Examen – Plataforma: Sistema de Gestión de Incidencias "HelpDesk Pro"

ACLARACIONES: No se permite el uso de internet más allá del manual de PHP o la Moodle para acceder a la tarea correspondiente. Las pantallas deberán grabarse desde el inicio de la prueba. No se corregirá ninguna prueba que no incluya el correspondiente vídeo en el que se muestre la realización completa del examen. La **no entrega del vídeo** junto con la prueba o la detección de cualquier actividad sospechosa supondrá la calificación de **suspenso automático**. Se valorará, además de la funcionalidad del programa, la calidad del código, claridad, legibilidad, indentaciones, uso de nombres apropiados, modularidad, limpieza del código, organización general y buenas prácticas de programación.

Se entregará todo comprimido en un zip en la propia tarea creada en la Moodle.

Contexto

La empresa "HelpDesk Pro" necesita una aplicación interna para gestionar incidencias técnicas. El sistema permite que los *usuarios* registren problemas y los *administradores* los resuelvan o eliminen.

Se proporciona el esquema de base de datos (*database.sql*) y un modelo HTML con estilos tanto para el login como el dashboard (*login.php* y *dashboard.php*). Además, la expresión regular para validar el email (*otros.txt*).

Apartado 1 – Arquitectura de clases (1.75 puntos):

Diseña un modelo de clases en un archivo llamado *clases.php* que cumpla con:

- Una interfaz, *IInformable*, para estandarizar la salida de información de los objetos, es decir, con un método *getResumen*().
- Una clase base, *Ticket*, que gestione los datos comunes y valide el formato del correo electrónico del técnico. Tendrá como atributos: *título* y *email* y el método *getResumen()* como abstracto.
- Una clase derivada, *Incidencia*, que gestione la prioridad y determine visualmente el estado de la incidencia (colores) en la tabla del panel.
    - Tendrá como atributo privado: prioridad (tendrá como valores, Baja, Media y Alta).
    - Método *getResumen()*: Devuelve el título en mayúsculas y la prioridad entre corchetes. Ejemplo:

      NUEVA INCIDENCIA: Prueba1 [Baja]

    - Método *getEstilo($resuelto)*: Devuelve una cadena de estilos. Si la incidencia está resuelta, fondo verde, si no está resuelta y es de prioridad Alta, fondo rojo y texto en negrita. (Mirar archivo: *otros.txt*).

Nota: No es obligatorio implementar los métodos Getter ni Setter en ninguna clase. El sistema de colores debe priorizar el estado "Resuelto" sobre cualquier otro.

Apartado 2 (1.75 puntos):

Debes implementar un sistema de autenticación (login.php) que proteja el acceso al panel usando sentencias preparadas. Además, tras iniciar sesión con éxito, el sistema debe recordar (durante 24h) el último correo electrónico utilizado para iniciar sesión mediante mecanismos de persistencia en el cliente y que se mostrará automáticamente en el campo del formulario al volver a entrar.

Apartado 3 (2.5 puntos):

En *dashboard.php*, el listado de incidencias debe cumplir:

- Muestre un listado completo que muestre: Título de la incidencia, prioridad, nombre de la categoría, técnico (nombre del usuario) y estado.
- Los administradores deben ver primero las incidencias de prioridad alta, mientras que el resto de usuarios verán solo sus propias incidencias por orden de fecha.
- Se aplique la lógica de estilos definida en las clases para diferenciar cada fila de la tabla.

Apartado 4: Lógica de Estados (2 puntos):

- Alta de incidencias (usuario): será quien registre nuevas incidencias. Al finalizar, mostrará el resumen de la incidencia creada.
- Resolución de incidencias (admin): Permite a los administradores marcar incidencias como finalizadas.
- Borrado de incidencias (admin): El administrador puede seleccionar varias incidencias mediante *checkboxes* y eliminarlas, pero por seguridad, el sistema solo debe permitir borrar aquellas que ya hayan sido resueltas previamente.

Apartado 5: Seguridad y código (2 puntos): Se deberá garantizar un flujo de navegación seguro mediante el uso correcto de sesiones y cookies (si las hubiese), asegurando que el acceso al panel esté estrictamente protegido y que las redirecciones se ejecuten de forma coherente en todo momento. Se exigirá una implementación impecable de la seguridad en las consultas. Asimismo, se valorará la calidad interna del código, la cual debe destacar por una indentación limpia, una organización modular lógica y una semántica apropiada en el nombrado de variables y funciones. Finalmente, el programa deberá ejecutarse de manera fluida y profesional.

Adicional: Token de Seguridad Anti-F5 (+1 punto extra): Optimiza el formulario de registro para evitar la duplicidad de datos producida por el refresco accidental de la página (F5) o reenvíos del navegador. El sistema debe invalidar la petición una vez procesada correctamente, es decir, al registrar una nueva incidencia.