

Sistema de Gestión de Servicios —

Centro de Atención Tecnológica (CAT)

Repository: Parcial-analisis

Autor: Tito Alejandro Burbano Plazas - 20231020169

Fecha: 1 de diciembre de 2025

Índice

1. Resumen
2. Historias de usuario
3. Requerimientos del sistema
 - a. Requerimientos funcionales
 - b. Requerimientos no funcionales
4. Backlog inicial e Issues sugeridos (para GitHub)
5. Planificación (Poker Planning)
6. Diagrama de proceso (BPMN)
 - a. Descripción
 - b. Imagen PNG
7. Diagrama de Gantt

1. Resumen ejecutivo

El Centro de Atención Tecnológica (CAT) requiere un sistema para modernizar el registro, seguimiento y resolución de solicitudes de soporte técnico. El objetivo es mejorar la trazabilidad, reducir tiempos de respuesta y generar métricas que permitan optimizar recursos.

El siguiente documento incluye historias de usuario, requerimientos, backlog inicial (issues para GitHub), un diagrama BPMN del proceso principal, planificación mediante poker planning y un diagrama de Gantt alineado con Scrum y la gestión en GitHub.

2. Historias de usuario

1. **Como estudiante**, quiero registrar una solicitud de soporte, para recibir ayuda técnica de forma organizada.
2. **Como docente**, quiero adjuntar evidencias (capturas o documentos) a mi solicitud, para que el técnico entienda mejor el problema.
3. **Como administrativo**, quiero seleccionar la prioridad de mi solicitud, para que se atienda según la urgencia.
4. **Como técnico**, quiero visualizar todas las solicitudes asignadas, para gestionarlas de manera eficiente.
5. **Como técnico**, quiero actualizar el estado de una solicitud (pendiente, en proceso, resuelto, cerrado), para mantener la trazabilidad.
6. **Como coordinador del CAT**, quiero asignar solicitudes a técnicos, para distribuir mejor la carga de trabajo.
7. **Como coordinador del CAT**, quiero generar reportes semanales, para evaluar productividad y tiempos de respuesta.
8. **Como usuario**, quiero recibir notificaciones cuando el estado de mi solicitud cambie, para estar informado del avance.
9. **Como usuario**, quiero consultar el histórico de mis solicitudes, para revisar eventos pasados y soluciones aplicadas.
10. **Como coordinador**, quiero visualizar métricas de tiempo promedio de resolución por categoría, para mejorar la calidad del servicio.
11. **Como técnico**, quiero comentar internamente en una solicitud (nota privada), para comunicarme con el equipo sin notificar al usuario.
12. **Como coordinador**, quiero configurar tiempos de SLA por prioridad, para automatizar escalados.

3. Requerimientos del sistema

Requerimientos funcionales

- **RF1 — Registro de solicitudes:** El sistema debe permitir que usuarios (estudiantes, docentes, administrativos) registren solicitudes con título, descripción, categoría, prioridad, archivos adjuntos y contacto.
- **RF2 — Asignación:** El sistema debe permitir que el coordinador (o un proceso automático) asigne solicitudes a técnicos y re-asigne cuando sea necesario.

- **RF3 — Gestión de estados:** El sistema debe permitir cambiar el estado de una solicitud (Pendiente, En Proceso, Resuelta, Cerrada) y guardar el historial de estados con marcas de tiempo y responsable.
- **RF4 — Notificaciones:** El sistema debe notificar a los usuarios cuando el estado cambie o cuando el técnico solicite información adicional.
- **RF5 — Reportes:** El sistema debe generar reportes (por rango de fechas) con métricas: número de solicitudes, tiempo promedio de resolución, solicitudes por técnico y por categoría.

