



Universidad de Sonora

División de Ingeniería

Ing. en Sistemas de Información

Desarrollo de sistemas 2

Profesor Sánchez Schmitz Guzmán Gerardo Alfonso

“Trabajo final control de acceso basado en roles”

Arroyo Lopez Miguel Angel
222214253

Gallardo Carrillo Danna Sofia
222220993

PASOS PARA CREAR UN PROYECTO

Descripción del proyecto:

Crear un control de acceso web basado en roles como un modelo de seguridad que permite asignar funciones y autorizaciones en la infraestructura informática de una organización.

Pasos para crear proyectos:

Planteamiento del problema	Programa en java con clases y paquetes para asignar o controlar los derechos de acceso que posee cada usuario de una organización en función de sus roles y tareas. Los datos ingresados por el usuario deberán ser modificables y almacenados en un sistema de gestión de bases de datos mariadb. Uso de webapps.
Investigar	<ul style="list-style-type: none">• https://en.wikipedia.org/wiki/Role-based_access_control#/media/File:Role-based_access_control.jpg• https://www.visual-guard.com/ES/net-powerbuilder-application-security-authentication-permission-access-control-rbac-articles/dotnet-security-article-ressources/control-acceso-basado-roles.html• https://www.jairogarciarincon.com/clase/interfaces-de-usuario-con-java-swing/layout-managers-o-gestores-de-composicion• https://www.adictosaltrabajo.com/2011/02/25/tutorial-basico-jdbc/• https://www.buscaminegocio.com/cursos-de-java/layouts-en-java.html
Análisis	<p>En general, la implementación del programa implica el diseño de clases y paquetes, la gestión de roles y tareas, la interacción con la base de datos y la consideración de aspectos de seguridad.</p> <ul style="list-style-type: none">• El programa se organiza mediante el uso de clases y paquetes en Java. Los paquetes permiten agrupar clases relacionadas y facilitan la organización y el mantenimiento del código. Con las clases se representarán los usuarios,

	<p>roles, tareas y otros elementos necesarios para la gestión de derechos de acceso.</p> <ul style="list-style-type: none"> • Utiliza un sistema para asignar y controlar los derechos de acceso de los usuarios. Lo que implica definir los roles y las tareas que pueden realizar los usuarios en la organización. Cada usuario puede tener asignado roles que determinan los derechos y privilegios que se les otorgan. • Permite que los usuarios modifiquen los datos relacionados con sus derechos de acceso. Esto implica implementar funcionalidades para agregar, modificar o eliminar roles, tareas y usuarios. Se deben establecer las restricciones adecuadas para garantizar que los cambios se realicen de acuerdo con las reglas establecidas. • Los datos ingresados por el usuario se deben almacenar en un sistema de gestión de bases de datos. Esto implica utilizar el lenguaje SQL para definir la estructura de las tablas necesarias y establecer las relaciones entre ellas. El programa debe incluir la lógica para conectarse a la base de datos, realizar consultas y actualizaciones, y manejar los errores. • Para garantizar la seguridad, el programa debe incluir mecanismos de autenticación para verificar la identidad de los usuarios antes de otorgarles acceso. Esto puede implicar el uso de contraseñas encriptadas, autenticación de dos factores u otras técnicas de seguridad. • Se presenta una interfaz gráfica e intuitiva para que los usuarios puedan realizar las acciones necesarias.
Requisitos	<ul style="list-style-type: none"> • Programar un control de acceso basado en roles. • Crear la interfaz con código. • Utilizar clases, herencia, implementación, paquetes, etc. • Utilizar los conocimientos adquiridos respecto a base de datos.

Diseño	<p>Para el diseño se tomó en cuenta algunas consideraciones generales:</p> <ul style="list-style-type: none"> • Deberá ser intuitivo y fácil de usar para los usuarios. Utilizará elementos visuales claros y organizados, como menús desplegables, botones, listas y pestañas, para facilitar la navegación y la comprensión de las funciones del programa. • Mantendrá una apariencia y estructura coherente en todas las pantallas del programa. Lo que ayudará a crear una experiencia de usuario unificada y profesional. Jerarquía visual. • Iconos, símbolos visuales o mensajes para representar diferentes acciones o estados, lo que ayudará a los usuarios a identificar rápidamente la información relevante. • Diseño responsivo para que el programa se adapte a diferentes tamaños de pantalla y dispositivos. Esto garantizará una experiencia de usuario óptima.
---------------	---

Plan:

Crear el código en java y utilizar mariadb para la implementación de base de datos. Desde intellij se asignan los roles y el control de datos. Al correr el programa, en la pantalla principal los usuarios podrán iniciar sesión y posteriormente acceder y modificar los datos, respectivamente de los privilegios de acceso que se le ha asignado.

