

Universidad Tecnológica de El Salvador



Vicerrectoría Académica Facultad de Informática y Ciencias Aplicadas Escuela de Informática

Asignatura: estándares de programación

Docente: Ing. Ruben Escobar Ortegon

Actividad: Tema: Helpdesk + base de conocimiento +
SLA

Alumnos grupo #6:

Barraza Barahona, Eduardo José

Córnejo Valdez, Saul Adalberto

Gámez Oporto, Saul Vicente

Rosales Martínez, Carlos Javier

Silvestre Ramírez, Brandon Antonio 29-1771-2022

San Salvador 10 de febrero de 2026

INDICE

Introducción	2
Objetivos	3
Objetivo general.....	3
Objetivos específicos.....	3
Alcance del proyecto.....	4
❖ Lo que sí incluye	4
❖ Lo que NO incluye (por ahora)	4
Contexto del problema (situación real)	5
Usuarios / Roles del sistema	5
Requerimientos funcionales (RF)	5
❖ Tickets	5
❖ Categorías y prioridades	6
❖ SLA y escalamiento.....	6
❖ Base de conocimiento.....	6
❖ Seguridad / acceso	7
Requerimientos no funcionales (RNF)	7
Reglas de negocio (RB)	7
Casos de uso (CU) (en forma simple).....	8
❖ CU-01 Crear ticket	8
❖ CU-02 Asignar ticket.....	8
❖ CU-03 Responder y cambiar estado.....	8
❖ CU-04 Configurar SLA.....	8
❖ CU-05 Consultar métricas SLA	9
❖ CU-06 Gestionar artículos	9
Criterios de aceptación	9
Repositorio y estructura.....	10
Conclusión.....	10

Introducción

En esta primera entrega nos enfocamos en la parte de requerimientos del proyecto. Nuestro tema es un sistema tipo Helpdesk que sirve para que los usuarios creen tickets cuando tienen un problema o solicitud, y que el equipo de soporte los atienda. Además, el sistema incluye una base de conocimiento con artículos de ayuda (y que estén versionados), y también mide los tiempos usando SLA para saber si se están cumpliendo los tiempos de respuesta y resolución.

En esta unidad no era obligatorio tener la aplicación desarrollada todavía, pero sí tenemos que tener el repositorio creado y organizado, y dejar lista la documentación.

Objetivos

Objetivo general

Desarrollar una plataforma web tipo Helpdesk que permita gestionar de forma ordenada las solicitudes de soporte mediante tickets, incluyendo control de categorías, prioridades, asignaciones, escalamiento, medición de SLA (tiempos de respuesta y resolución) y una base de conocimiento con artículos versionados.

Objetivos específicos

- Permitir que los usuarios creen tickets cuando necesiten ayuda, registrando información como título, descripción, categoría y prioridad.
- Facilitar que los agentes de soporte puedan dar seguimiento a los tickets, responder, resolver y cerrar cada caso, dejando historial del proceso.
- Implementar la administración de categorías, prioridades y asignaciones de tickets para organizar y distribuir mejor el trabajo.
- Incorporar un mecanismo de escalamiento para tickets que lleven demasiado tiempo sin avance o que estén cerca de incumplir tiempos.
- Medir y controlar el cumplimiento de SLA, calculando el tiempo de primera respuesta y el tiempo de resolución.
- Desarrollar una base de conocimiento que permita consultar artículos de ayuda y mantenerlos con versiones para conservar historial de cambios.

Alcance del proyecto

❖ Lo que sí incluye

En este proyecto esperamos implementar:

- Gestión completa de tickets (crear, ver, actualizar, cerrar).
- Estados de tickets (por ejemplo: Abierto, En progreso, En espera, Resuelto, Cerrado).
- Asignación de tickets a agentes.
- Manejo de categorías y prioridades.
- Reglas de SLA por prioridad (y/o por categoría).
- Métricas básicas del SLA (cumplimiento, tiempos promedio, etc.).
- Base de conocimiento con artículos.
- Versionado de artículos (para guardar cambios y poder ver historial).
- Roles de usuario (usuario normal, agente, administrador).

❖ Lo que NO incluye (por ahora)

Para no complicar demasiado la primera versión, por ahora no tenemos pensado incluir:

- Integración con WhatsApp o sistemas externos.
- Chat en tiempo real.
- Inteligencia artificial para sugerir respuestas automáticas (opcional a futuro).
- App móvil nativa (solo web).

Contexto del problema (situación real)

En muchos lugares, cuando alguien necesita soporte, a veces se resuelve por mensajes sueltos (WhatsApp, llamadas, etc.) y se pierde el orden. Entonces la idea del Helpdesk es que todo quede registrado, se le dé prioridad según el caso, se asigne a un agente y se pueda medir si el soporte está cumpliendo los tiempos (SLA). Además, una base de conocimiento ayuda a que las personas encuentren solución sin tener que levantar ticket para todo.

Usuarios / Roles del sistema

En nuestro sistema manejar estos roles:

- **Usuario/Cliente:** crea tickets, revisa el estado, responde si le piden más información, consulta artículos.
- **Agente de soporte:** atiende tickets, responde, cambia estados, resuelve y cierra.
- **Administrador/Supervisor:** gestiona usuarios, configura categorías, prioridades, reglas SLA, asigna tickets, revisa métricas.

Requerimientos funcionales (RF)

Aquí definimos las funciones principales que el sistema debe cumplir.

❖ **Tickets**

- **RF-01:** El sistema debe permitir crear un ticket con título, descripción, categoría y prioridad.
- **RF-02:** El usuario debe poder ver la lista de sus tickets y su estado actual.
- **RF-03:** El agente debe poder cambiar el estado del ticket (Abierto, En progreso, En espera, Resuelto, Cerrado).
- **RF-04:** El sistema debe permitir asignar un ticket a un agente.
- **RF-05:** El sistema debe guardar un historial de cambios (estado, asignación, fecha, quién lo hizo).