Requerimientos no funcionales

- **RNF1 — Rendimiento:** Las operaciones críticas (registrar solicitud, listar solicitudes asignadas) deben completarse en menos de 3 segundos bajo carga normal.
- **RNF2 — Disponibilidad:** El servicio debe alcanzar una disponibilidad mínima del 99% mensual (excepto ventanas de mantenimiento programado).
- **RNF3 — Seguridad:** Autenticación integrada con credenciales institucionales (por ejemplo, SSO mediante LDAP o SAML). Comunicación cifrada (HTTPS/TLS).
- **RNF4 — Usabilidad y accesibilidad:** Interfaz web responsive, usable desde móviles y cumpliendo criterios básicos de accesibilidad (contraste, navegación por teclado, etiquetas ARIA).
- **RNF5 — Escalabilidad:** Arquitectura que permita escalar horizontalmente (microservicios o servicios desacoplados) para soportar al menos 500 usuarios concurrentes.

4. Backlog inicial e Issues.

Backlog:

1. **Diseñar modelo de datos (solicitudes, usuarios, técnicos, historial de estado).** — *backend, alta prioridad.*
2. **Implementar autenticación institucional (SSO/LDAP).** — *backend, alta prioridad.*
3. **API: Crear endpoint para registrar solicitudes.** — *backend, alta prioridad.*
4. **Frontend: Formulario de creación de solicitudes (con adjuntos).** — *frontend, alta prioridad.*
5. **Funcionalidad: Asignación manual de solicitudes (coordinador).** — *backend/frontend, media prioridad.*

6. **Dashboard técnico: listar solicitudes asignadas y filtros.** — *frontend, media prioridad.*
7. **Sistema de notificaciones (correo o in-app).** — *backend, alta prioridad.*
8. **Reportes: tiempo promedio de resolución y solicitudes por técnico.** — *backend/frontend, media prioridad.*
9. **Página de historial del usuario (mis solicitudes).** — *frontend, media prioridad.*
10. **Configuración de SLAs y reglas de escalado.** — *backend, baja prioridad.*
11. **Pruebas automatizadas: unitarias e integración (CI).** — *backend, documentación.*
12. **Documentación del API (OpenAPI/Swagger).** — *documentación, media prioridad.*

5. Planificación (Poker Planning)

Usando la escala de Fibonacci (1, 2, 3, 5, 8) como base.

| Historia / Issue | Estimación (puntos) | Comentario |
|---------------------------------|------------------------|---|
| Diseñar modelo de datos | 5 | Incluye diagramas y scripts de BD |
| Autenticación institucional | 8 | Integración SSO/Ldap puede ser compleja |
| Endpoint registro solicitudes | 5 | CRUD mínimo + validaciones |
| Formulario solicitud + adjuntos | 5 | UI + manejo de archivos |
| Asignación manual | 3 | Interfaz simple para coordinador |
| Dashboard técnico | 5 | Filtros y paginación |
| Notificaciones | 8 | Configurar colas o servicio de mail |
| Reportes y métricas | 8 | Consultas agregadas y gráficos |
| Historial usuario | 3 | Consulta por usuario |
| Configurar SLAs | 5 | Reglas y parámetros |

Justificación: Se priorizan elementos que permiten comenzar a operar registro, autenticación, asignación, mientras que los reportes quedan para sprints posteriores.

