1. Fase 6: Implementación de Roles y Gestión de Permisos

2. Descripción General

En esta fase, el sistema se expande para incluir roles y gestión de permisos. Se asigna un rol específico a los usuarios administradores y se añade una validación estricta para acceder a las funcionalidades avanzadas. Esta implementación garantiza que solo los usuarios autorizados puedan realizar operaciones críticas, como la búsqueda avanzada o la gestión de registros. El enfoque principal está en fortalecer la seguridad y la jerarquía en la administración del sistema.

3. Actualizaciones de Código y Cambios Principales

logeoAdmin.java- Actualización

Se modifica la clase para incluir una validación robusta de inicio de sesión, utilizando listas de usuarios y contraseñas predeterminadas. Además, se integra un control más claro de los intentos de acceso para roles administrativos.

Cambios Principales:

- Validación más estricta de usuarios y contraseñas.
- Mejora en la estructura de mensajes para guiar a los administradores en el acceso al sistema.

Código Actualizado:

```
public void Inicio(Scanner sc) {
    boolean IngresoExitoso = false;

    do {
        TextoCentrado("************************************", 1);
        TextoCentrado("SISTEMA GESTION DE ASAMBLEAS.", 1);
        TextoCentrado("************************************", 1);
        TextoCentrado("Digite su usuario administrador: ", 0);
        usuario = sc.nextLine();
        TextoCentrado("Digite su contraseña: ", 0);
        pass = sc.nextInt();
        sc.nextLine(); // Limpiar el buffer

        for (int j = 0; j < usuarios.length; j++) {
            if (usuario.equals(usuarios[j]) && pass == passDf) {
                IngresoExitoso = true;
                TextoCentrado("BIENVENIDO AL SISTEMA DE ASAMBLEAS.", 1);
                break;
            }
        }

        if (!IngresoExitoso) {
            System.out.println("Usuario o contraseña incorrectos.");
            System.out.println("Intente nuevamente.");
        }

    } while (!IngresoExitoso);
}
```

main.java - Actualización

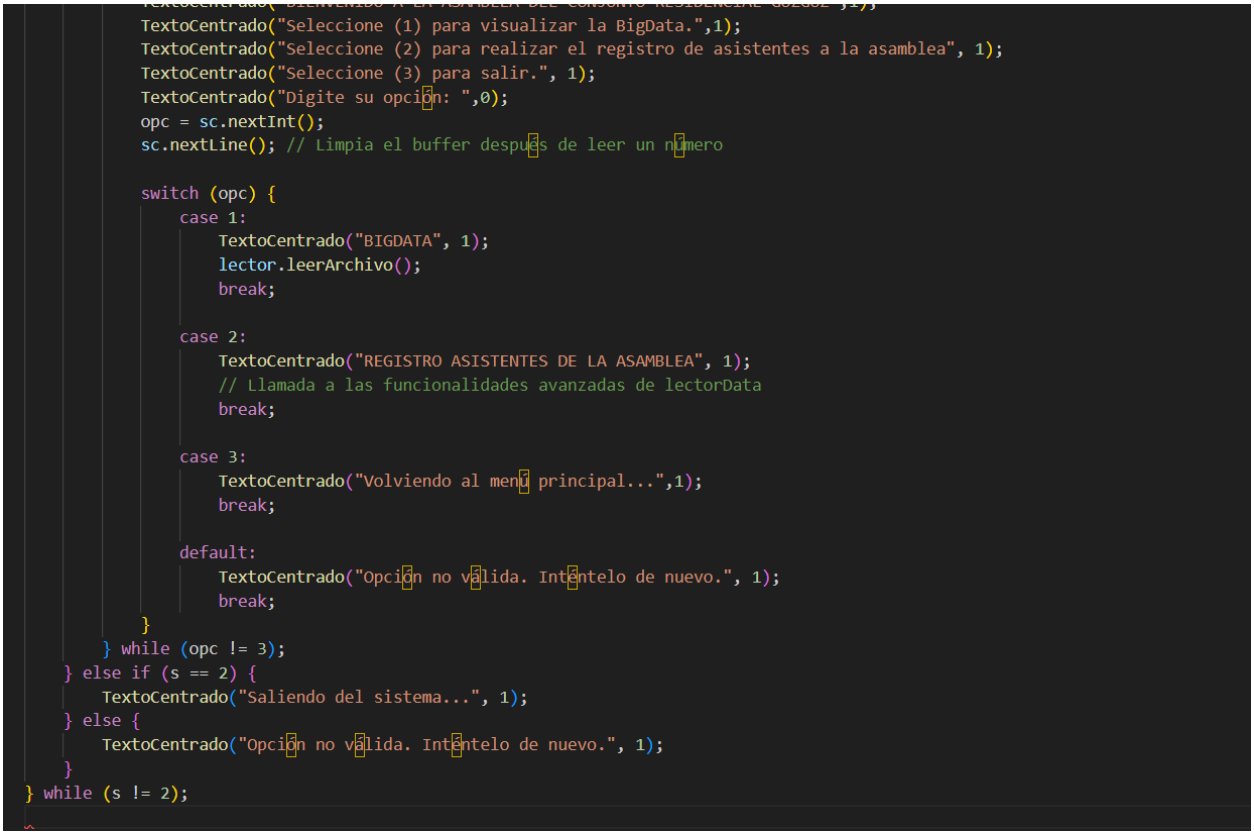
Se integra la validación de permisos para garantizar que las opciones avanzadas del menú estén disponibles solo para usuarios administradores. Si un usuario no está autenticado correctamente, se le impide acceder a ciertas funcionalidades.

