

Práctica 4 — Gestor de incidencias internas (Helpdesk)

1. Objetivo de la práctica

Desarrollar un sistema web que permita registrar, gestionar y resolver incidencias internas dentro de una empresa o centro educativo. El sistema debe incluir control de acceso por roles, gestión de incidencias, comentarios, estados y filtrado.

El resultado debe ser una aplicación funcional, clara y bien estructurada.

2. Descripción general de la aplicación

La aplicación debe incluir:

- Un sistema de login con roles diferenciados.
- Un panel principal con acceso a incidencias y estadísticas.
- Un CRUD de incidencias (según rol).
- Un sistema de cambio de estado de las incidencias.
- Un sistema de comentarios dentro de cada incidencia.
- Listados de incidencias con filtros y ordenación.
- Uso de arrays para generar estadísticas básicas.
- Clases PHP que representen usuarios, incidencias y comentarios.

3. Requisitos funcionales obligatorios

3.1. Autenticación y roles

El sistema debe tener, como mínimo, los siguientes roles:

Administrador

- CRUD completo de incidencias.
- Ver todas las incidencias.
- Cambiar el estado de cualquier incidencia.
- Eliminar incidencias.
- Gestionar usuarios.

Técnico

- Ver todas las incidencias.
- Cambiar el estado de las incidencias.
- Añadir comentarios.
- No puede eliminar incidencias.

Usuario

- Crear incidencias.
- Ver sus propias incidencias.
- Añadir comentarios en sus incidencias.
- No puede cambiar estados ni ver incidencias de otros usuarios.

Requisitos generales del login

- Formulario de login con usuario y contraseña.
- Comprobación de credenciales contra la base de datos.
- Uso de sesiones para mantener la sesión iniciada.
- Páginas protegidas según rol.
- Página de logout que destruya la sesión.

3.2. Gestión de incidencias (CRUD completo)

Cada incidencia debe tener:

- Usuario que la crea
- Título
- Descripción
- Prioridad (baja, media, alta)
- Fecha de creación
- Fecha de resolución (si aplica)
- Estado (abierta, en proceso, resuelta)

El sistema debe permitir:

- Crear incidencias (usuario, técnico, administrador).
- Editar incidencias (solo técnico y administrador).
- Cambiar el estado de una incidencia.
- Eliminar incidencias (solo administrador).
- Listar incidencias en una tabla HTML con filtros.

3.3. Sistema de comentarios

Cada incidencia debe permitir añadir comentarios.

Cada comentario debe tener:

- Usuario que comenta
- Texto del comentario
- Fecha y hora del comentario

Requisitos:

- Los usuarios pueden comentar solo en sus propias incidencias.
- Técnicos y administradores pueden comentar en cualquier incidencia.
- Los comentarios deben mostrarse en orden cronológico.

3.4. Filtros y ordenación

Al listar incidencias, el sistema debe permitir:

- Filtrar por estado (abierta, en proceso, resuelta).
- Filtrar por prioridad.
- Filtrar por usuario (solo técnico y administrador).
- Ordenar por fecha de creación o prioridad.

3.5. Posibles estructuras de almacenamiento *(Tienen libertad si consideran otras formas de estructuración)*

A) Array de incidencias. Al cargar las incidencias desde la BD:

- Crear un array de objetos Incidencia.
- Permitir filtrar y ordenar usando arrays.
- Permitir agrupar por estado o prioridad.

B) Estadísticas básicas

Generar estadísticas usando arrays, por ejemplo:

- Número de incidencias por estado.
- Número de incidencias por prioridad.
- Incidencias resueltas en la última semana.

Estas estadísticas pueden mostrarse en el panel principal.

3.6. Programación orientada a objetos. Debes crear, como mínimo, las clases:

Clase Usuario

- Propiedades básicas (id, nombre, rol)
- Constructor
- Métodos getters/setters

Clase Incidencia

- Propiedades: usuario, título, descripción, prioridad, fechas, estado
- Constructor
- Método *cambiarEstado(\$nuevoEstado)*

- Método *estaResuelta()*
- Métodos getters/setters

Clase Comentario

- Propiedades: usuario, incidencia, texto, fecha
- Constructor
- Métodos getters/setters

3.7. Validación de formularios

- Título obligatorio, mínimo 3 caracteres
- Descripción obligatoria
- Prioridad válida (lista predefinida)
- Estado válido (lista predefinida)
- Comentarios no vacíos
- Se pueden usar expresiones regulares donde sea adecuado

4. Extras opcionales

- Notificación mediante cookie cuando una incidencia del usuario cambia de estado
- Buscador de incidencias por texto
- Sistema de adjuntos (subir archivos a una incidencia)
- Historial de cambios de estado