



Manual de usuario Stigma Sistema Gestor De Riesgos En El CDC

Historial de Versiones

Versión	Fecha	Detalle	Autor
0.9	14/12/2018	Comienzo	VMMR
1.0	16/12/2018	Termino	VALM

Contenido

Arquitectura del Sistema.....	6
I. Login	9
Conexión.....	10
Consultar Usuario	10
Consultar Correo electrónico	10
Consultar Contraseña.....	11
Consultar Tipo de Usuario	11
Entrar.....	11
II. Administrador	13
2.1 Pantalla Principal de Administrador.....	13
Código Fuente	13
Ventana de Gestión de Proyectos.....	13
Ventana de Gestión de Usuarios.....	14
Ventada de Matriz de Riesgos.....	14
Salida	14
2.2 Gestión de Proyectos	15
Botón de Menú-Proyectos	15
Botón de Menú-Usuarios	15
Botón de Menú-Matriz de Riesgos.....	16
Botón de Menú-Salir	16
2.2.1 Alta de Proyectos	16
Botón Agregar	17
Método Agregar Proyecto.....	17
2.2.2 Baja de Proyecto	18
Botón Eliminar.....	19
Método Eliminar Proyecto	19
2.2.3 Modificar Proyecto.....	20
Botón Modificar	21
Método Modificar Proyecto.....	21
2.2.4 Consulta de proyectos.....	22
Caja de Texto	22
Método de Consulta General	23

Método de Consulta Específica	23
2.3 Gestión de Usuarios	24
Botón de Menú-Proyectos	24
Botón de Menú-Usuarios	24
Botón de Menú-Matriz de Riesgos.....	25
Botón de Menú-Salir	25
2.3.1 Alta de Usuario	25
Botón Agregar Usuario	26
Método Agregar Usuario.....	26
2.3.2 Baja de Usuario	27
Botón de Eliminar	28
Método Eliminar Usuario	28
2.3.3 Modificación de Usuario	29
Botón Modificar	30
Método Modificar	30
2.3.4 Consulta de Usuario	31
Caja de Texto	31
Consulta General de Usuarios	32
Consulta Específica de Usuarios.....	32
2.4 Matriz de Riesgos	33
ComboBox de Proyecto	33
Consultar Matriz.....	33
III. Líder Técnico	34
3.1 Pantalla Principal de Líder Técnico.....	34
Botón de Menú-Gestión de Riesgos.....	34
Botón de Menú-Asignación de Riesgos.....	34
Botón de Menú-Acciones de Mitigación	34
Botón de Menú-Acciones de Contingencia	35
Botón de Menú-Matriz de Riesgos.....	35
Botón de Menú-Salir	35
3.2 Gestión de Riesgos	36
3.2.1 Alta de Riesgos	36
Botón Agregar Riesgo	37

Método Agregar Riesgo.....	38
3.2.2 Baja de Riesgos.....	39
Botón Eliminar Riesgo	40
Método de Eliminar Riesgo	40
3.2.3 Modificación de Riesgos.....	41
Botón de Modificar Riesgo.....	42
Método de Modificar Riesgo.....	42
3.2.4 Consulta de Riesgos.....	43
ComboBox de Proyectos	43
Método de Consulta General de Riesgos	44
3.3 Asignación de Riesgos	44
Botón de Asignar Riesgo.....	45
Método de Asignar Riesgo	45
3.4 Acciones de Mitigación	46
Botón de Agregar Acción de Mitigación.....	46
Método de Agregar Acción de Mitigación	47
3.5 Acciones de Contingencia	47
Botón de Agregar Acción de Contingencia.....	48
Método de Agregar Acción de Contingencia	48
3.6 Matriz de Riesgos	49
ComboBox de Proyectos	49
Método de Consulta.....	49
IV. Desarrollador.....	50
4.1 Página Principal de Desarrollador	50
Botón de Menú-Asignar-Inicio	50
Botón de Menú-Asignar-Probabilidad e Impacto	50
Botón de Menú-Seguimiento	50
Botón de Menú-Salir	51
4.2 Asignar Probabilidad e Impacto	51
Botón de Agregar Probabilidad e Impacto.....	52
Método de Agregar Probabilidad e Impacto.....	53
Método para obtener Promedio de la Probabilidad de un Riesgo	53
Método para obtener Promedio del Impacto de un Riesgo	54

Método para Agregar los Valores a la Tabla Principal 54

Arquitectura del Sistema

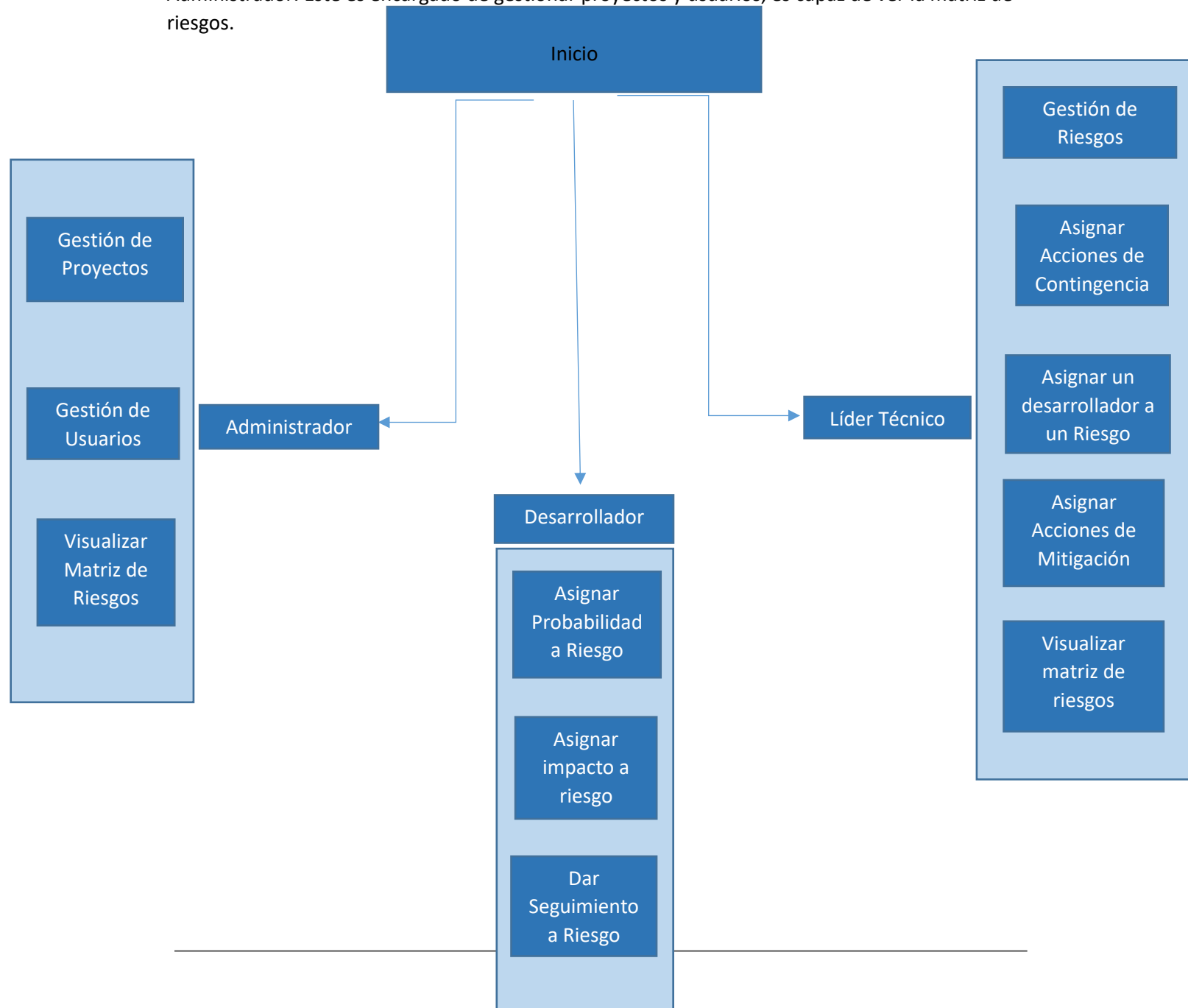
El sistema es creado con el fin de gestionar y asignar riesgos así como la probabilidad e impacto que tendrán, También tendrá usuarios que asignan acciones de mitigación y contingencia en caso de que llegaran a pasar.

