



DESARROLLO DE APLICACIONES WEB CON SOFTWARE INTERPRETADO EN EL SERVIDOR – DSS404

Facultad de Ingeniería
Escuela de Computación
Técnico en Ingeniería en Computación

Desafío práctico #2

Competencias

- Análisis de problemas para la correcta interpretación funciones en PHP.
- Desarrolla aplicaciones para PHP utilizando la lógica de servicio REST y sesiones

Fecha de desarrollo

Semana 12

Entrega: Sábado 12/04/2025

Se entregará un archivo .zip con los HTML y scripts PHP creados, en el espacio asignado por el docente en el aula digital. Además, se realizará una defensa donde presentaran brevemente (10-15 min) los ejercicios realizados, se compartirá un documento de Excel para que inscriban su horario de defensa.

Forma de desarrollo: Individual o en parejas

Si el estudiante o estudiantes presenta un documento igual al de otro será considerado plagio y ambos grupos tendrán una nota de 0.0

Enunciado

Desarrollar una aplicación web en PHP que permita a los usuarios autenticados gestionar sus documentos personales de manera segura, con funcionalidades CRUD, almacenamiento en el servidor y una API REST pública para acceder a la lista de los archivos subidos.

Requisitos Funcionales

- **Registro e inicio de sesión de usuarios**
 - El sistema debe permitir a los usuarios registrarse y autenticarse.
 - Usa sesiones y cookies para mantener la sesión activa.
 - Las contraseñas de los usuarios deben almacenarse encriptadas en la base de datos.
- **Gestión de documentos**
 - Una vez autenticado, el usuario podrá ver una lista de sus documentos, subir documentos (PDF, DOCX, TXT) o eliminarlos.
 - Los archivos deben almacenarse en el sistema de archivos del servidor, organizados en carpetas por usuario.

- La información de cada archivo (nombre, fecha de subida, ruta, tipo) debe almacenarse en una base de datos MySQL.
- **Validación con Expresiones Regulares**
 - Valida campos del formulario (correo, contraseña, nombres de archivos, etc.) usando expresiones regulares.
- **API REST**
 - Implementa un servicio web REST que permita:
 - Obtener lista de todos los documentos almacenados.
- **SQL**
 - Query para la creación de la base de datos

Requisitos Técnicos

- Usar **Programación Orientada a Objetos** (clases para usuarios, archivos, autenticación, servicios API).
- Utilizar **expresiones regulares** para validaciones.
- Almacenar y recuperar información de una base de datos **MySQL**.
- Manipular archivos y carpetas usando funciones nativas de PHP.
- Implementar **sesiones y cookies** para la autenticación web.
- Implementar un **WebService RESTful** con respuestas en JSON.

Diccionario de datos

Tabla: usuarios

Campo	Tipo de Dato	Restricción	Descripción
Id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador único del usuario
nombre	VARCHAR(100)	NOT NULL	Nombre completo del usuario
Email	VARCHAR(150)	NOT NULL, UNIQUE	Correo electrónico del usuario
password_hash	VARCHAR(255)	NOT NULL	Contraseña hasheada
fecha_registro	DATETIME	DEFAULT CURRENT_TIMESTAMP	Fecha de creación del usuario

Tabla: archivos

Campo	Tipo de Dato	Restricción	Descripción
Id	INT	PRIMARY KEY, AUTO_INCREMENT	Identificador único del archivo
usuario_id	INT	NOT NULL, FOREIGN KEY (usuarios.id)	ID del usuario que subió el archivo
nombre	VARCHAR(255)	NOT NULL	Nombre original del archivo

Ruta	VARCHAR(255)	NOT NULL	Ruta del archivo en el sistema de archivos
Tipo	VARCHAR(20)	NOT NULL	Tipo de archivo (ej: pdf, docx, txt)
fecha_subida	DATETIME	DEFAULT CURRENT_TIMESTAMP	Fecha y hora de la subida del archivo

Test a realizar

Autenticación y Sesiones

- 1. Registro válido**
Registrar un usuario con datos correctos.
→ Se espera: Usuario creado correctamente, sesión iniciada.
- 2. Registro con email inválido**
Probar registro con email mal formado.
→ Se espera: Mensaje de error y no se crea usuario.
- 3. Inicio de sesión correcto**
Ingresar con email y contraseña válidos.
→ Se espera: Acceso al panel de usuario.
- 4. Acceso sin sesión**
Intentar acceder a zonas privadas sin haber iniciado sesión.
→ Se espera: Redirección al login o mensaje de acceso denegado.

Gestión de Archivos y Directorios

- 5. Subida de archivo válido**
Subir un archivo permitido (PDF, TXT, DOCX).
→ Se espera: Archivo guardado correctamente, entrada creada en la base de datos.
- 6. Subida de archivo no permitido**
Intentar subir archivo con extensión prohibida (.exe, .js, etc.).
→ Se espera: Rechazo con mensaje de error.
- 7. Estructura de directorios por usuario**
Verificar que los archivos se almacenen en carpetas individuales por usuario.
→ Se espera: Rutas separadas, por ejemplo /uploads/usuario_5/documento.pdf.
- 8. Eliminación de archivos**
Eliminar archivo desde el panel.
→ Se espera: Se borra del disco y de la base de datos.

API REST

- 9. Consulta de archivos vía API**
Llamar al servicio REST para obtener la lista de archivos.
→ Se espera: Lista de archivos en formato JSON.

Rubrica de evaluación				
Actividad a evaluar	Criterio a evaluar	Cumplió		Puntaje
		Si	No	
Actividad (100%)	Creación de la base de datos relacional funcional y conexión correcta (10%)			
	Implementación de Programación Orientada a Objetos en el código solicitado (10%)			
	Implementación del manejo de archivos para ser almacenados según lo indicado (10%)			
	Implementación de las sesiones y cookies es correcta (10%)			
	Implementación validaciones en el back end con expresiones regulares (10%)			
	Implementación de Servicio REST obtener la lista de archivos (15%)			
	Pruebas descritas para la aplicación funcionan correctamente (20%)			
	Defensa individual (15%)			
Total				