Ideas:

- Realizar pruebas de usabilidad.
- Utilizar los layouts.
- Permitir al administrador la asignación de derechos de acceso.
- Aprobación de solicitudes de acceso
- Añadir o eliminar usuarios.
- Permitir el cambio de contraseñas y correos.

Dificultades al hacer el proyecto:

- La interacción con la base de datos, la conexión e implementación adecuada de consultas SQL para leer y escribir datos.

- Definir correctamente los roles y los permisos asociados a cada usuario, ya que, si el diseño de roles es incorrecto, puede resultar en una asignación incorrecta o inconsistente de los derechos de acceso de los usuarios.
- El tener que realizar constantes pruebas debido a que es un proceso de gestión meticuloso, pero que nos garantiza la precisión y la coherencia.
- El manejo de carga de trabajo.

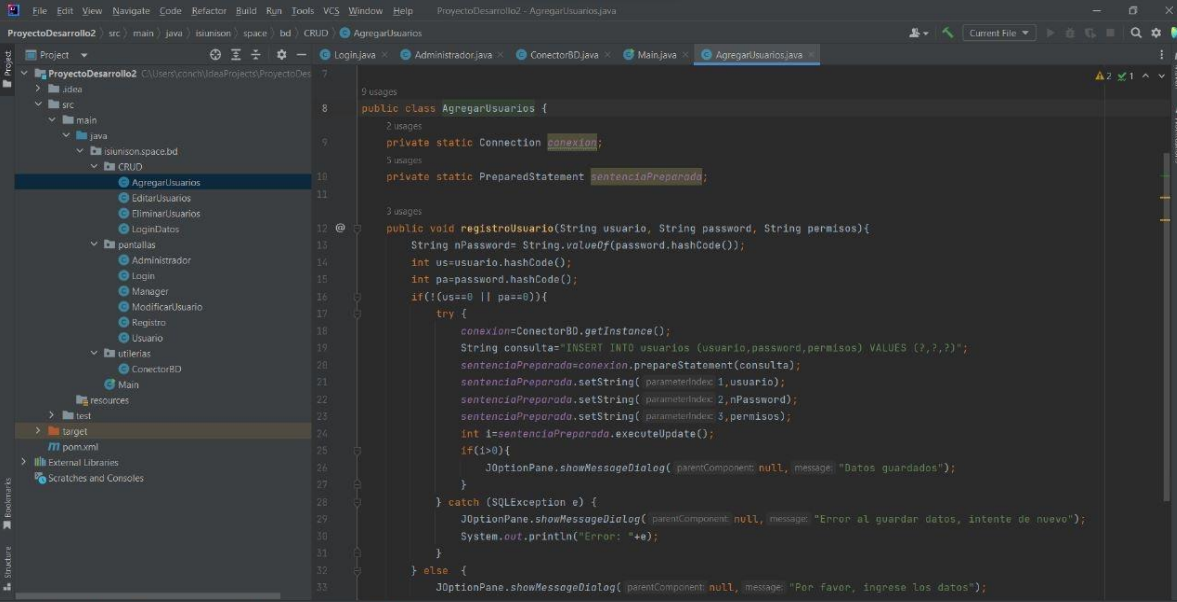
Lo aprendido:

- Las bases para la programación de control de acceso basado en roles.
- Interactuar con base de datos y análisis de la información. Practicar con interfaces.
- Uso de paquetes y clases, interfaces.
- Implementar medidas de seguridad, como autenticación de usuarios, encriptación de datos.

Lo que nos gustaría aprender:

- No hubo tiempo suficiente para aprender acerca de webapps. Por lo que el proyecto fue realizado de manera local.

Capturas...