Existen 3 tipos de usuarios

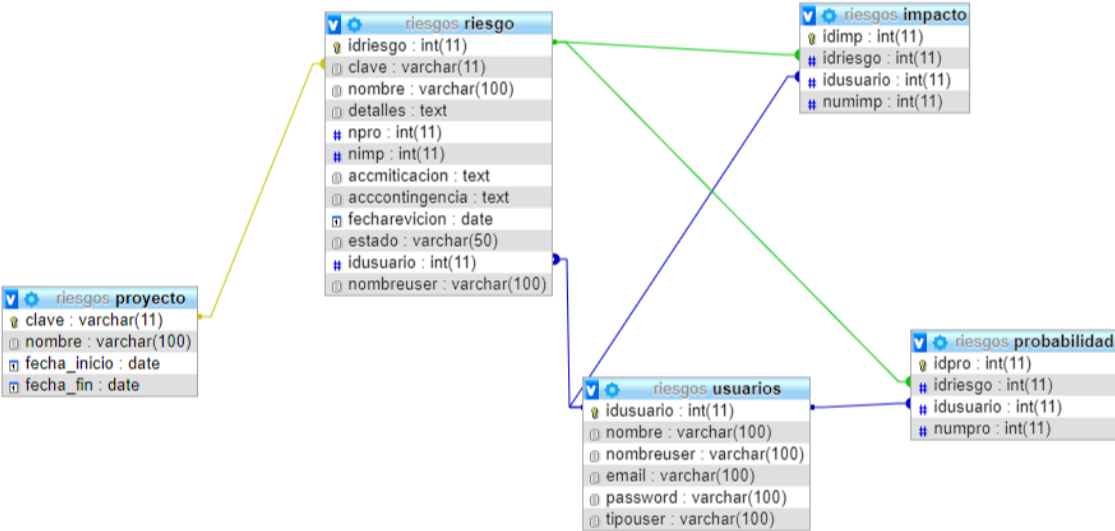
Desarrollador: Él es capaz de asignar probabilidad e impacto a todos los riesgos y dar seguimiento a riesgos asignados.

Líder Técnico: Él es encargado de gestionar riesgos y asignar un desarrollador, asignar acciones de mitigación y contingencia a un riesgo, es capaz de ver la matriz de riesgos.

Administrador: Este es encargado de gestionar proyectos y usuarios, es capaz de ver la matriz de riesgos.



Paquete	Componente	Descripción
Login	Login	Verifica nombre de usuario o correo electrónico y contraseña, dependiendo de qué tipo de usuario sea tendrán diferentes accesos al programa
Administrador	Gestión de Proyectos	Permite la consulta, modificación, eliminación y creación proyectos
	Gestión de Usuarios	Permite la consulta, modificación, eliminación y creación Usuarios
	Matriz de Riesgos	Permite Visualizar los diferentes riesgos de cada proyecto con su probabilidad, impacto, exposición, acciones de mitigación, acciones contingencia y el usuario a cargo
Líder Técnico	Gestión de Riesgos	Permite la consulta, modificación, eliminación y creación de Riesgos
	Asignar Acciones de Mitigación	Asigna acciones de mitigación a un riesgo
	Asignar Acciones de Contingencia	Asigna acciones de contingencia a un riesgo
	Asignar Riesgos	Asigna un usuario a un riesgo para darle seguimiento
	Matriz de Riesgos	Permite Visualizar los diferentes riesgos de cada proyecto con su probabilidad, impacto, exposición, acciones de mitigación, acciones contingencia y el usuario a cargo
Desarrollador	Asignar Probabilidad e Impacto	Asignar la probabilidad de que ocurra e impacto que tendrá en caso de ocurrir
	Seguimiento	Un usuario da seguimiento en diferentes aspectos a un riesgo



I. Login

En Formulario se autentifica el usuario y se envía a distintos formularios, dependiendo del tipo de usuario (Administrador, Líder Técnico, Desarrollador).

The image shows a login form with a purple gradient background. At the top center is a white outline icon of two people. Below this are two black input fields. The first field has a small white user icon on its left. The second field has a small white padlock icon on its left. At the bottom of the form are two white buttons with black text: 'Aceptar' on the left and 'Salir' on the right.

Conexión

```

public boolean Conectar() {
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        Conexion = DriverManager.getConnection(
            "jdbc:mysql://localhost/riesgos", "root", "");
        if (Conexion != null) {
            return true;
        } else {
            return false;
        }
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

public void Desconectar() {
    try {
        this.Conexion.close();
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
}

```

Consultar Usuario

```

public String ConsultaUser(String User) {
    String Usuario = "Incorrecto";
    Statement consulta;
    ResultSet resultado;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select nombreuser from usuarios where nombreuser = '" + User + "' or email = '" + User + "';");
        while (resultado.next()) {
            Usuario = resultado.getString("nombreuser");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "No Existe");
        e.printStackTrace();
    }

    return Usuario;
}

```

Consultar Correo electrónico

```

public String ConsultaCorreo(String User) {
    String Email = "";
    Statement consulta;
    ResultSet resultado;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select email from usuarios where nombreuser = '" + User + "' or email = '" + User + "';");
        while (resultado.next()) {
            Email = resultado.getString("email");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "No Existe");
        e.printStackTrace();
    }

    return Email;
}

```

Consultar Contraseña

```

public String ConsultaPassword(String User) {
    String Pass = "";
    Statement consulta;
    ResultSet resultado;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select password from usuarios where nombreuser = '" + User + "' or email = '" + User + "';");
        while (resultado.next()) {
            Pass = resultado.getString("password");
        }
        //JOptionPane.showMessageDialog(null, "Existe");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Contraseña Incorrecta");
        e.printStackTrace();
    }

    return Pass;
}

```

Consultar Tipo de Usuario

```

public String ConsultaTipo(String User) {
    String Tipo = "";
    Statement consulta;
    ResultSet resultado;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select tipouser from usuarios where nombreuser = '" + User + "' or email = '" + User + "';");
        while (resultado.next()) {
            Tipo = resultado.getString("tipouser");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "No Existe");
        e.printStackTrace();
    }

    return Tipo;
}

```

Entrar

```

private void BtnLoginMouseClicked(java.awt.event.MouseEvent evt) {
    if (TxtUser.equals("") || TxtPass.equals("")) {
        JOptionPane.showMessageDialog(null, "Ingrese Nombre de usuario o Correo Electronico y Contraseña");
    } else {
        if (mBD.Conectar()) {
            String User, Pass, UserC, PassC, Tipo, Email = "";

            User = TxtUser.getText();
            Pass = TxtPass.getText();
            UserC = mBD.ConsultaUser(User);
            PassC = mBD.ConsultaPassword(User);
            Email = mBD.ConsultaCorreo(User);

            if (User.equals(UserC) || User.equals(Email) ) {
                if (Pass.equals(PassC) ) {
                    Tipo = mBD.ConsultaTipo(User);

                    switch (Tipo){
                        case "Administrador":
                            PrincipalAdmin Admin = new PrincipalAdmin();
                            Admin.setVisible(true);
                            this.dispose();
                            break;
                        case "Lider Tecnico":
                            PrincipalLider Lider = new PrincipalLider();
                            Lider.setVisible(true);
                            this.dispose();
                            break;
                    }
                }
            }
        }
    }
}

```

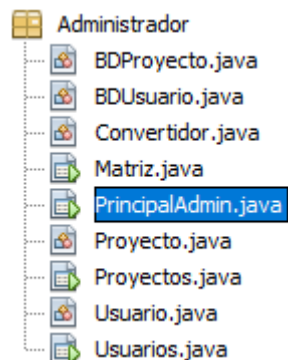
```
        case "Desarrollador":
            Usuario.User = "";
            Usuario.User = mBD.ConsultaUser(User);
            PrincipalDesarrollador Des = new PrincipalDesarrollador();
            Des.setVisible(true);
            this.dispose();
            break;
    }
    } else {
        JOptionPane.showMessageDialog(null, "Contraseña Invalida");
    }
    } else {
        JOptionPane.showMessageDialog(null, "Usuario o Correo Invalidos");
    }
    } else {
        JOptionPane.showMessageDialog(null, "Error");
    }
    }
    mBD.Desconectar();
}
}
```

II. Administrador

2.1 Pantalla Principal de Administrador



Código Fuente



Ventana de Gestión de Proyectos

```
private void jMenuItemMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Proyectos mProyectos = new Proyectos();  
    this.dispose();  
    mProyectos.setVisible(true);  
}
```

Ventana de Gestión de Usuarios

```
private void jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) {  
    Usuarios mUsuarios = new Usuarios();  
    mUsuarios.setVisible(true);  
    this.dispose();  
}
```

Ventana de Matriz de Riesgos

```
private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {  
    Matriz mMatriz = new Matriz();  
    mMatriz.setVisible(true);  
    this.dispose();  
}
```

Salida

```
private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {  
    Login mLogin = new Login();  
    mLogin.setVisible(true);  
    this.dispose();  
}
```

2.2 Gestión de Proyectos

The screenshot shows a Java Swing window titled "Gestión de Proyectos". It features a standard window control bar (minimize, maximize, close). Below the title bar is a menu bar with items: "Inicio", "Proyectos", "Usuarios", "Matriz De Riesgos", and "Salir". Underneath the menu bar is a toolbar with four buttons: "Alta", "Baja", "Consulta", and "Modificar". The main content area is a purple-themed form with the following elements:

- Clave:** A single-line text input field.
- Nombre:** A single-line text input field.
- Descripcion:** A large multi-line text area.
- Fecha de Inicio:** A date input field with a calendar icon button to its right.
- Fecha Fin:** A date input field with a calendar icon button to its right.
- Buttons:** "Agregar" (bottom left) and "Cancelar" (bottom right).

