

Sistema de Mensajería - Resumen y Explicación

1. Descripción General

El sistema es una plataforma web básica de mensajería interna entre usuarios, con dos tipos de roles: **administrador** y **usuario común**. Los usuarios pueden registrarse, iniciar sesión, enviar mensajes, leerlos, responder, reenviar y eliminar mensajes. Los administradores tienen además opciones para administrar usuarios (crear, modificar, eliminar).

2. Archivos principales y su función

Archivo	Función principal
index.php	Pantalla de inicio / login. Valida usuario y contraseña.
admin.php	Panel para administradores. Gestiona usuarios y acceso a mensajes.
usuario.php	Panel para usuarios comunes. Acceso a sus mensajes.
mensajes.php	Maneja la bandeja de entrada, salida, creación, respuesta y eliminación de mensajes.
registro.php	Página para registrar nuevos usuarios (no mostrada aquí).
logout.php	Cierra sesión del usuario.
claseFinal.php	Define las clases Usuarios y Mensajes con sus métodos para manejar la lógica (crear, eliminar, modificar, listar).
clase.Usuario.php	Define la clase Usuario para manejar un solo usuario (por ejemplo, para obtener datos de un usuario).
conexion.php	Contiene la función conectar() para conectar con la base de datos MySQL.
stylesheet.css	Estilos para el diseño y formato visual de la página.

2. Archivos y Funciones Clave

index.php

- Entrada al sistema.
- Recibe usuario y contraseña.
- Usa el método Usuarios:: iniciarSesion() para validar.
- Según el rol, redirige a admin.php o usuario.php.

admin.php

- Acceso exclusivo para administradores.

- Verifica la sesión y el rol antes de mostrar el contenido.
- Permite:
 - Crear usuarios (alta).
 - Buscar usuarios (busqueda).
 - Listar usuarios (lista).
 - Modificar o eliminar usuarios.
- También tiene acceso a mensajes.php para gestionar mensajes.

usuario.php

- Panel para usuarios comunes.
- Sólo se accede si la sesión es activa y el rol es 2.
- Ofrece botones para:
 - Ver mensajes (mensajes.php).
 - Enviar mensajes (mensajes.php con la opción nuevo mensaje).

mensajes.php

- Maneja toda la funcionalidad de mensajes (recibidos, enviados, crear, responder, reenviar, eliminar).
- Usa variables GET para navegar entre las vistas.
- Usa métodos de la clase Mensajes para operar con la base de datos.
- Permite enviar mensajes sin necesidad de un archivo separado.
- Incluye comprobaciones como evitar enviarse mensajes a sí mismo.

claseFinal.php : Contiene dos clases fundamentales:

Admin

- Métodos importantes:
 - **alta()**: Registra usuario nuevo tras validar que no exista.
 - **eliminar(\$id)**: Elimina usuario por id.
 - **modificar(\$id)**: Modifica datos de usuario.
 - **listar()**: Devuelve un array con todos los usuarios.
 - **iniciarSesion(\$usuario, \$pass)**: Verifica credenciales y crea sesión.
- Importante: la contraseña se compara tal cual, no cifrada (mejorar en producción).

clase.Usuario.php

- Clase más simple que encapsula datos de un usuario.
- Usa consultas preparadas para evitar inyección SQL.

- Método obtenerPorUsuario(\$usuario) devuelve objeto Usuario con datos completos.

Explicaciones de Código Importantes

Control de acceso con sesiones y roles

- Garantiza que sólo usuarios con sesión activa y rol adecuado vean páginas protegidas.

Envío de mensajes (en mensajes.php)

- Evita que un usuario se envíe mensajes a sí mismo.
- Usa la clase Mensajes para guardar el mensaje.

Listar mensajes con estados

- Los mensajes eliminados no se muestran.
- Se aplica CSS para diferenciar mensajes leídos y no leídos.

Alta de usuario con comprobación

- Verifica que no se repita el nombre de usuario.

Modificación de usuario

- Actualiza datos del usuario identificado por id.

Mejores prácticas y recomendaciones

- Seguridad: Implementar cifrado de contraseñas con password_hash() y password_verify().
- Validación y Sanitización: Usar consultas preparadas para todas las consultas SQL.
- Interfaz: Mejorar mensajes y errores para mayor claridad al usuario.
- Separación de responsabilidades: Mantener la lógica (clases) separada de la presentación (HTML).