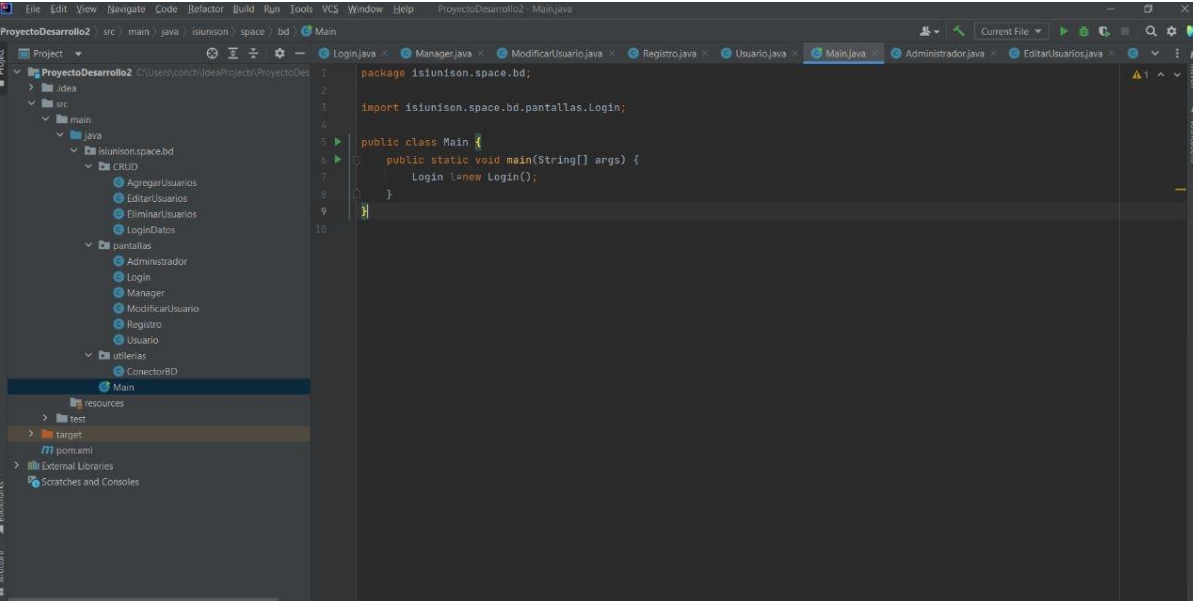


```
7
8 public class AgregarUsuarios {
9
10     private static Connection conexion;
11     private static PreparedStatement sentenciaPreparada;
12
13     3 usages
14     public void registroUsuario(String usuario, String password, String permisos){
15         String nPassword= String.valueOf(password.hashCode());
16         int us=usuario.hashCode();
17         int pa=password.hashCode();
18         if((us==0 || pa==0)){
19             try {
20                 conexion=ConectorBD.getInstance();
21                 String consulta="INSERT INTO usuarios (usuario,password,permisos) VALUES (?,?,?)";
22                 sentenciaPreparada=conexion.prepareStatement(consulta);
23                 sentenciaPreparada.setString( parametros: 1,usuario);
24                 sentenciaPreparada.setString( parametros: 2,nPassword);
25                 sentenciaPreparada.setString( parametros: 3,permisos);
26                 int i=sentenciaPreparada.executeUpdate();
27                 if(i>0){
28                     JOptionPane.showMessageDialog( parentComponent: null, message: "Datos guardados");
29                 }
30             } catch (SQLException e) {
31                 JOptionPane.showMessageDialog( parentComponent: null, message: "Error al guardar datos, intente de nuevo");
32                 System.out.println("Error: "+e);
33             }
34         } else {
35             JOptionPane.showMessageDialog( parentComponent: null, message: "Por Favor, ingrese los datos");
36         }
37     }
38 }
```

Version Control | TODO | Problems | Terminal | Services | Dependencies

Suggested plugin Protocol Buffers available for dependency 'java.com.google.protobuf:protobuf-java'. // Don't suggest this plugin (moments ago)

