**Desarrollo para front end**

**Unidad 3**

**Integradora 3. Animaciones y efectos**

****

**Alumno: OMAR ALONSO GARCIA CASTELLANOS**

**Carlos Alejandro Mendoza Alvarez**

**19/11/2024**

1. Comprensión de los Requerimientos

El primer paso fue analizar los requerimientos planteados. El objetivo era crear una interfaz funcional y profesional para la gestión de datos en una tabla de la base de datos gestion\_personas. Los aspectos clave requerían:

Conexión a la base de datos MySQL.

Implementación de funcionalidades CRUD (Crear, Leer, Actualizar y Eliminar).

Uso de AJAX para solicitudes asíncronas.

Animaciones para la adición y eliminación de registros.

Uso de "dialogs" para manejar notificaciones y errores.

Implementación de filtros (búsqueda, ordenamiento, paginación).

Validación de datos y manejo seguro de sesiones mediante tokens.

Estilo visual y profesional con CSS.

2. Preparación del Entorno

Base de datos: Creé la base de datos gestion\_personas y la tabla personas con los campos: id, apellido, correo, telefono, direccion y fecha\_registro.

Archivo SQL: Se generó un archivo SQL para estructurar y poblar la tabla con datos iniciales, facilitando las pruebas.

Servidor local: Usé un entorno local de servidor (XAMPP) para ejecutar PHP y MySQL.

3. Desarrollo de Archivos

Se crearon los siguientes archivos, organizados funcionalmente:

config.php Configura la conexión segura a la base de datos con PDO.

session.php Implementa la seguridad mediante sesiones y tokens para prevenir accesos no autorizados.

index.php Estructura principal del proyecto con filtros, tabla dinámica y llamadas a AJAX.

get\_personas.php Responde a las solicitudes AJAX para obtener los registros, aplicar filtros y paginación.

script.js Contiene la lógica de frontend: AJAX para solicitudes CRUD, animaciones, y manejo de errores/notificaciones con "dialogs".

styles.css Proporciona una apariencia profesional y limpia, incluyendo estilos para la tabla y los botones.

4. Validación y Pruebas

Se realizaron pruebas exhaustivas para garantizar que cada funcionalidad operara como se esperaba:

Conexión a la Base de Datos: Verifiqué la conexión usando mensajes de depuración en PHP.

CRUD: Probé las operaciones de agregar, modificar y eliminar registros desde la interfaz.

Filtros: Validé la búsqueda, el orden ascendente/descendente y la paginación.

AJAX: Comprobé que las solicitudes fueran asíncronas y retornaran respuestas correctamente.

Animaciones: Verifiqué las transiciones al agregar o eliminar registros y al actualizar la tabla.

Notificaciones: Implementé y probé los diálogos informativos y de error.

5. Resolución de Problemas

Durante el desarrollo enfrenté los siguientes desafíos:

Problema:  
Al principio, las animaciones no se aplicaban correctamente al agregar o eliminar registros.  
Solución:  
Modifiqué el DOM utilizando métodos como .hide().fadeIn() y .fadeOut().

Problema:  
La búsqueda y los filtros no retornaban datos esperados cuando se usaba una combinación de filtros.  
Solución:  
Ajusté las consultas SQL para que incluyeran parámetros dinámicos correctamente con LIKE y cláusulas LIMIT.

Problema:  
La validación del token en el archivo session.php no se aplicaba en todas las rutas.  
Solución:  
Incluí verificaciones explícitas en cada archivo PHP que interactúa con la base de datos.

Problema:  
Durante la eliminación de registros, no se actualizaban automáticamente los datos mostrados en la tabla.  
Solución:  
Agregué una recarga dinámica de los datos tras cada operación exitosa.

Conclusión

El proyecto resultó en una aplicación funcional que cumple con todos los requisitos planteados. Implementar animaciones y manejar AJAX de forma eficiente mejoró la experiencia del usuario. El diseño profesional y la incorporación de notificaciones mejoraron la interacción.

Dificultades:  
El mayor reto fue integrar animaciones sin comprometer el rendimiento de las solicitudes AJAX, así como garantizar la seguridad mediante tokens.

Aprendizajes:  
Este proyecto reforzó mis conocimientos sobre la interacción entre frontend y backend, optimización de consultas SQL y la importancia de validar datos para evitar vulnerabilidades.

En general, el trabajo realizado fue satisfactorio y cumplió con los objetivos propuestos.