6. Diagrama de proceso (BPMN)

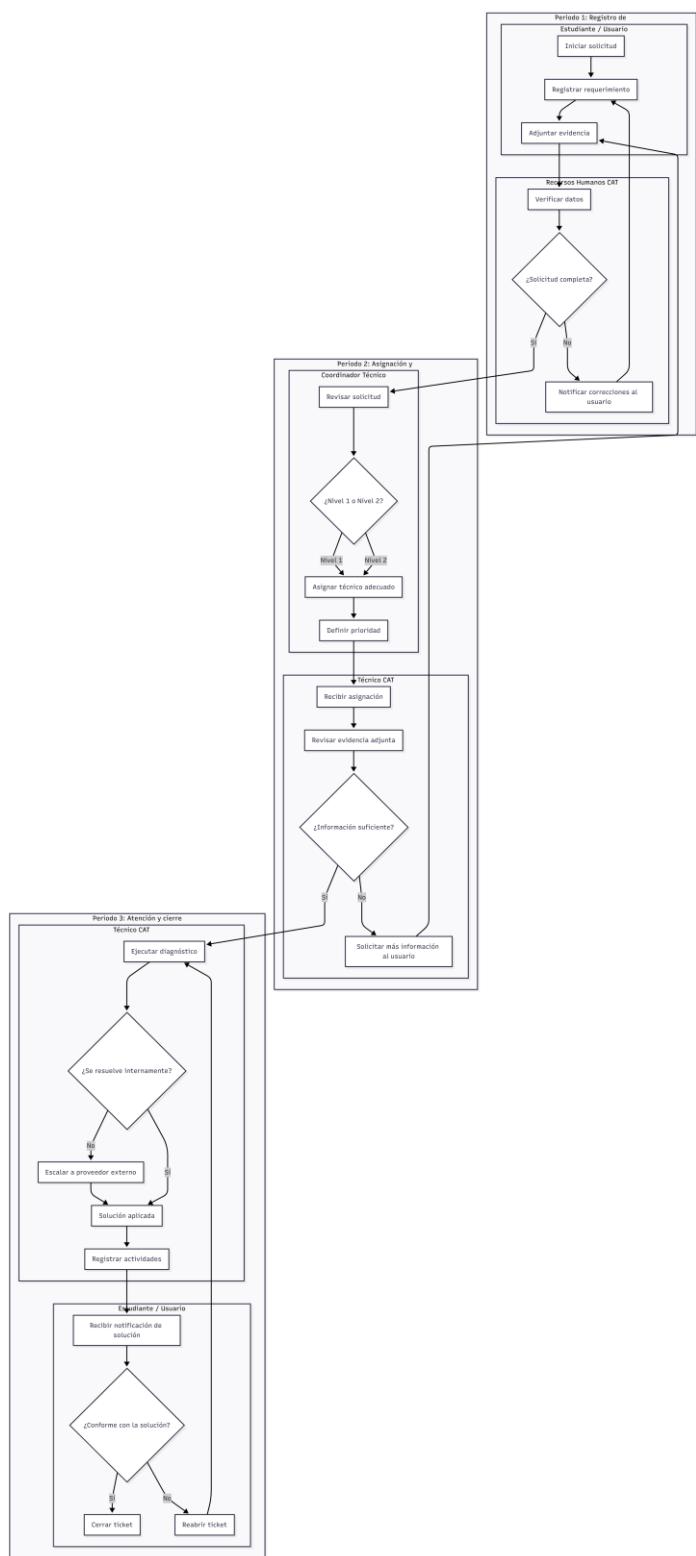
Descripción del proceso: "Gestión de Solicitud de Soporte"

Participantes (pools): Usuario, Sistema CAT, Coordinador, Técnico.

Flujo principal:

1. Usuario crea solicitud (Formulario).
2. Sistema valida y guarda la solicitud.
3. Sistema notifica al Coordinador.
4. Coordinador asigna solicitud a un Técnico (o sistema asigna automáticamente).
5. Técnico revisa y actualiza estado (En Proceso, Resuelta).
6. Si requiere más información, el técnico solicita información al usuario (notificación y espera de respuesta).
7. Una vez resuelta, el técnico marca la solicitud como resuelta.
8. Usuario confirma resolución o solicita reapertura.
9. Coordinador cierra la solicitud y se genera registro en reportes.

Imagen PNG del diagrama (NOTA: El diagrama se puede visualizar mejor en el repositorio)