8:14 CRLF UTF-8 4 spaces

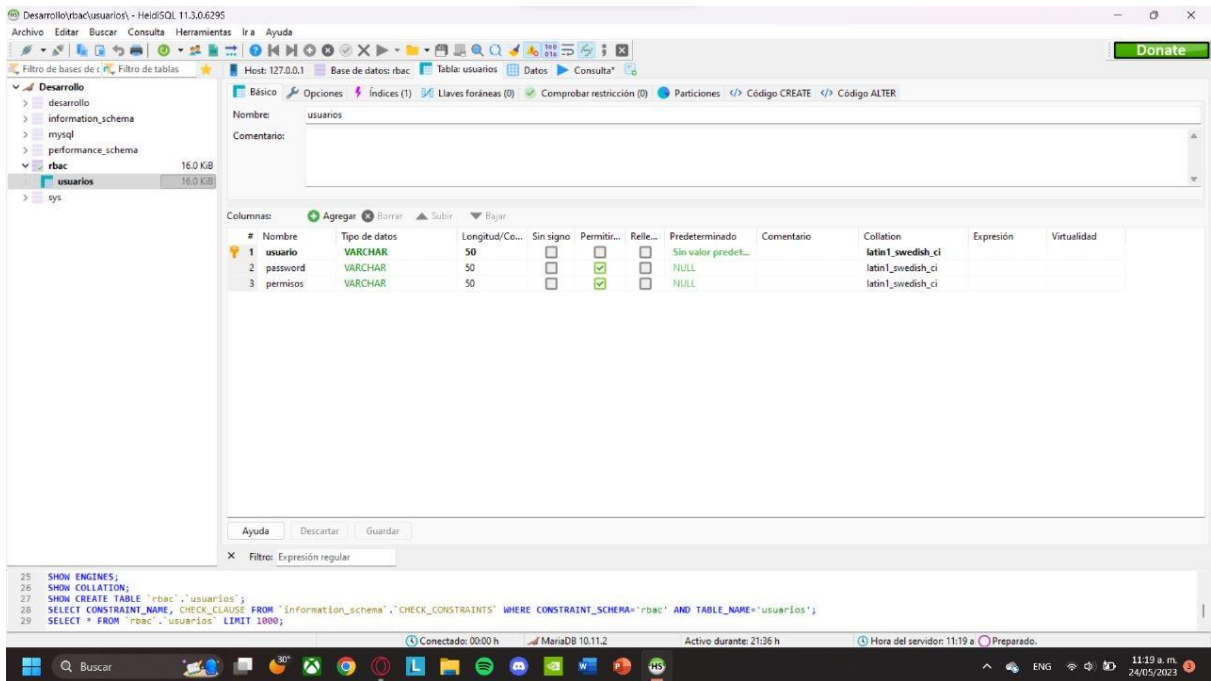
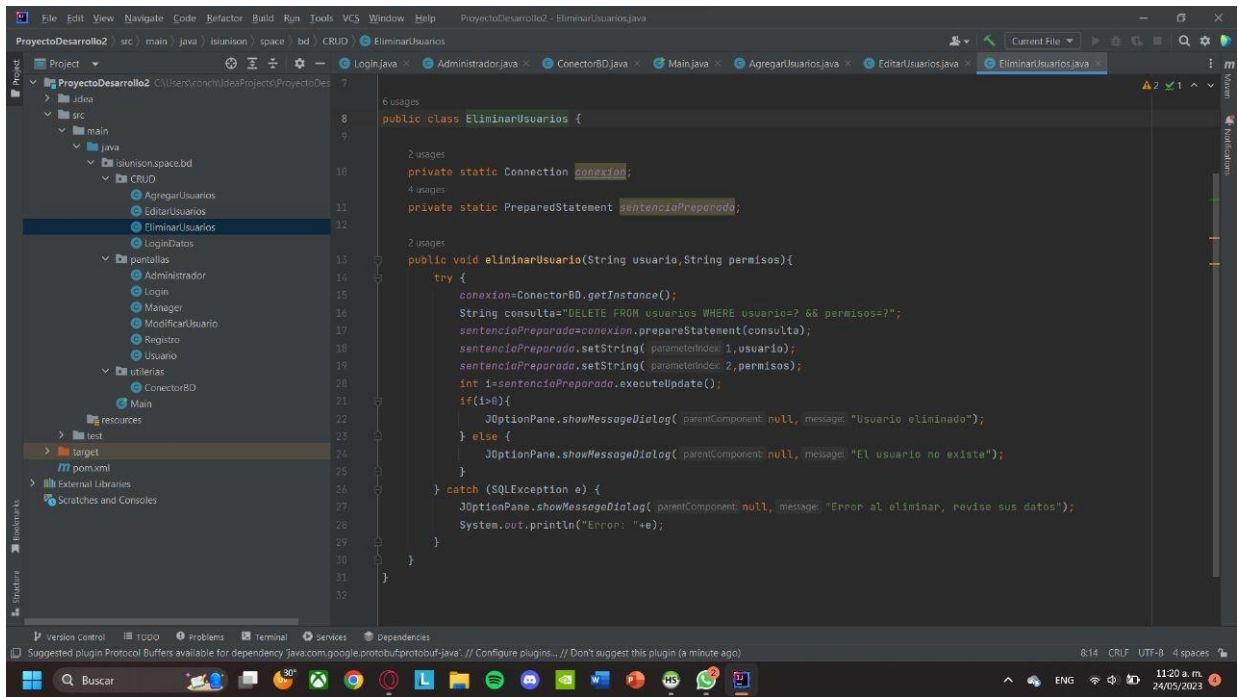


```
1 package isunison.space.bd;
2
3 import isunison.space.bd.pantallas.Login;
4
5 public class Main {
6     public static void main(String[] args) {
7         Login l=new Login();
8     }
9 }
10
```