Cambios Principales:

- Validación del rol administrativo antes de acceder a las opciones avanzadas del sistema.
- Mejora en la estructura del menú para reflejar los cambios.

Código Actualizado:

```
do {
    TextoCentrado("SISTEMA GESTION DE ASAMBLEAS", 1);
    TextoCentrado("Seleccione (1) para iniciar sesión como administrador", 1);
    TextoCentrado("Seleccione (2) para salir", 1);
    TextoCentrado("Digite su opción: ",0);
    s = sc.nextInt();
    sc.nextLine(); // Limpia el buffer de entrada después de leer un número

    if (s == 1) {
        logeoAdmin login = new logeoAdmin();
        login.Inicio(sc); // Validación de inicio de sesión

        do {
            TextoCentrado("BIENVENIDO A LA ASAMBLEA DEL CONJUNTO RESIDENCIAL GUZGUZ",1);
            TextoCentrado("Seleccione (1) para visualizar la BigData.",1);
            TextoCentrado("Seleccione (2) para realizar el registro de asistentes a la asamblea", 1);
            TextoCentrado("Seleccione (3) para salir.", 1);
            TextoCentrado("Digite su opción: ",0);
            opc = sc.nextInt();
            sc.nextLine(); // Limpia el buffer después de leer un número

            switch (opc) {
                case 1:
                    TextoCentrado("BIGDATA", 1);
                    lector.leerArchivo();
                    break;

                case 2:
                    TextoCentrado("REGISTRO ASISTENTES DE LA ASAMBLEA", 1);
                    // Llamada a las funcionalidades avanzadas de lectorData
                    break;

                case 3:
                    TextoCentrado("Volviendo al menú principal...",1);
                    break;
```



```
            TextoCentrado("Seleccione (1) para visualizar la BigData.",1);
            TextoCentrado("Seleccione (2) para realizar el registro de asistentes a la asamblea", 1);
            TextoCentrado("Seleccione (3) para salir.", 1);
            TextoCentrado("Digite su opción: ",0);
            opc = sc.nextInt();
            sc.nextLine(); // Limpia el buffer después de leer un número

            switch (opc) {
                case 1:
                    TextoCentrado("BIGDATA", 1);
                    lector.leerArchivo();
                    break;

                case 2:
                    TextoCentrado("REGISTRO ASISTENTES DE LA ASAMBLEA", 1);
                    // Llamada a las funcionalidades avanzadas de lectorData
                    break;

                case 3:
                    TextoCentrado("Volviendo al menú principal...",1);
                    break;

                default:
                    TextoCentrado("Opción no válida. Inténtelo de nuevo.", 1);
                    break;
            }
        } while (opc != 3);
    } else if (s == 2) {
        TextoCentrado("Saliendo del sistema...", 1);
    } else {
        TextoCentrado("Opción no válida. Inténtelo de nuevo.", 1);
    }
} while (s != 2);
```

lectorData.java - Actualización

Se adapta para que las funcionalidades críticas (como búsquedas avanzadas) dependan de la validación de administrador.

Cambios Principales:

- La búsqueda avanzada de asistentes solo puede ser ejecutada por usuarios que hayan iniciado sesión con éxito como administradores.
- Se refuerza la seguridad del acceso a las funcionalidades.

Aspectos Técnicos y Consideraciones

Implementación de Seguridad

El sistema asegura que solo los administradores autorizados puedan acceder a funcionalidades avanzadas mediante una validación de inicio de sesión. Aunque las contraseñas son simples y no están encriptadas, esto puede mejorarse en futuras fases. Además, se refuerza el manejo seguro de los datos al evitar accesos no autorizados.

Manejo de Entrada de Usuario

El sistema guía al usuario administrador con mensajes claros para ingresar sus credenciales. Si el acceso falla, se permite reintentar hasta que se logre una autenticación exitosa o se decida salir del sistema.

Estructura de Control

Se integra una jerarquía clara en el acceso a las funcionalidades, dividiendo los accesos entre opciones básicas y avanzadas según el nivel de permisos del usuario. Esto asegura una gestión ordenada y controlada del sistema.

Limitaciones Actuales

- Las contraseñas y usuarios están almacenados de forma estática en el código, lo que limita la escalabilidad y la seguridad.
- No existe un sistema de roles dinámicos o permisos configurables.
- Las credenciales no están protegidas por encriptación.