Botón de Menú-Proyectos

```
private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Prpyectos mProyectos = new Proyectos();  
    this.dispose();  
    mProyectos.setVisible(true);  
}
```

Botón de Menú-Usuarios

```
private void jMenu2MouseClicked(java.awt.event.MouseEvent evt) {  
    Usuarios mUsuarios = new Usuarios();  
    mUsuarios.setVisible(true);  
    this.dispose();  
}
```

Botón de Menú-Matriz de Riesgos

```
private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {  
    Matriz mMatriz = new Matriz();  
    mMatriz.setVisible(true);  
    this.dispose();  
}
```

Botón de Menú-Salir

```
private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {  
    Login mLogin = new Login();  
    mLogin.setVisible(true);  
    this.dispose();  
}
```

2.2.1 Alta de Proyectos

Gestión de Proyectos

Inicio Proyectos Usuarios Matriz De Riesgos Salir

Alta Baja Consulta Modificar

Clave:

Nombre:

Descripcion:

Fecha de Inicio:

Fecha Fin:

Agregar Cancelar

Botón Agregar

```

private void BtnAddMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los elementos necesarios no esten vacios
    if (TxtClave.equals("") || TxtNom.equals("") || TxtDes.equals("") || Inicio.getDate() == null || Fin.getDate() == null) {
        JOptionPane.showMessageDialog(null, "Llena los campos antes de guardar");
    } else {
        if (mBD.ConectarProyecto()) {
            Proyecto mProyecto = new Proyecto();
            mProyecto.setClave(TxtClave.getText());
            mProyecto.setNombre(TxtNom.getText());
            mProyecto.setDescripcion(TxtDes.getText());
            mProyecto.setFechaInicio(formatofecha.format(Inicio.getDate()));
            mProyecto.setFechaFin(formatofecha.format(Fin.getDate()));
            //Da de alta el proyecto
            if (mBD.AgregarProyecto(mProyecto)) {
                JOptionPane.showMessageDialog(null, "Proyecto Agregado con Exito");
                Borrar();
                Llenar();
                TxtClave.setText("");
                TxtNom.setText("");
                TxtDes.setText("");
                Inicio.setDate(null);
                Fin.setDate(null);
            } else {
                JOptionPane.showMessageDialog(null, "Error en BD");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
        mBD.DesconectarProyecto();
    }
}

```

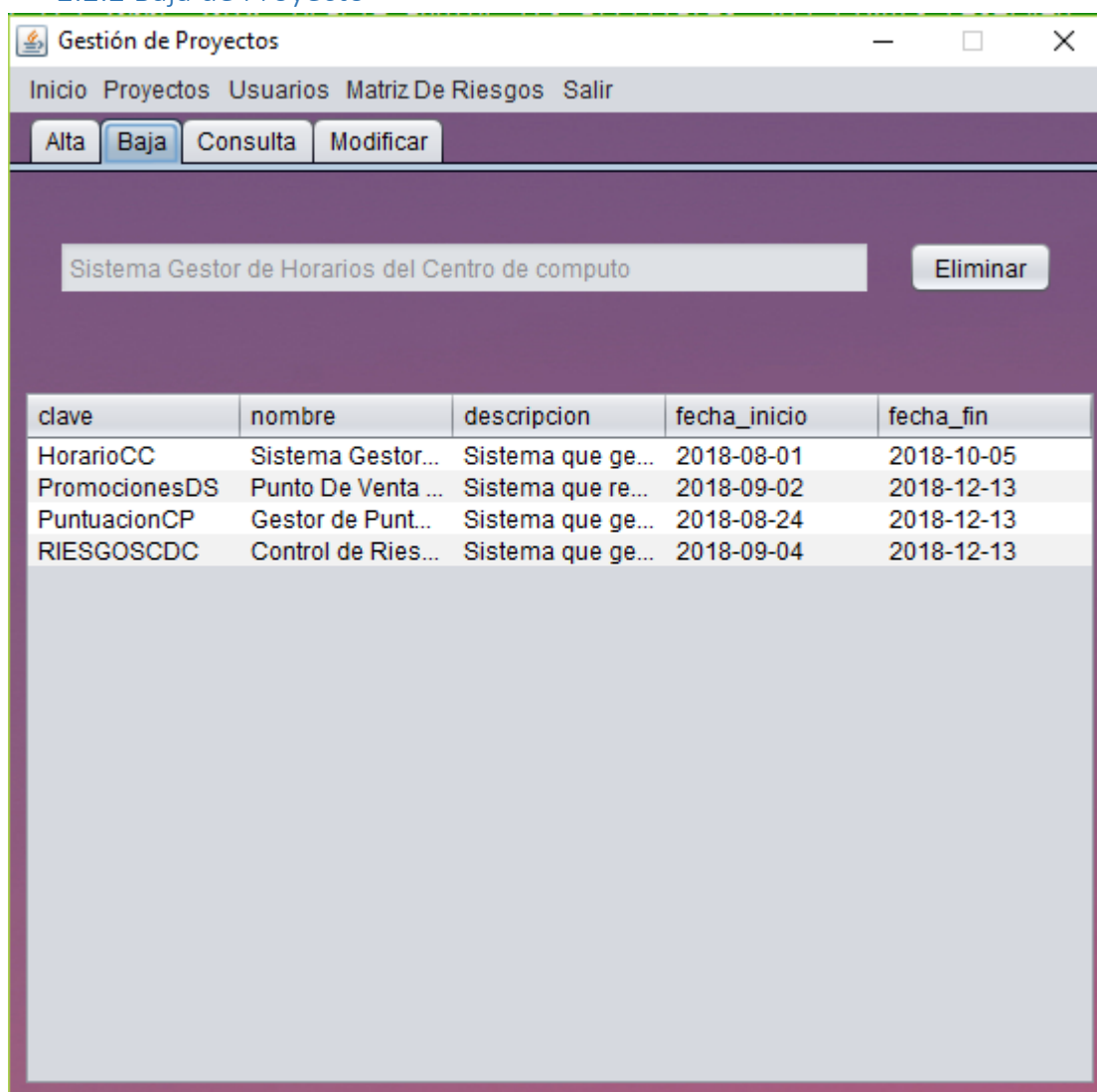
Método Agregar Proyecto

```

public boolean AgregarProyecto(Proyecto mProyecto) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "INSERT INTO riesgos.proyecto "
            + "(clave,nombre,descripcion,fecha_inicio,fecha_fin) "
            + "VALUES ('"
            + mProyecto.getClave() + "','"
            + mProyecto.getNombre() + "','"
            + mProyecto.getDescripcion() + "','"
            + mProyecto.getFechaInicio() + "','"
            + mProyecto.getFechaFin() + "')";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

2.2.2 Baja de Proyecto



The screenshot shows a web application window titled "Gestión de Proyectos". The navigation bar includes "Inicio", "Proyectos", "Usuarios", "Matriz De Riesgos", and "Salir". Below the navigation bar are four buttons: "Alta", "Baja" (highlighted), "Consulta", and "Modificar".

Below the buttons is a search bar containing the text "Sistema Gestor de Horarios del Centro de computo" and an "Eliminar" button.

Below the search bar is a table with the following data:

clave	nombre	descripcion	fecha_inicio	fecha_fin
HorarioCC	Sistema Gestor...	Sistema que ge...	2018-08-01	2018-10-05
PromocionesDS	Punto De Venta ...	Sistema que re...	2018-09-02	2018-12-13
PuntuacionCP	Gestor de Punt...	Sistema que ge...	2018-08-24	2018-12-13
RIESGOSCDC	Control de Ries...	Sistema que ge...	2018-09-04	2018-12-13

Botón Eliminar

```
private void BtnEliminarMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que a Clave no este vacia
    if ( Clave == "" ) {
        JOptionPane.showMessageDialog(null, "Seleccine un proyecto");
    } else {
        if (mBD.ConectarProyecto()) {
            //Elimina el proyecto seleccionado
            if (mBD.EliminarProyecto(Clave)) {
                JOptionPane.showMessageDialog(null, "Proyecto Dado de Baja");
                TxtEliminar.setText("");
                Borrar();
                Llenar();
            } else {
                JOptionPane.showMessageDialog(null, "Error al dar de Baja");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en a BD");
        }
    }
}
}
```

Método Eliminar Proyecto