- **RF-06:** El agente debe poder responder dentro del ticket (comentarios o respuestas internas/externas).
- **RF-07:** El sistema debe permitir cerrar un ticket cuando ya esté resuelto.

❖ Categorías y prioridades

- **RF-08:** El administrador debe poder crear/editar/eliminar categorías y subcategorías.
- **RF-09:** El sistema debe permitir definir prioridades (Baja, Media, Alta, Crítica).

❖ SLA y escalamiento

- **RF-10:** El sistema debe permitir definir SLA por prioridad (tiempo máximo para primera respuesta y tiempo máximo para resolución).
- **RF-11:** El sistema debe calcular si un ticket cumple o incumple SLA.
- **RF-12:** El sistema debe permitir escalar tickets manualmente o automáticamente cuando estén cerca de incumplir SLA.

❖ Base de conocimiento

- **RF-13:** El sistema debe permitir ver artículos de ayuda.
- **RF-14:** El sistema debe permitir buscar artículos por palabra clave y categoría.
- **RF-15:** El administrador/agente debe poder crear y editar artículos.
- **RF-16:** El sistema debe manejar versiones de artículos (guardar versiones anteriores con fecha y autor).
- **RF-17:** El sistema debe permitir relacionar un artículo con un ticket resuelto (opcional, pero útil).

❖ **Seguridad / acceso**

- **RF-18:** El sistema debe permitir inicio de sesión y control de permisos por roles.

Requerimientos no funcionales (RNF)

Estos son requisitos de calidad del sistema:

- **RNF-01 Seguridad:** Las contraseñas deben guardarse cifradas y no en texto plano.
- **RNF-02 Usabilidad:** La interfaz debe ser fácil de entender, tipo sistema web normal, y responsive.
- **RNF-03 Rendimiento:** Las páginas principales deben cargar en un tiempo razonable.
- **RNF-04 Disponibilidad:** El sistema debe estar disponible para consultarse y crear tickets sin caídas constantes.
- **RNF-05 Auditabilidad:** Debe existir historial de acciones importantes (cambios de estado, asignaciones, cierres).

Reglas de negocio (RB)

- **RB-01:** Un ticket en estado “Cerrado” ya no se puede editar por el usuario (solo admin si hace falta).
- **RB-02:** Los tickets “Críticos” tienen SLA más corto que los de prioridad baja o media.
- **RB-03:** Si un ticket pasa demasiado tiempo sin respuesta, debe marcarse como posible incumplimiento y escalarse.
- **RB-04:** Un artículo publicado debe conservar el historial de versiones (no se borra la versión anterior).
- **RB-05:** Todo ticket debe tener categoría y prioridad obligatoria.

Casos de uso (CU) (en forma simple)

❖ CU-01 Crear ticket

- **Actor:** Usuario
- **Descripción:** El usuario llena un formulario y crea el ticket.
- **Resultado:** El ticket queda en estado “Abierto” y se registra fecha/hora.

❖ CU-02 Asignar ticket

- **Actor:** Admin/Supervisor
- **Descripción:** Se asigna un ticket a un agente.
- **Resultado:** Queda guardado quién lo atenderá y desde cuándo.

❖ CU-03 Responder y cambiar estado

- **Actor:** Agente
- **Descripción:** El agente responde al ticket y puede cambiarlo a “En progreso”.
- **Resultado:** Se guarda respuesta + historial del cambio.

❖ CU-04 Configurar SLA

- **Actor:** Admin
- **Descripción:** Define tiempos de SLA por prioridad.
- **Resultado:** El sistema usa esos tiempos para medir cumplimiento.

❖ **CU-05 Consultar métricas SLA**

- **Actor:** Admin/Supervisor
- **Descripción:** Ve estadísticas: tickets por estado, promedio de respuesta, cumplimiento SLA.
- **Resultado:** Permite controlar desempeño del soporte.

❖ **CU-06 Gestionar artículos**

- **Actor:** Admin/Agente
- **Descripción:** Crea o edita un artículo de ayuda.
- **Resultado:** Se genera una nueva versión del artículo.

Criterios de aceptación

- **Para RF-01:** Dado que estoy logueado como usuario, cuando lleno título, descripción, categoría y prioridad y le doy “Crear”, entonces el sistema registra el ticket y lo muestra en mi lista.
- **Para RF-10:** Dado que soy admin, cuando defino SLA para “Crítico”, entonces el sistema lo guarda y lo usa para medir tickets críticos.
- **Para RF-16:** Dado que edito un artículo ya publicado, cuando guardo cambios, entonces el sistema crea una nueva versión y conserva la anterior.

Repository y estructura

Para esta entrega, creamos el repositorio desde ya, como se pidió, y lo organizamos con la estructura mínima:

<https://github.com/KHOOPPER/helpdesk>

- **/docs** - aquí van a ir los PDFs de cada entrega
- **/src** - aquí irá el código fuente de la aplicación en las siguientes entregas
- **/tests** - aquí irán las pruebas del sistema
- **/db** - aquí irán scripts o estructura de la base de datos cuando la trabajemos

También dejamos un **README** en la raíz explicando de forma simple qué va en cada carpeta y el enlace al documento.

Conclusión

En esta primera entrega nos enfocamos en definir bien qué debe hacer el sistema, qué roles tendrá, qué reglas de negocio lo controlan y cómo se medirá el SLA. Con esto ya queda claro el “mapa” del proyecto para que en las próximas unidades avancemos con diseño, pruebas, implementación y despliegue.