

Análisis del Sistema y Modelado UML: Sistema de Tickets de Servicio

1. Análisis del Sistema

Objetivo del Sistema: Desarrollar un Sistema de Tickets de Servicio en Java con una interfaz gráfica en JavaFX, permitiendo la gestión eficiente de solicitudes de soporte dentro de una organización. El sistema facilitará la creación, seguimiento y resolución de tickets, además de ofrecer configuraciones para personalizar el flujo de trabajo y los roles de los usuarios.

Alcance del Sistema:

- Configuración de parámetros generales del sistema.
- Gestión de roles y permisos.
- Registro de departamentos y áreas de trabajo.
- Registro de usuarios (Administrador, Técnico, Usuario).
- Definición de estados de los tickets.
- Configuración de flujos de trabajo de los tickets.
- Creación, modificación y consulta de tickets.
- Seguimiento de solicitudes pendientes.
- Cambio de estado de un ticket.
- Agregar notas a un ticket.

Actores del Sistema:

1. **Administrador:**
 - Configura los parámetros del sistema.
 - Gestiona roles, permisos, usuarios y departamentos.
 - Supervisa el flujo de trabajo de los tickets.
2. **Técnico:**
 - Consulta la lista de tickets asignados.
 - Cambia el estado de un ticket.
 - Agrega notas a los tickets.
3. **Usuario:**
 - Crea nuevos tickets de servicio.
 - Consulta el estado de sus tickets.

Requisitos No Funcionales:

- El sistema debe ser robusto y manejar excepciones adecuadamente.
- Persistencia de datos en PostgreSQL en la nube.
- Guardado de respaldo mediante serialización de objetos a archivos.
- Interfaz gráfica intuitiva desarrollada en JavaFX.

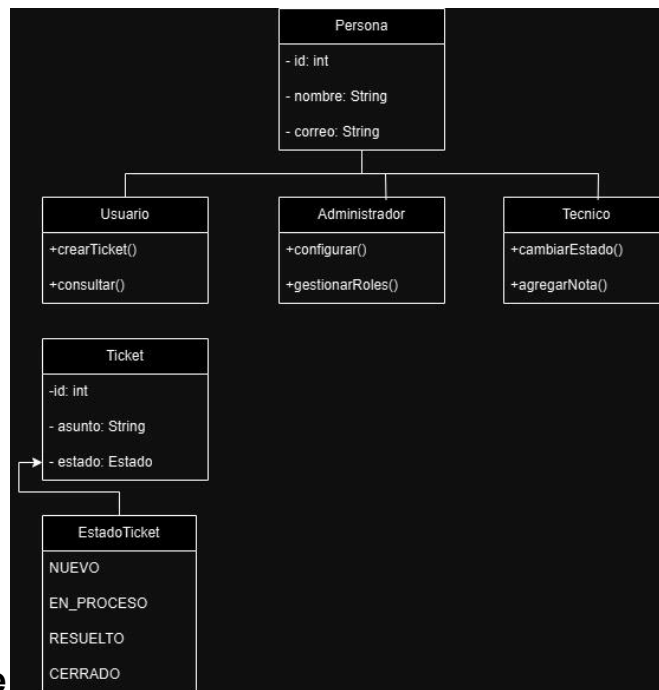
2. Identificación de Clases y Relaciones

Clases Principales:

1. **Persona (abstracta):**
 - id: int
 - nombre: String
 - correo: String
2. **Administrador (hereda de Persona):**
 - configurarParametros()
 - gestionarRoles()
3. **Técnico (hereda de Persona):**
 - cambiarEstadoTicket()
 - agregarNota()
4. **Usuario (hereda de Persona):**
 - crearTicket()
 - consultarEstado()
5. **Ticket:**
 - id: int
 - asunto: String
 - descripcion: String
 - estado: EstadoTicket
 - historialCambios: Stack
 - agregarNota(nota: String)
6. **Departamento:**
 - id: int
 - nombre: String
 - colaTickets: Queue
7. **EstadoTicket (enumerado):**
 - NUEVO, EN_PROCESO, RESUELTO, CERRADO

Relaciones entre Clases:

- Persona es una clase abstracta que es heredada por Administrador, Técnico y Usuario (Herencia).
- Cada Ticket está asociado a un Departamento (Composición).
- El historial de cambios de un Ticket se almacena en una pila (Asociación).
- Los Departamentos gestionan una cola de tickets (Asociación).



3. Diagrama de Clase

4. Diagrama de Casos de Uso

Casos de Uso Principales:

1. Configurar Parámetros del Sistema (Administrador)
2. Gestionar Roles y Permisos (Administrador)
3. Registrar Departamentos (Administrador)
4. Registrar Usuarios (Administrador)
5. Definir Estados de los Tickets (Administrador)
6. Configurar Flujos de Trabajo (Administrador)
7. Gestionar Tickets (Usuario/Técnico/Administrador)
8. Consultar Lista de Solicitudes Pendientes (Técnico/Administrador)
9. Cambiar Estado de un Ticket (Técnico/Administrador)
10. Agregar Nota a un Ticket (Técnico/Administrador)