```
public boolean EliminarProyecto(String Clave) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        consulta.execute("delete from proyecto"
            + " where clave = '" + Clave + "';");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}
```

2.2.3 Modificar Proyecto

The screenshot shows a web application window titled "Gestión de Proyectos". The window has a menu bar with "Inicio", "Proyectos", "Usuarios", "Matriz De Riesgos", and "Salir". Below the menu bar is a toolbar with buttons for "Alta", "Baja", "Consulta", and "Modificar". The "Modificar" button is highlighted. The main content area has a purple background and contains two input fields: "Nombre:" and "Descripcion:". Below these fields is a table with five columns: "clave", "nombre", "descripcion", "fecha_inicio", and "fecha_fin". The table contains four rows of data. At the bottom of the window are two buttons: "Modificar" and "Cancelar".

clave	nombre	descripcion	fecha_inicio	fecha_fin
HorarioCC	Sistema Gest...	Sistema que ...	2018-08-01	2018-10-05
Promociones...	Punto De Ven...	Sistema que ...	2018-09-02	2018-12-13
PuntuacionCP	Gestor de Pu...	Sistema que ...	2018-08-24	2018-12-13
RIESGOSCDC	Control de Ri...	Sistema que ...	2018-09-04	2018-12-13

Botón Modificar

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los elementos necesarios no esten vacios
    if (TxtNom1.getText() == "" || TxtDes1.getText() == "" || ClaveM == "") {
        JOptionPane.showMessageDialog(null, "Seleccione un Proyecto y llene los campos a modificar");
    } else {
        Proyecto mP = new Proyecto();
        mP.setClave(ClaveM);
        mP.setNombre(TxtNom1.getText());
        mP.setDescripcion(TxtDes1.getText());

        if (mBD.ConectarProyecto()) {
            //Modifica el Proyecto seleccionado
            if (mBD.ModificarProyecto(mP)) {
                JOptionPane.showMessageDialog(null, "Modificacion realizada con exito");
                TxtNom1.setText("");
                TxtDes1.setText("");
                Borrar();
                this.Llenar();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
        mBD.DesconectarProyecto();
    }
}

```

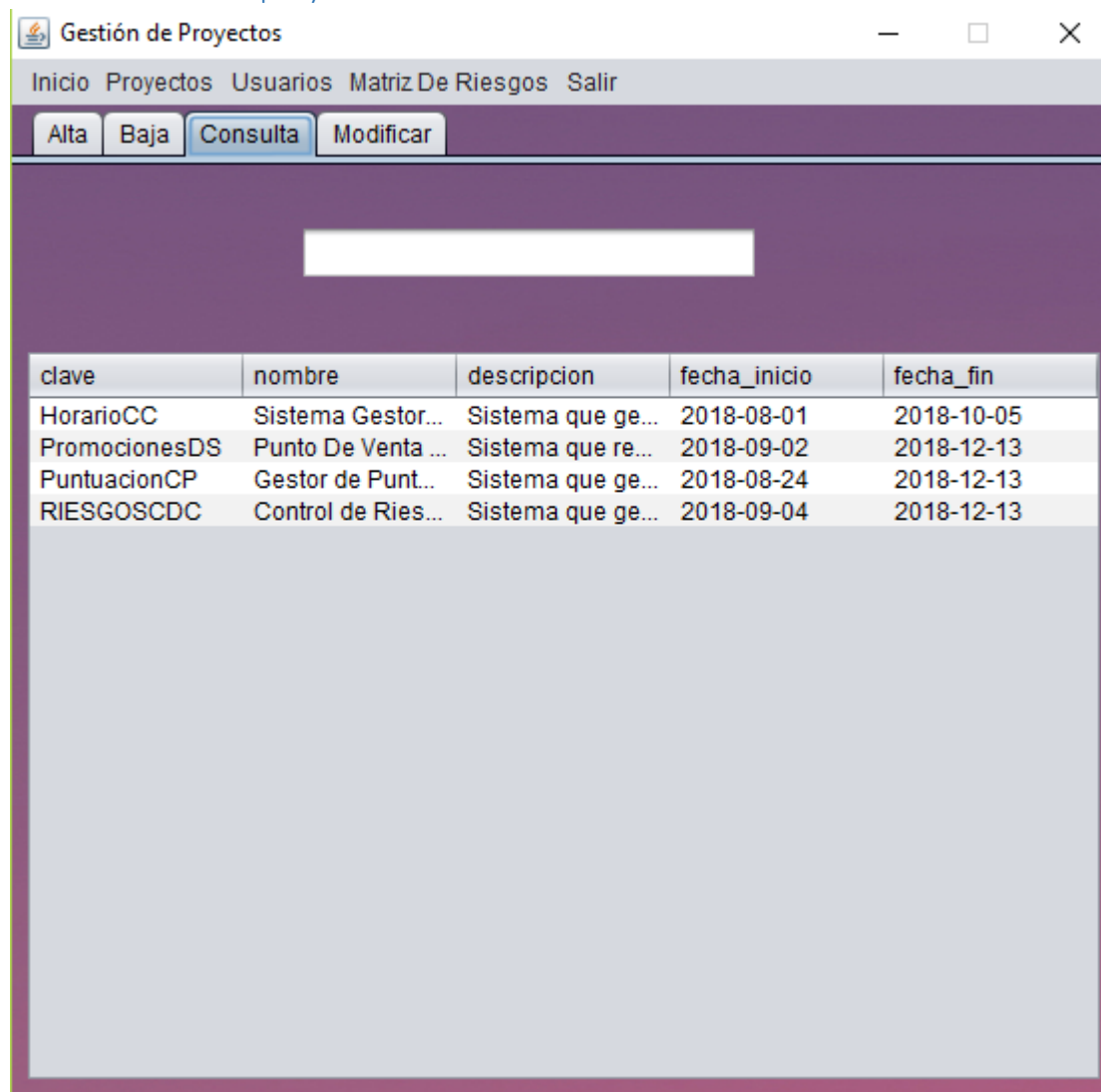
Método Modificar Proyecto

```

public boolean ModificarProyecto(Proyecto mP) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "UPDATE riesgos.proyecto SET "
            + "nombre = '" + mP.getNombre() + "',"
            + "descripcion = '" + mP.getDescripcion() + "'"
            + " where clave = '" + mP.getClave() + "';";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

2.2.4 Consulta de proyectos



Caja de Texto

```
private void TxtCriterioKeyTyped(java.awt.event.KeyEvent evt) {
    // Verifica que en caso de que el criterio este vacio, se hafa una consulta general de los proyectos
    if (this.TxtCriterio.equals("")) {
        Borrar();
        Llenar();
    }
    //Borra el contenido de la tabla y la llena de acuerdo con el criterio de busqueda
    BorrarC();
    if (this.mBD.ConectarProyecto()) {
        ResultSet Lista = null;
        Lista = mBD.ConsultaFiltroP(TxtCriterio.getText());
        this.TBProyecto3.setModel(Convertidor.convertir(Lista));
    } else {
        JOptionPane.showMessageDialog(null, "Error en BD");
    }
}
}
```

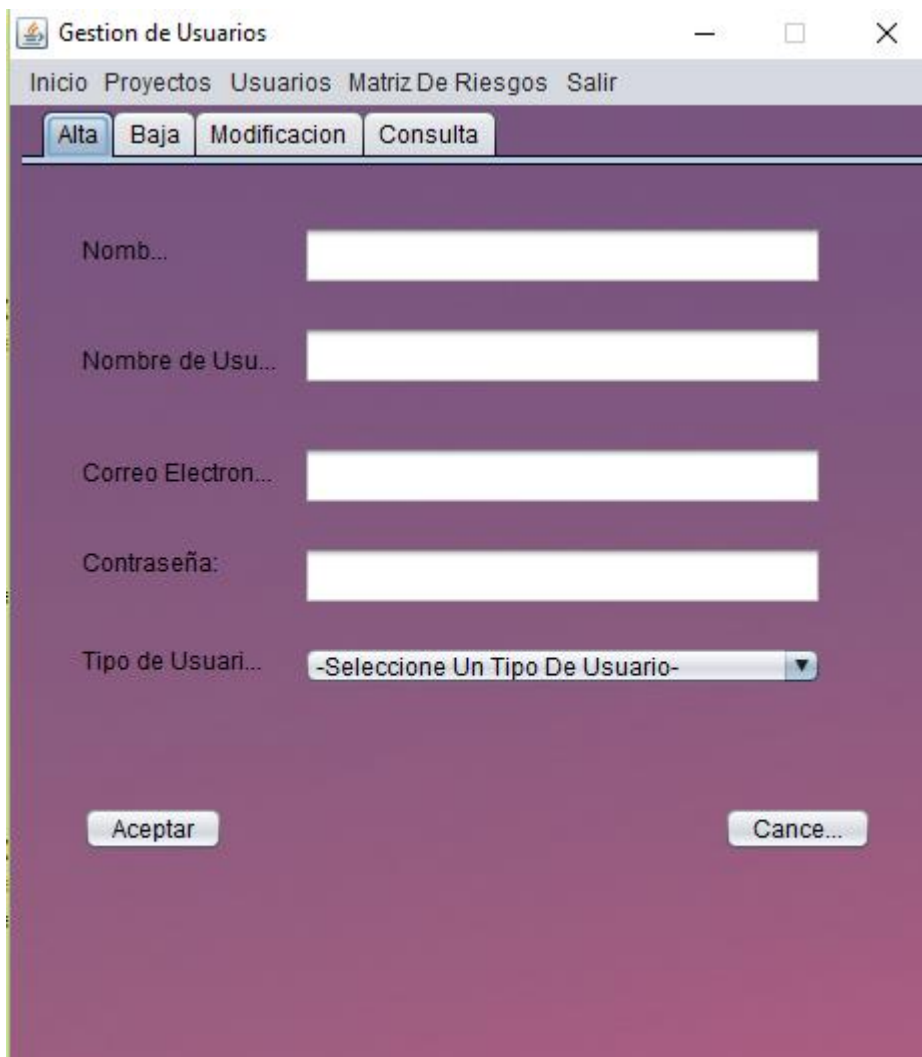
Método de Consulta General

