídica de contratos.

## 4. Estrategia de Mantenimiento

El mantenimiento del sistema **GESCOMPÁH** se concibe como un proceso integral que busca garantizar la continuidad del servicio, prevenir fallos, adaptar el software a nuevas necesidades y mejorarlo progresivamente. Para ello, se aplicarán los siguientes tipos de mantenimiento:

### 4.1. Mantenimiento Correctivo

Este tipo de mantenimiento se aplica para solucionar fallos y errores detectados en el sistema que afecten su funcionamiento.

Ejemplos:

- Error en el inicio de sesión (fallo de autenticación JWT).
- Reporte incorrecto en el cálculo de la mora de pagos.
- Caída de la base de datos por exceso de conexiones concurrentes.

Procedimiento paso a paso:

- El usuario reporta la incidencia a través del formulario de soporte o correo institucional.
- El equipo de soporte valida el error, lo reproduce y clasifica según la severidad:
  - Crítica: caída total del sistema.
  - Alta: afecta a los administradores en procesos esenciales.
  - Media: afecta a arrendatarios o ciudadanos en funciones específicas.
  - Baja: errores menores en interfaz gráfica o mensajes de validación.
- El equipo de desarrollo recibe el caso, genera un ticket en Trello/Jira y aplica la corrección en un entorno de pruebas.

- Una vez validado, se despliega en producción siguiendo el flujo de integración continua (GitHub Actions o pipeline CI/CD).
- Se notifica al usuario final y se documenta el incidente en el historial de mantenimiento.

#### 4.2. Mantenimiento Preventivo

Busca evitar fallos futuros mediante la revisión periódica de la infraestructura y el software.

Actividades principales:

- Limpieza y depuración de logs en backend y base de datos.
- Optimización de consultas SQL para mantener el rendimiento en < 5 segundos..
- Verificación de la integridad de las copias de seguridad diarias.
- Revisión de la seguridad del sistema (roles, permisos, intentos fallidos de acceso).
- Actualización de librerías y dependencias de Angular y .NET, para prevenir vulnerabilidades.

Beneficios esperados:

- Reducción de paradas inesperadas del sistema.
- Menor acumulación de deuda técnica.
- Mayor estabilidad en el tiempo.

#### 4.3. Mantenimiento Adaptativo

Se refiere a los cambios realizados para ajustar el sistema a nuevas condiciones externas.

Ejemplos:

- Cambios normativos en la Ley de arriendos o en las tasas de mora.

- Migración a nuevas versiones de frameworks (ejemplo: Angular 20 → Angular 21).
- Ajustes por compatibilidad con nuevos navegadores o dispositivos móviles.

Procedimiento:

- Se analiza el cambio externo que impacta el sistema.
- El equipo de desarrollo diseña las modificaciones necesarias.
- Se realizan pruebas de regresión para garantizar que los módulos actuales no se vean afectados.
- Se actualiza la documentación técnica y funcional del sistema.

#### 4.4. Mantenimiento Perfectivo

Tiene como objetivo mejorar la calidad del sistema sin que exista un error previo.

Ejemplos:

- Mejorar la interfaz de usuario para arrendatarios con gráficos estadísticos.
- Ampliar reportes de ocupación con indicadores de tendencia.
- Optimizar el consumo de memoria y CPU del backend.

Metodología:

- Se recogen sugerencias de los usuarios y se documentan en el backlog de mejoras.
- El Product Owner (representante de la Alcaldía) prioriza qué mejoras son más relevantes.
- Se planifican en ciclos ágiles (sprints de 2–3 semanas).

- Se implementan y validan en entornos de pruebas antes del despliegue en producción.

## 5. Plan de Soporte

El soporte del sistema se organiza en niveles jerárquicos de atención que aseguran la resolución eficiente de incidentes y la asistencia continua a los usuarios.

### 5.1. Niveles de soporte

- Nivel 1 – Soporte al usuario final

Responsable: mesa de ayuda de la Alcaldía.

Funciones:

- Resolver dudas básicas (ejemplo: cómo recuperar una contraseña).
- Guiar a los usuarios en procesos simples (consultar contrato, solicitar cita).
- Escalar a Nivel 2 si no es posible resolver.

- Nivel 2 – Equipo de soporte técnico

Responsable: equipo de TI de la Alcaldía o equipo asignado al proyecto.

Funciones:

- Identificación y solución de problemas funcionales.
- Revisión de logs del sistema y base de datos.
- Soporte a administradores en la gestión de usuarios, contratos y reportes.
- Escalamiento a Nivel 3 si el error requiere modificación en el código.

- Nivel 3 – Equipo de desarrollo

Responsable: desarrolladores de GESCOMPÁH.

#### Funciones:

- Depuración y corrección de bugs en código fuente.
- Migración de base de datos.
- Desarrollo de mejoras evolutivas.
- Actualizaciones de frameworks, librerías y seguridad.

#### 5.2. Canales de soporte

- Formulario interno de soporte en el sistema.
- Correo electrónico institucional.
- Línea telefónica de la Alcaldía (horario laboral).

#### 5.3. Flujo de Atención

- Usuario reporta la incidencia.
- Se asigna prioridad:
  - Crítica: afecta disponibilidad del sistema completo.
  - Alta: afecta a un rol clave (ej. administrador).
  - Media: afecta funciones individuales.
  - Baja: mejoras cosméticas o sugerencias.
- Escalamiento al nivel de soporte correspondiente.
- Retroalimentación al usuario sobre estado de la solicitud.

### 6. Herramientas de Gestión

- Trello / Jira: gestión de incidencias y backlog de mantenimiento
- GitHub: control de versiones y trazabilidad de cambios.
- SQL Server Tools: monitoreo de base de datos.
- Logs en .NET: seguimiento de errores y auditoría.

### 7. Indicadores de Calidad

Para garantizar que el mantenimiento y soporte sean efectivos, se definen KPIs (Indicadores Clave de Desempeño) medibles:

### 7.1. Disponibilidad del sistema

- Meta:  $\geq 95\%$  de disponibilidad mensual.
- Medición: monitoreo de uptime en el servidor y registros de caídas.

### 7.2. Tiempo medio de resolución (MTTR)

- Incidencias críticas:  $\leq 24$  horas.
- Incidencias altas:  $\leq 48$  horas.
- Incidencias medias:  $\leq 5$  días hábiles.
- Incidencias bajas:  $\leq 10$  días hábiles.

### 7.3. Rendimiento del sistema

- Meta: tiempo de respuesta  $< 5$  segundos en operaciones principales (consultar contratos, listar locales, generar reportes)  
01-Informe-de-especificacion-de...
- Medición: pruebas de carga semestrales.

### 7.4. Tasa de recurrencia de errores

- Meta:  $\leq 5\%$  de incidencias repetidas por mes.
- Medición: análisis de tickets cerrados vs. reabiertos.

### 7.5. Satisfacción del usuario

- Meta:  $\geq 80\%$  en encuestas de satisfacción post-atención.
- Medición: formulario automático tras cierre de incidencia.

## 8. Conclusiones

El presente plan asegura la continuidad del sistema GESCOMPAH, garantizando un soporte estructurado a los usuarios y una evolución controlada del software frente a cambios técnicos y normativos. Su aplicación permitirá mantener la confiabilidad, disponibilidad y escalabilidad del sistema en el tiempo.