7. Diagrama de Gantt (12 semanas)

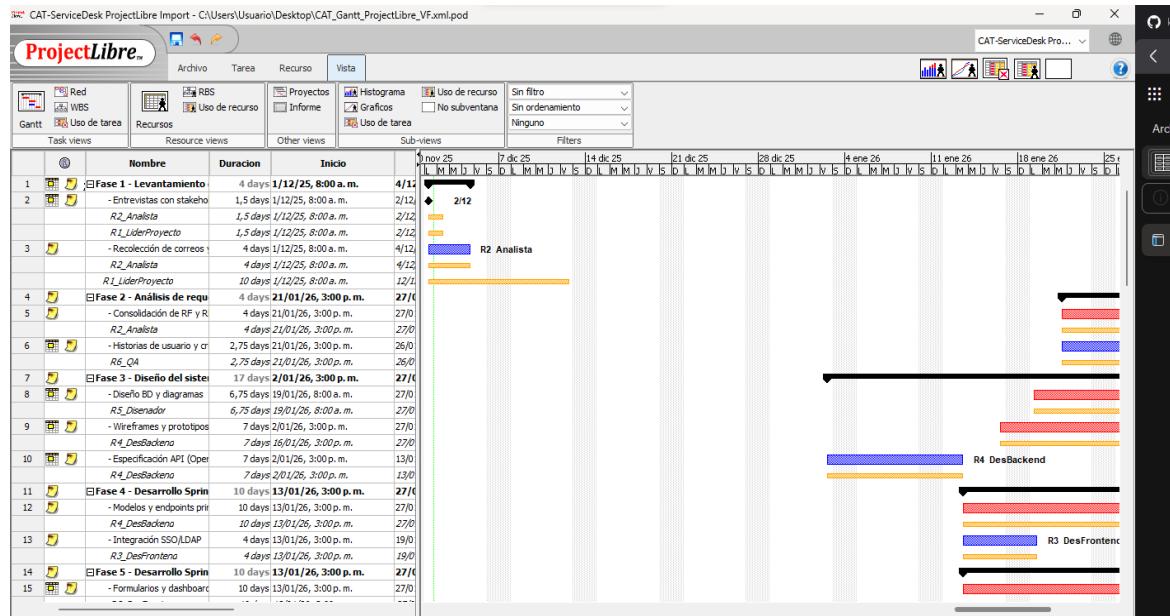
Resumen de sprints y actividades (SCRUM):

- **Semana 1 — Levantamiento de información:** entrevistas con stakeholders, recopilación de correos y ejemplos actuales, definición de alcance mínimo viable.
- **Semana 2 — Análisis de requerimientos:** consolidación de RF y RNF, definición de historias de usuario finales y criterios de aceptación.
- **Semanas 3–4 — Diseño del sistema:** diseño de BD, especificación API.
- **Semanas 5–6 — Desarrollo Sprint 1 (Backend básico):** modelos, endpoints principales.
- **Semanas 7–8 — Desarrollo Sprint 2 (Frontend + integración):** formularios, dashboards básico, subida de archivos, integración con API.
- **Semanas 9–10 — Desarrollo Sprint 3:** notificaciones, reportes y configuración.
- **Semana 11 — Pruebas:** pruebas unitarias, integración, UAT con usuarios reales.
- **Semana 12 — Documentación y entrega:** preparación de manuales, despliegue, entrega y cierre.

Tabla Gantt simplificada (por semana)

| Semana | Actividad |
|--------|--|
| 1 | Levantamiento de información |
| 2 | Análisis de requerimientos |
| 3 | Diseño del sistema (DB, API, UI) |
| 4 | Diseño del sistema (wireframes, prototipos) |
| 5 | Desarrollo Sprint 1 — Backend (endpoints básicos) |
| 6 | Desarrollo Sprint 1 — Backend (SSO, pruebas unitarias) |
| 7 | Desarrollo Sprint 2 — Frontend (formularios, dashboards) |
| 8 | Desarrollo Sprint 2 — Integración Frontend-Backend |
| 9 | Desarrollo Sprint 3 — Notificaciones y reportes |
| 10 | Desarrollo Sprint 3 — SLAs y ajustes finales |
| 11 | Pruebas (unitarias, integración, UAT) |
| 12 | Documentación, despliegue, entrega y cierre |

TIJERAS DE REFERENCIA DEL DIAGRAMA DE GANTT (NOTA: EL DIAGRAMA SE PUEDE VER EXPLICADO EN SU TOTALIDAD EN EL VIDEO ADJUNTADO, O BIEN SIENDO DESCARGADO DESDE EL REPOSITORIO)