```
public ResultSet ConsultaProyectos() {  
    ArrayList mLista = new ArrayList();  
    Proyecto mProyecto = new Proyecto();  
    Statement consulta;  
    ResultSet resultado = null;  
  
    try {  
        consulta = Conexion.createStatement();  
        resultado = consulta.executeQuery("select * from proyecto;");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return resultado;  
}
```

Método de Consulta Específica

```
public ResultSet ConsultaFiltroP(String Criterio) {  
    ArrayList mLista = new ArrayList();  
    Proyecto mProyecto = new Proyecto();  
    Statement consulta;  
    ResultSet resultado = null;  
  
    try {  
        consulta = Conexion.createStatement();  
        resultado = consulta.executeQuery("select * from Proyecto where clave like '%" + Criterio + "%' "  
            + "or nombre like '%" + Criterio + "%';");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return resultado;  
}
```

2.3 Gestión de Usuarios



Botón de Menú-Proyectos

```
private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Proyectos mProyectos = new Proyectos();
    this.dispose();
    mProyectos.setVisible(true);
}
```

Botón de Menú-Usuarios

```
private void jMenu2MouseClicked(java.awt.event.MouseEvent evt) {
    Usuarios mUsuarios = new Usuarios();
    mUsuarios.setVisible(true);
    this.dispose();
}
```


Botón de Menú-Matriz de Riesgos

```
private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {  
    Matriz mMatriz = new Matriz();  
    mMatriz.setVisible(true);  
    this.dispose();  
}
```

Botón de Menú-Salir

```
private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {  
    Login mLogin = new Login();  
    mLogin.setVisible(true);  
    this.dispose();  
}
```

2.3.1 Alta de Usuario

The screenshot shows a Java Swing window titled "Gestion de Usuarios". The window has a menu bar with "Inicio", "Proyectos", "Usuarios", "Matriz De Riesgos", and "Salir". Below the menu bar is a tabbed interface with four tabs: "Alta", "Baja", "Modificacion", and "Consulta". The "Alta" tab is currently selected. The form contains the following fields and controls:

- "Nomb...": A text input field.
- "Nombre de Usu...": A text input field.
- "Correo Electron...": A text input field.
- "Contraseña:": A text input field.
- "Tipo de Usuari...": A dropdown menu with the text "-Seleccione Un Tipo De Usuario-".
- "Aceptar": A button at the bottom left.
- "Cance...": A button at the bottom right.

Botón Agregar Usuario

```

private void BtnAddMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los elementos necesarios no esten vacios
    if ((this.TxtNom.equals(null)) || (this.TxtNomUser.equals(null)) || (this.TxtEmail.equals(null))
        || (TxtPass.equals(null)) || CBTipo.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(null, "Llene todos los campos antes de guardar");
    } else {
        Usuario mUsuario = new Usuario();
        mUsuario.setNombre(TxtNom.getText());
        mUsuario.setNomUser(TxtNomUser.getText());
        mUsuario.setEmail(TxtEmail.getText());
        mUsuario.setPassword(TxtPass.getText());
        mUsuario.setTipoUser(CBTipo.getSelectedItem().toString());
        //Da de alta el usuario
        if (mBD.ConectarUsuario()) {
            if (mBD.AgregarUsuario(mUsuario)) {
                JOptionPane.showMessageDialog(null, "Usuario Dado de alta de forma exitosa");
                TxtNom.setText(null);
                TxtNomUser.setText(null);
                TxtEmail.setText(null);
                TxtPass.setText(null);
                CBTipo.setSelectedIndex(0);
                Borrar();
                setFilas();
            } else {
                JOptionPane.showMessageDialog(null, "Error al dar de Alta");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error");
        }
        mBD.DesconectarUsuario();
    }
}

```

Método Agregar Usuario

```

public boolean AgregarUsuario(Usuario mUsuario) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "INSERT INTO riesgos.usuarios "
            + "(idusuario,nombre,nombreuser,email,password,tipouser) "
            + "VALUES ("
            + "null" + "," + "
            + mUsuario.getNombre() + "," + "
            + mUsuario.getNomUser() + "," + "
            + mUsuario.getEmail() + "," + "
            + mUsuario.getPassword() + "," + "
            + mUsuario.getTipoUser() + ")";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

2.3.2 Baja de Usuario

Gestion de Usuarios

Inicio Proyectos Usuarios Matriz De Riesgos Salir

Alta **Baja** Modificacion Consulta

Eliminar

idusuario	nombre	nombreuser	email	tipouser
1	Luis Moisé...	AdminVALM	l_vega@gm...	Administra...
2	Aaron Sern...	SJA31	SJA1211@...	Lider Tecnico
4	Omar Almar...	CAO21	cordovaom...	Desarrollad...
5	Daniel Arre...	Admin	daniels@ci...	Administra...
6	Manuel Ra...	Curva89	Curva465@...	Desarrollad...
7	Manuel Ign...	AdminISC	manuelIS@...	Administra...
8	Abraham M...	Lider10	A_lider_10...	Lider Tecnico
12	UsuarioPru...	UsPrueba3	prueba2@e...	Administra...
13	Giselle Mire...	gis21	gis21@gm...	Desarrollad...
14	Eneida Mire...	NM21	n@gmail.co...	Desarrollad...

Botón de Eliminar

```

private void BtnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    //Verifica que el ID del usuario no este vacio
    if (this.TxtEliminar.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Seleccione un usuario antes");
    } else {
        if (mBD.ConectarUsuario()) {
            //Elimina al usuario seleccionado
            if (mBD.EliminarUsuario(ID)) {
                JOptionPane.showMessageDialog(null, "Usuario dado de baja");
                TxtEliminar.setText("");
                Borrar();
                this.setFilas();
            } else {
                JOptionPane.showMessageDialog(null, "Error al Eliminar");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error");
        }
        mBD.DesconectarUsuario();
    }
}

```

Método Eliminar Usuario

```

public boolean EliminarUsuario(int ID) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        consulta.execute("delete from usuarios"
            + " where idusuario = " + ID + ";");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

2.3.3 Modificación de Usuario

Gestion de Usuarios

Inicio Proyectos Usuarios Matriz De Riesgos Salir

Alta Baja **Modificacion** Consulta

ID:

Nombre Usuario:

Nueva Contraseña:

idusuario	nombre	nombreuser	email	tipouser
1	Luis Moisé...	AdminVALM	L_vega@g...	Administra...
2	Aaron Sern...	SJA31	SJA1211@...	Lider Tecni...
4	Omar Alma...	CAO21	cordovaom...	Desarrolla...
5	Daniel Arre...	Admin	daniels@ci...	Administra...
6	Manuel Ra...	Curva89	Curva465...	Desarrolla...
7	Manuel Ign...	AdminISC	manuelIS...	Administra...
8	Abraham M...	Lider10	A_lider_10...	Lider Tecni...
12	UsuarioPru...	UsPrueba3	prueba2@...	Administra...
13	Giselle Mir...	gis21	gis21@gm...	Desarrolla...
14	Enaida Mir...	NM21	n@amail...	Desarrolla...

Modificar Cancelar

Botón Modificar

