

**1.**

**UNIVERSIDAD INTERNACIONAL SAN ISIDRO LABRADOR S.A**

**Bachillerato en ingeniería de sistemas**

**(ISB-27) INGENIERÍA Y UTILIZACIÓN DE SOFTWARE**

Presenta

**Álvaro Daniel Rodríguez Morales**

Profesor

**RODRIGUEZ ACUÑA RUDDY GABRIEL**

Proyecto Final

**Sistema de Gestión de Tickets**

**Agosto 06**

**2.**

El presente documento describe el diseño, desarrollo e implementación del **Sistema de Gestión de Tickets y Pendientes**, una solución web desarrollada en ASP.NET MVC con conexión a SQL Server.  
Este sistema fue creado para la administración eficiente de tickets de soporte en una empresa con más de 5000 colaboradores, permitiendo gestionar solicitudes de soporte, asignación de roles, seguimientos, reportería y control de tickets finalizados.

**3.**

Las empresas con gran volumen de usuarios y solicitudes de soporte requieren un sistema que centralice la gestión de incidencias. Este sistema:

* Optimiza la atención de tickets mediante roles de Usuario, Soportista y Supervisor.
* Mejora la comunicación mediante notificaciones (por correo).
* Facilita la trazabilidad mediante una bitácora de tickets finalizados y reportería avanzada.

**4. Objetivos**

**4.1. Objetivo General**

Desarrollar un sistema web para la gestión de tickets y pendientes que permita administrar las solicitudes de soporte técnico de forma eficiente.

**4.2. Objetivos Específicos**

* Implementar un sistema de login con control de roles y permisos.
* Permitir la creación, edición, eliminación y seguimiento de tickets.
* Asignar soportistas y categorías con tiempos estimados de resolución.
* Generar reportería de tickets por estado, usuario, tipo, categoría y soportista.
* Mantener una bitácora de tickets finalizados para consulta histórica.
* Enviar avances del ticket al usuario mediante correo electrónico.

**5. Requisitos Funcionales**

1. **Login con logo y control de roles.**
2. **Usuarios:**
   * Crear, editar y marcar usuarios como activos/inactivos.
   * Roles: Usuario, Soportista, Supervisor.
3. **Tickets:**
   * Crear, editar, eliminar tickets.
   * Adjuntar evidencia (imágenes o PDF).
   * Asignar categoría (Bajo, Medio, Alto).
   * Cambiar estado (Abierto, En revisión, En proceso, Finalizado).
4. **Seguimientos:**
   * Agregar comentarios de soporte con fecha y hora.
   * Notificación al usuario por correo.
5. **Bitácora:**
   * Al finalizar un ticket, se guarda en una bitácora.
6. **Reportes:**
   * Filtrar tickets por estado, usuario, tipo, categoría, soportista.

**6. Arquitectura del Sistema**

* **Patrón:** ASP.NET MVC (Modelo, Vista, Controlador).
* **Capas:**
  + **Entidades (Modelos):** Representan tablas de la base de datos.
  + **AccesoDatos:** Procedimientos almacenados y lógica ADO.NET.
  + **LogicaNegocio:** Contiene la lógica del negocio.
  + **Web:** Interfaz gráfica (vistas Razor).
* **Base de datos:** SQL Server con procedimientos almacenados para CRUD.

**7. Módulos del Sistema**

* **Login:** Validación de credenciales, control de sesión y roles.
* **Gestión de Usuarios:** Alta, edición, listado y baja lógica (activo/inactivo).
* **Gestión de Tickets:** CRUD de tickets con evidencias y filtros avanzados.
* **Seguimientos:** Registro y visualización de comentarios de soporte.
* **Bitácora:** Listado histórico de tickets finalizados.
* **Reportería:** Filtros por estado, usuario, tipo, categoría y soportista.

**8. Conclusiones**

El sistema desarrollado cumple con los requerimientos funcionales planteados, optimizando la atención de tickets y garantizando una trazabilidad clara mediante seguimientos y bitácora. Además, su integración con correo electrónico mejora la comunicación con los usuarios.