ProjectLibre - CAT-ServiceDesk ProjectLibre Import - C:\Users\Usuario\Desktop\CAT_Gantt_ProjectLibre_VF.xml.pod

| | Nombre | Trabajo | Duración | Inicio | Final |
|----|-------------------------------------|---------|-----------|-------------------------------|----------|
| 1 | Fase 1 - Levantamiento | Trabajo | 136 horas | 4 días 1/12/25, 8:00 a.m. | 4/12/25 |
| 2 | - Entrevistas con stakeholders | Trabajo | 24 horas | 1,5 días 1/12/25, 8:00 a.m. | 2/12/25 |
| | R2_Analista | Trabajo | 12 horas | 1,5 días 1/12/25, 8:00 a.m. | 2/12/25 |
| | R1_LiderProyecto | Trabajo | 12 horas | 1,5 días 1/12/25, 8:00 a.m. | 2/12/25 |
| 3 | - Recolección de correos | Trabajo | 32 horas | 4 días 1/12/25, 8:00 a.m. | 4/12/25 |
| | R2_Analista | Trabajo | 32 horas | 4 días 1/12/25, 8:00 a.m. | 4/12/25 |
| | R1_LiderProyecto | Trabajo | 80 horas | 10 días 1/12/25, 8:00 a.m. | 10/12/25 |
| 4 | Fase 2 - Análisis de requerimientos | Trabajo | 54 horas | 4 días 21/01/26, 3:00 p.m. | 27/01/26 |
| 5 | - Consolidación de RF y R | Trabajo | 32 horas | 4 días 21/01/26, 3:00 p.m. | 27/01/26 |
| | R2_Analista | Trabajo | 32 horas | 4 días 21/01/26, 3:00 p.m. | 27/01/26 |
| 6 | - Historias de usuario y cr | Trabajo | 22 horas | 2,75 días 21/01/26, 3:00 p.m. | 26/01/26 |
| | R6_QA | Trabajo | 22 horas | 2,75 días 21/01/26, 3:00 p.m. | 26/01/26 |
| 7 | Fase 3 - Diseño del sistema | Trabajo | 166 horas | 17 días 2/01/26, 3:00 p.m. | 27/01/26 |
| 8 | - Diseño BD y diagramas | Trabajo | 54 horas | 6,75 días 19/01/26, 8:00 a.m. | 27/01/26 |
| | R5_Desarrollador | Trabajo | 54 horas | 6,75 días 19/01/26, 8:00 a.m. | 27/01/26 |
| 9 | - Wireframes y prototipos | Trabajo | 56 horas | 7 días 2/01/26, 3:00 p.m. | 27/01/26 |
| | R4_DesBackend | Trabajo | 56 horas | 7 días 16/01/26, 3:00 p.m. | 27/01/26 |
| 10 | - Especificación API (Oper | Trabajo | 56 horas | 7 días 2/01/26, 3:00 p.m. | 13/01/26 |
| | R4_DesBackend | Trabajo | 56 horas | 7 días 2/01/26, 3:00 p.m. | 13/01/26 |
| 11 | Fase 4 - Desarrollo Sprint | Trabajo | 112 horas | 10 días 13/01/26, 3:00 p.m. | 27/01/26 |
| 12 | - Modelos y endpoints prí | Trabajo | 80 horas | 10 días 13/01/26, 3:00 p.m. | 27/01/26 |
| | R4_DesBackend | Trabajo | 80 horas | 10 días 13/01/26, 3:00 p.m. | 27/01/26 |
| 13 | - Integración SSO/LDAP | Trabajo | 32 horas | 4 días 13/01/26, 3:00 p.m. | 19/01/26 |
| | R3_DesFrontend | Trabajo | 32 horas | 4 días 13/01/26, 3:00 p.m. | 19/01/26 |
| 14 | Fase 5 - Desarrollo Sprint | Trabajo | 112 horas | 10 días 13/01/26, 3:00 p.m. | 27/01/26 |
| 15 | - Formularios y dashboard | Trabajo | 80 horas | 10 días 13/01/26, 3:00 p.m. | 27/01/26 |