```

private void BtnModMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los elementos necesarios no esten vacios
    if (this.TxtNomUser1.getText() == "" || this.TxtPass1.getText() == "" || this.TxtID.getText() == "") {
        JOptionPane.showMessageDialog(null, "Seleccione un usuario y llene los campos");
    } else {
        if (mBD.ConectarUsuario()) {
            Usuario mUsuario = new Usuario();
            Usuario nUsuario = new Usuario();
            String Tl = "";

            mUsuario.setID(Integer.parseInt(TxtID.getText()));
            nUsuario.setNomUser(TxtNomUser1.getText());
            nUsuario.setPassword(TxtPass1.getText());
            //Modifica el usuario seleccionado
            if (mBD.ModificarUsuario(mUsuario, nUsuario)) {
                JOptionPane.showMessageDialog(null, "Usuario Modificado con Exito");
                TxtID.setText("");
                TxtNomUser1.setText("");
                TxtPass1.setText("");
                Borrar();
                setFilas();
            } else {
                JOptionPane.showMessageDialog(null, "Error en la BD");
            }
            mBD.DesconectarUsuario();
        }
    }
}

```

Método Modificar

```

public boolean ModificarUsuario(Usuario mUsuario, Usuario nUsuario){
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "UPDATE riesgos.usuarios SET "
            + "nombreuser = '" + nUsuario.getNomUser() + "',"
            + "password = '" + nUsuario.getPassword() + "'"
            + " where idusuario = '" + mUsuario.getID() + "';";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

2.3.4 Consulta de Usuario



Caja de Texto

```
private void TxtCriterioKeyTyped(java.awt.event.KeyEvent evt) {
    // Verifica que en caso de que el criterio de busqueda este vacio presente todos los usuarios
    if (TxtCriterio.equals("")) {
        Borrar();
        setFilas();
        JOptionPane.showMessageDialog(null, "Indique un criterio para realizar busqueda");
    } else {
        //Borra los elementos de la tabla y vuelve a llenarla dependiendo del criterio escrito en a caja de texto
        BorrarC();
        if (mBD.ConectarUsuario()) {
            ResultSet Lista = null;
            Lista = mBD.ConsultaFiltro(TxtCriterio.getText());

            this.TBUsuarios3.setModel(Convertidor.convertir(Lista));
        } else {
            JOptionPane.showMessageDialog(null, "Error en la Base de Datos");
        }
        mBD.DesconectarUsuario();
    }
}
```

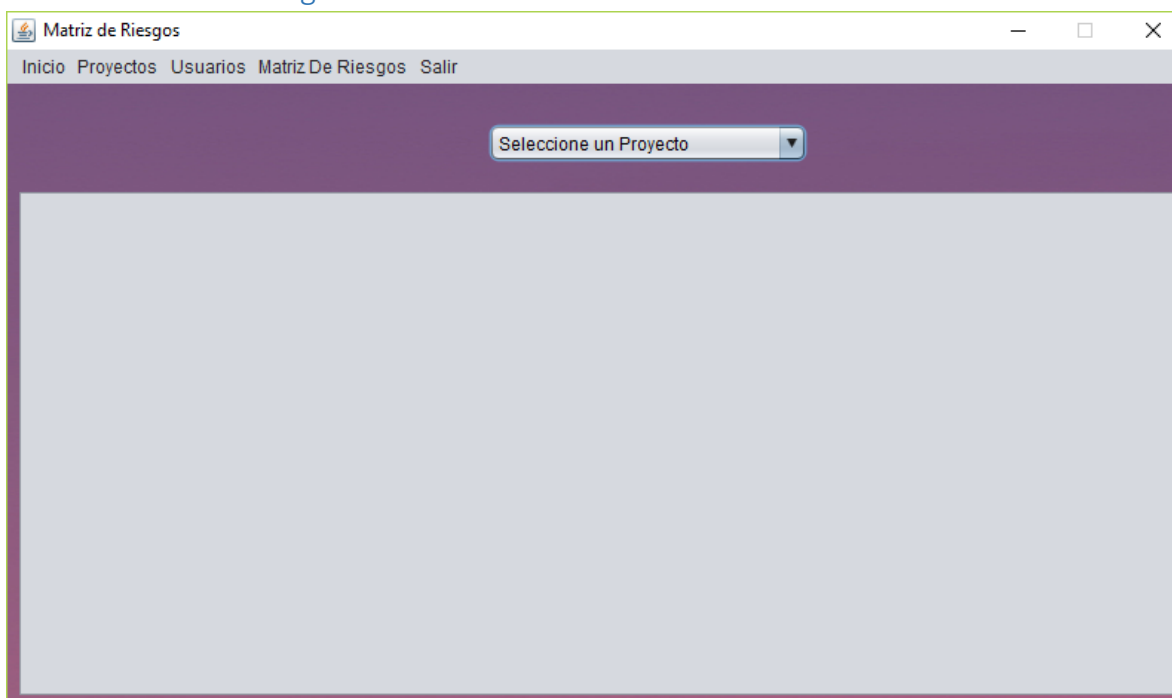
Consulta General de Usuarios

```
public ResultSet Consulta() {  
    ArrayList mLista = new ArrayList();  
    Usuario mUsuario = new Usuario();  
    Statement consulta;  
    ResultSet resultado = null;  
  
    try {  
        consulta = Conexion.createStatement();  
        resultado = consulta.executeQuery("select idusuario, nombre, nombreuser, email, tipouser from usuarios;");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return resultado;  
}
```

Consulta Específica de Usuarios

```
public ResultSet ConsultaFiltro(String Criterio) {  
    Usuario mUsuario = null;  
    Statement consulta;  
    ResultSet resultado = null;  
    ArrayList Lista = new ArrayList();  
  
    try {  
        mUsuario = new Usuario();  
        consulta = Conexion.createStatement();  
        resultado = consulta.executeQuery("select idusuario, nombre, nombreuser, email, tipouser"  
            + " from usuarios where idusuario like '%" + Criterio + "%' "  
            + "or nombre like '%" + Criterio + "%' or nombreuser like '%" + Criterio +  
            "%' or email like '%" + Criterio + "%';");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return resultado;  
}
```


2.4 Matriz de Riesgos



ComboBox de Proyecto

```
private void CBProyectoItemStateChanged(java.awt.event.ItemEvent evt) {
    // Llena la tabla dependiendo del elemento seleccionado
    if (listo) {
        if (mBD.ConectarProyecto()) {
            String C = "";
            C = CBProyecto.getSelectedItem().toString();
            ResultSet Lista = mBD.Matriz(C);
            this.TBProyecto.setModel(Desarrollador.Convertidor.convertir(Lista));
        }
        mBD.DesconectarProyecto();
    }
    //sout tabular
}
```

Consultar Matriz

```
public ResultSet Matriz(String C) {
    ArrayList mLista = new ArrayList();
    Proyecto mProyecto = new Proyecto();
    Statement consulta;
    ResultSet resultado = null;

    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select idriesgo, clave, nombre, npro, nimp, "
            + "exposicion, accmitigacion, acccontingencia, fecharevision, estado, nombreuser from riesgo"
            + " where clave='" + C + "';");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return resultado;
}
```

III. Líder Técnico

3.1 Pantalla Principal de Líder Técnico



Botón de Menú-Gestión de Riesgos

```
private void jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Riesgos mR = new Riesgos();
    mR.setVisible(true);
    this.dispose();
}
```

Botón de Menú-Asignación de Riesgos

```
private void jMenuItem9MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    AsignarRiesgo mAR = new AsignarRiesgo();
    mAR.setVisible(true);
    this.dispose();
}
```

Botón de Menú-Acciones de Mitigación

```
private void jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Mitigacion mM = new Mitigacion();
    mM.setVisible(true);
    this.dispose();
}
```

Botón de Menú-Acciones de Contingencia

```
private void jMenuItem10MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Contingencia mC = new Contingencia();  
    mC.setVisible(true);  
    this.dispose();  
}
```

Botón de Menú-Matriz de Riesgos

```
private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Matriz mMZ = new Matriz();  
    mMZ.setVisible(true);  
    this.dispose();  
}
```

Botón de Menú-Salir

```
private void jMenuItem5MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Login mL = new Login();  
    mL.setVisible(true);  
    this.dispose();  
}
```

3.2 Gestión de Riesgos

3.2.1 Alta de Riesgos

The screenshot shows a web application window titled "Gestion de Riesgos". The window has a menu bar with the following items: Inicio, Asignacion de Riesgos, Acciones de Mitigacion, Acciones de Contingencia, Matriz de Riesgos, and Salir. Below the menu bar is a tabbed interface with four tabs: Alta (selected), Baja, Modificacion, and Consulta. The main content area is divided into three sections: "Clave de Proyecto:" with a dropdown menu showing "Seleccione un Proyecto"; "Titulo:" with a text input field; and "Descripcion:" with a large text area. At the bottom of the form are two buttons: "Aceptar" and "Cancelar".