3. Base de Datos

- **BdSanchezDario**: Base creada con tablas para usuarios, roles, mensajes, estados, y accesos.
- La tabla usuarios contiene datos de los usuarios, con sus roles.
- La tabla mensajes almacena los mensajes enviados, con referencias a quién envía y recibe.
- estados indica si un mensaje está leído, no leído o eliminado.
- roles diferencian entre administrador (1) y usuario (2).
- accesos registra los inicios de sesión.

4. Funcionamiento de Clases y Objetos

Clase Usuarios (en claseFinal.php)

- **Constructor:** Recibe datos como nombre, apellido, usuario, contraseña y rol.
- **alta():** Registra un nuevo usuario si el nombre de usuario no existe.
- **modificar(\$id):** Modifica datos del usuario dado su ID.
- **eliminar(\$id):** Elimina el usuario por ID.
- **listar():** Devuelve todos los usuarios.
- **iniciarSesion():** Valida usuario y contraseña y crea sesión.

Clase Mensajes (en claseFinal.php)

- **Constructor:** Recibe id del remitente, destinatario, asunto, mensaje y estado.
 - **alta():** Envía un mensaje (inserta en la base de datos).
 - **modificar(\$id):** Cambia estado a leído.
 - **eliminar(\$id):** Marca un mensaje como eliminado.
 - **listar():** Devuelve mensajes recibidos y enviados del usuario en sesión.
 - **buscar(\$id):** Busca un mensaje específico.
-

5. Flujo de usuario

1. Inicio de sesión (index.php):

- Usuario introduce usuario y contraseña.
- Se valida en la base.
- Si es administrador, redirige a admin.php.
- Si es usuario común, redirige a usuario.php.

2. Panel administrador (admin.php):

- Puede gestionar usuarios: crear, modificar, eliminar, buscar.
- Acceso a bandeja de mensajes (mensajes.php).

3. Panel usuario común (usuario.php):

- Puede ver bandeja de mensajes y redactar nuevos mensajes.
- Acceso a mensajes.php para ver mensajes recibidos y enviados.

4. Manejo de mensajes (mensajes.php):

- Lista mensajes recibidos y enviados.
- Permite crear un nuevo mensaje (sin necesidad de archivo aparte).
- Permite marcar mensajes como leídos.

- Permite responder, reenviar y eliminar mensajes.

6. Puntos clave de diseño

- **Un solo archivo para mensajes:** mensajes.php incluye todas las funciones de mensajería (crear, leer, responder, eliminar).
- **Sesiones para control de acceso:** Protege el acceso a cada área según el rol.
- **Interacción con base de datos:** A través de clases que encapsulan consultas SQL.
- **Estilos CSS:** Para interfaz amigable y ordenada.

7. Sugerencias y aspectos importantes

- En producción, las contraseñas deberían guardarse cifradas (ej. password_hash).
- Los mensajes usan estados para controlar si están leídos o eliminados.
- La separación clara entre administración y uso común permite seguridad básica.
- La clase Usuario es más simple y usada para obtener un usuario concreto.
- El sistema está orientado a un uso educativo y es simple para ese fin.

8. Resumen gráfico del sistema (para el PDF)

index.php --> Login --> admin.php (rol=1) --> Gestión usuarios + mensajes.php

|

--> usuario.php (rol=2) --> mensajes.php (mensajes personales)

mensajes.php --> Ver bandeja / Enviar / Responder / Eliminar mensajes

Clases:

Usuarios --> Crear/Modificar/Eliminar/Buscar/Listar usuarios

Mensajes --> Enviar/Listar/Marcar leído/Eliminar mensajes

"Este documento resume el sistema de mensajería interna desarrollado, "

"explicando sus archivos, clases y flujo, de manera clara y sencilla, "

"ideal para entender y presentar en una evaluación."

Explicación de clase Mensajes

Flujo y acceso

"Flujo de usuario:\n"

"1. El usuario inicia sesión con usuario y contraseña.\n"

"2. Según el rol, se redirige a panel administrador o usuario común.\n"

"3. Desde los paneles, se accede a la bandeja de mensajes y otras opciones.\n"

"4. La bandeja permite leer, enviar, responder, reenviar y eliminar mensajes.\n"

"5. El sistema protege páginas y funciones con validación de sesión y rol.\n"

Resumen final

Este sistema básico de mensajería cubre:

- Registro y login de usuarios.
- Control de roles y permisos.
- Gestión administrativa de usuarios.
- Envío, recepción, respuesta y gestión de mensajes.
- Control visual de estados de mensajes.
- Uso de programación orientada a objetos para manejar datos y acciones.