Version Control | TODO | Problems | Terminal | Services | Dependencies

Suggested plugin Protocol Buffers available for dependency 'java.com.google.protobuf:protobuf-java'. // Configure plugins... // Don't suggest this plugin (4 minutes ago)

9:2 CRLF UTF-8 4 spaces



MANUAL

Este programa es una aplicación diseñada para asignar y controlar los derechos de acceso de los usuarios en función de su rol dentro de la organización. El sistema utiliza una base de datos MariaDB para almacenar los datos y garantizar un acceso seguro a la información.

Todo usuario de la organización tiene autorizado iniciar sesión o registrarse. Al iniciar el programa, los usuarios tienen la opción de iniciar sesión, en caso de que ya tengan una cuenta, o en otro caso pueden registrarse y asignar sus permisos.



The screenshot shows a window titled "Login" with a light purple header. The main content area is titled "Registrarse" in a large, bold, italicized font. Below the title, there are three input fields: "Usuario:" (User), "Contraseña:" (Password), and "Permisos:" (Permissions). The "Permisos:" field has the text "usuario" entered. Below the input fields, there are two buttons: "Volver" (Return) and "Registrarse" (Register).



The screenshot shows a window titled "Login" with a light purple header. The main content area is titled "Inicio de Sesión" in a large, bold, italicized font. Below the title, there are two input fields: "Usuario:" (User) and "Contraseña:" (Password). Below the input fields, there are two buttons: "Iniciar Sesión" (Login) and "Registrarse" (Register).

En caso de iniciar sesión como administrador, se abrirá la pantalla del administrador donde en la parte superior aparecen tres botones el primero de izquierda a derecha permite volver a la pantalla de login, el segundo es para modificar a cualquiera de los usuarios ya existentes y, finalmente, el botón para salir de la aplicación.

The screenshot shows a window titled 'Administrador' with standard Windows window controls (minimize, maximize, close). At the top, there are three buttons: 'Login', 'Modificar', and 'Salir'. The main area is divided into two columns. The left column is titled 'Nuevo Usuario' and contains three input fields: 'Inserte nuevo usuario', 'Inserte nueva contraseña', and 'Seleccione los permisos' (which is a dropdown menu). Below these fields is an 'Agregar' button. The right column is titled 'Eliminar Usuario' and contains three input fields: 'Inserte usuario a eliminar', 'Inserte la contraseña del administrador', and 'Inserte los permisos del usuario a eliminar' (which is a dropdown menu). Below these fields is an 'Eliminar' button.

Del lado izquierdo es posible agregar un usuario nuevo, su contraseña y sus permisos. Del lado derecho, se solicita que escribas el nombre del usuario, su contraseña y los permisos que poseía para poder eliminar a dicho usuario. Dependiendo de la acción tomada (añadir o borrar usuario) se selecciona entre las dos opiniones de los botones inferiores.

En caso de presionar el botón modificar, este abrirá la pantalla Modificar Usuario. En la parte superior, debajo del botón para volver, se encuentra el buscador de usuarios existentes, después de ingresar al usuario se podrá modificar su información, como su correo, la contraseña y la nueva selección de permisos.

Modificar Usuario

Volver

Inserte el usuario a modificar:

Buscar

Viejo Usuario	Nuevo Usuario
Viejo correo	Inserte nuevo usuario
<input type="text"/>	<input type="text"/>
Vieja contraseña	Inserte nueva contraseña
<input type="text"/>	<input type="text"/>
Viejos permisos	Seleccione los permisos
<input type="text"/>	<input type="text"/>

Ingrese sus datos

Usuario:

Password:

Guardar

Cuando se termine de modificar será posible ingresar los datos y guardar los cambios. Toda la información recabada y modificada será almacenada en una base de datos sql.