Botón Agregar Riesgo

```
private void BtnDelMouseClicked(java.awt.event.MouseEvent evt) {  
    // Verifica que el ID se haya obtenido  
    if (this.TxtID.getText().equals("")) {  
        JOptionPane.showMessageDialog(null, "Seleccione un riesgo antes");  
    } else {  
        if (mBD.ConectarRiesgos()) {  
            //Elimina el riesgo seleccionado  
            if (mBD.EliminarRiesgo(ID)) {  
                JOptionPane.showMessageDialog(null, "Riesgo dado de baja");  
                TxtID.setText("");  
                BorrarE();  
                CBProyecto1.setSelectedIndex(0);  
            } else {  
                JOptionPane.showMessageDialog(null, "Error al Eliminar");  
            }  
        } else {  
            JOptionPane.showMessageDialog(null, "Error");  
        }  
        mBD.DesconectarRiesgos();  
    }  
}
```

Método Agregar Riesgo

```

public boolean AgregarRiesgo(Riesgo mRiesgo) {
    Statement consulta;
    Date Hoy = new Date();
    SimpleDateFormat formatofecha = new SimpleDateFormat("yyyy/MM/dd");
    try {

        consulta = Conexion.createStatement();
        String SQL = "INSERT INTO riesgos.riesgo "
            + "(idriesgo,clave,nombre,detalles,npro,nimp,exposicion,acccmitigacion,"
            + "accccontingencia, fecharevicion, estado,nombreuser)"
            + "VALUES ("
            + "NULL" + ","
            + mRiesgo.getClave() + ","
            + mRiesgo.getTitulo() + ","
            + mRiesgo.getDetalles() + ","
            + "0" + ","
            + "0" + ","
            + "0" + ","
            + "-" + ","
            + "-" + ","
            + formatofecha.format(Hoy) + ","
            + "No Presentado" + ","
            + "-" + ","
            + ");";

        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

3.2.2 Baja de Riesgos

The screenshot displays a web application window titled "Gestion de Riesgos". The window has a menu bar with the following items: Inicio, Asignacion de Riesgos, Acciones de Mitigacion, Acciones de Contingencia, Matriz de Riesgos, and Salir. Below the menu bar, there are four tabs: Alta, Baja (which is the active tab), Modificacion, and Consulta. The main content area of the "Baja" tab contains the following elements:

- A label "Clave de Proyecto:" followed by a dropdown menu with the text "Seleccione un Proyecto".
- A label "ID de Riesgo" followed by a text input field and a button labeled "Eliminar".
- A large, empty rectangular area below the input fields, likely intended for a list or table of risks.

Botón Eliminar Riesgo

```

private void BtnAceptarMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los campos necesarios no esten vacios
    if ((CBProyecto.getSelectedIndex() == 0)) {
        JOptionPane.showMessageDialog(null, "Llene los Campos antes de Guardar");
    } else if ((this.TxtTitulo.getText() == "")) {
        JOptionPane.showMessageDialog(null, "Llene los Campos antes de Guardar");
    } else if ((TxtDes.getText() == "")) {
        JOptionPane.showMessageDialog(null, "Llene los Campos antes de Guardar");
    } else {
        Riesgo mRiesgo = new Riesgo();
        mRiesgo.setClave(CBProyecto.getSelectedItem().toString());
        mRiesgo.setTitulo(TxtTitulo.getText());
        mRiesgo.setDetalles(TxtDes.getText());

        if (mBD.ConectarRiesgos()) {
            //Da de alta el riesgo
            if (mBD.AgregarRiesgo(mRiesgo)) {
                JOptionPane.showMessageDialog(null, "Riesgo dado de Alta exitosamente");
                CBProyecto.setSelectedIndex(0);
                TxtTitulo.setText("");
                TxtDes.setText("");
            } else {
                JOptionPane.showMessageDialog(null, "Error al dar de Alta");
            }
        }
    }
}

```

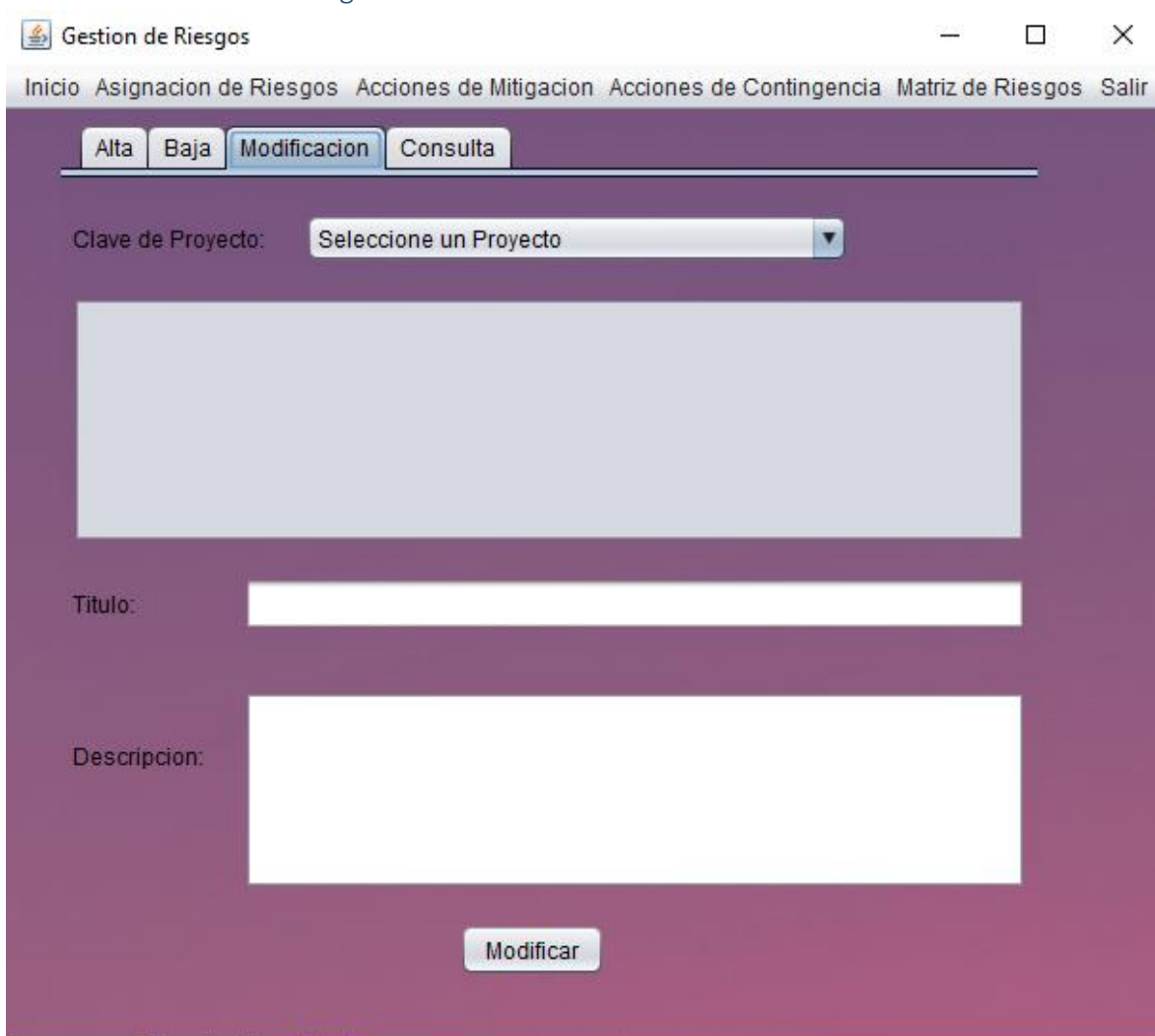
Método de Eliminar Riesgo

```

public boolean EliminarRiesgo(int ID) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        consulta.execute("delete from riesgo"
            + " where idriesgo = " + ID + ";");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```


3.2.3 Modificación de Riesgos



The screenshot displays a web application window titled "Gestion de Riesgos". The window has a menu bar with the following items: Inicio, Asignacion de Riesgos, Acciones de Mitigacion, Acciones de Contingencia, Matriz de Riesgos, and Salir. Below the menu bar, there are four tabs: Alta, Baja, Modificacion (which is currently selected), and Consulta. The main content area of the "Modificacion" tab contains the following elements:

- A label "Clave de Proyecto:" followed by a dropdown menu showing "Seleccione un Proyecto".
- A large, empty light blue rectangular box.
- A label "Titulo:" followed by a text input field.
- A label "Descripcion:" followed by a larger text input area.
- A "Modificar" button located at the bottom center of the form.

Botón de Modificar Riesgo

```

private void BtnModificarMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los campos necesarios no esten vacios
    if (this.TxtTitulo.equals("") || this.TxtDesl.equals("") || CBProyecto3.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(null, "Seleccione un Riesgo y llene los campos");
    } else {
        if (mBD.ConectarRiesgos()) {
            Riesgo mRiesgo = new Riesgo();
            Riesgo nRiesgo = new Riesgo();
            String Tl = "";

            mRiesgo.setClave(TxtID.getText());
            nRiesgo.setTitulo(TxtTitulo.getText());
            nRiesgo.setDetalles(TxtDesl.getText());
            //Modifica el riesgo seleccionado
            if (mBD.ModificarRiesgo(mRiesgo, nRiesgo)) {
                JOptionPane.showMessageDialog(null, "Proyecto Modificado con Exito");
                TxtID.setText("");
                TxtTitulo.setText("");
                TxtDesl.setText("");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en la BD");
        }
        mBD.DesconectarRiesgos();
    }
}

```

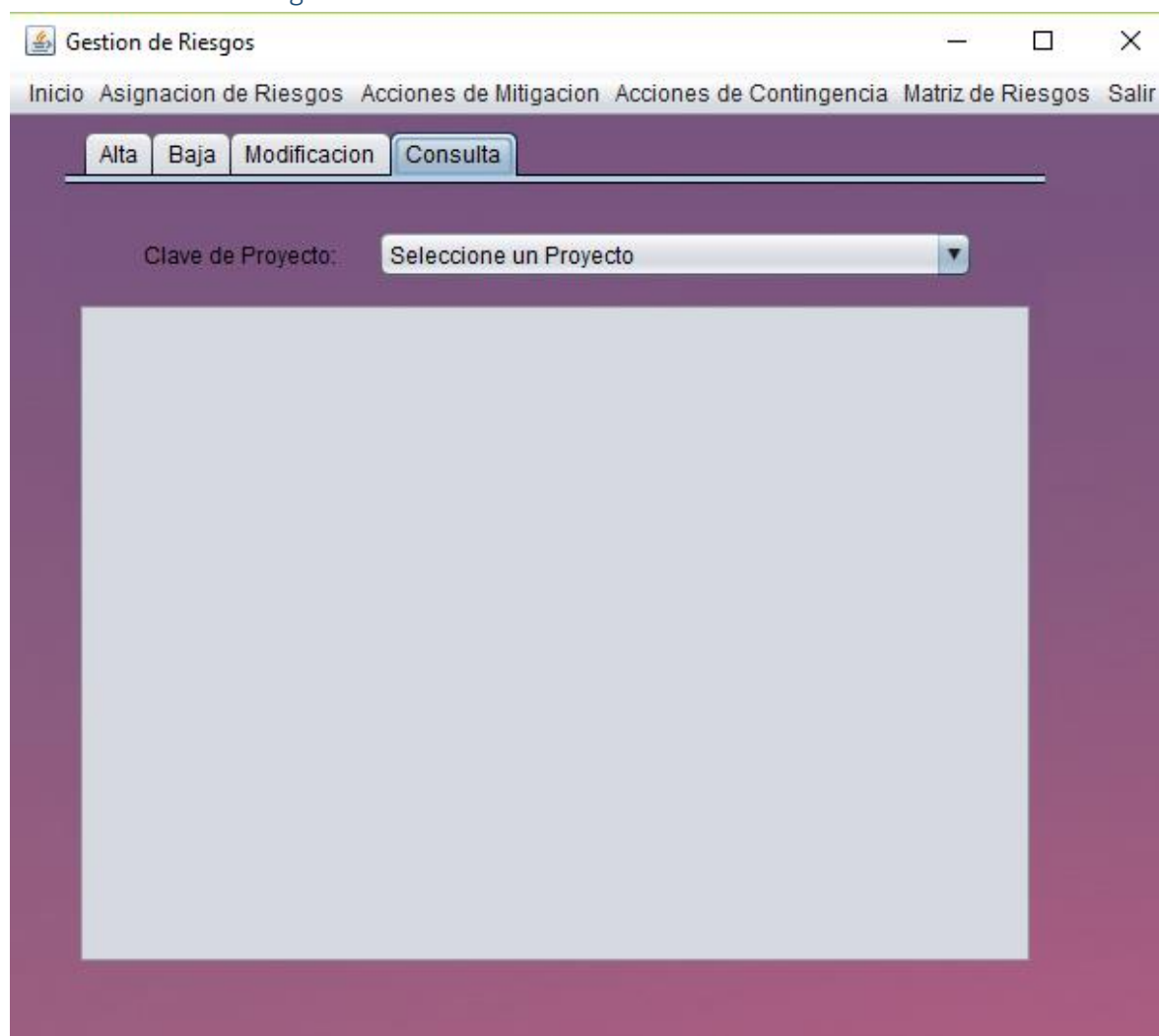
Método de Modificar Riesgo

```

public boolean ModificarRiesgo(Riesgo mRiesgo, Riesgo nRiesgo){
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "UPDATE riesgos.riesgo SET "
            + "nombre = '" + nRiesgo.getTitulo() + "',"
            + "detalles = '" + nRiesgo.getDetalles() + "'"
            + " where clave = '" + mRiesgo.getClave() + "';";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

3.2.4 Consulta de Riesgos



ComboBox de Proyectos

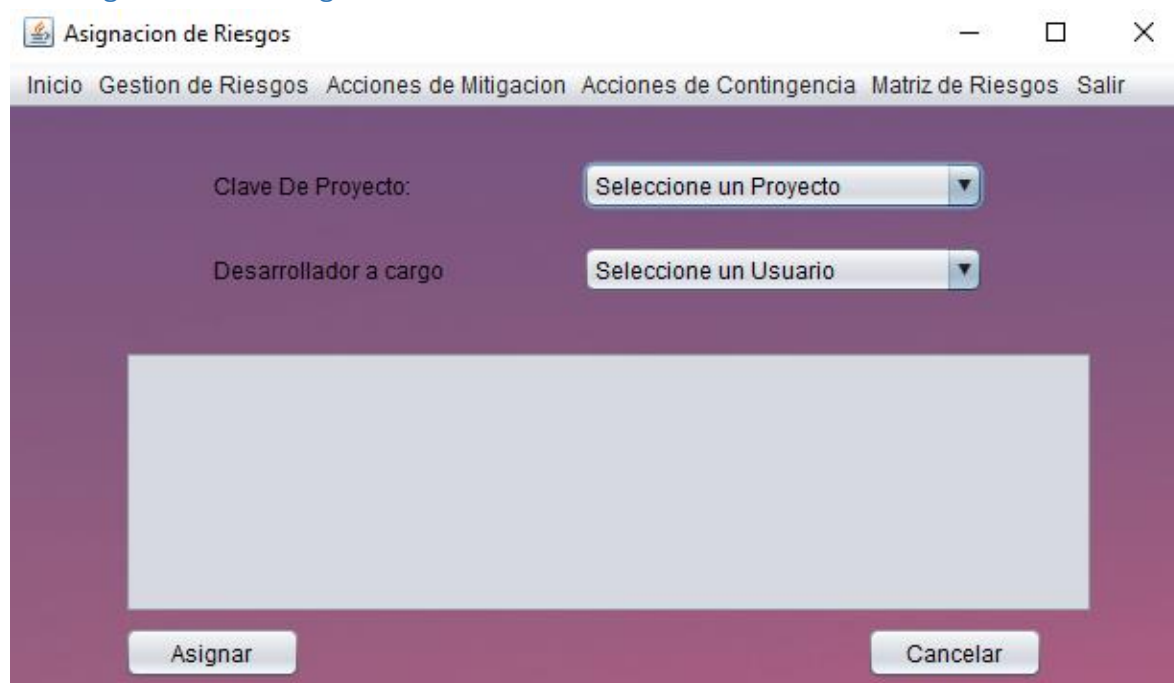
```
private void CBProyecto2ItemStateChanged(java.awt.event.ItemEvent evt) {
    // Llena la tabla en a pestaña de Consulta de riesgo
    if (listo) {
        if (mBD.ConectarRiesgos()) {
            String C = "";
            C = CBProyecto2.getSelectedItem().toString();
            ResultSet ListaC = mBD.ConsultaRiesgos(C);

            this.TBRiesgosC.setModel(Convertidor.convertir(ListaC));
        }
        mBD.DesconectarRiesgos();
    }
}
```

Método de Consulta General de Riesgos

```
public ResultSet ConsultaRiesgos(String C){  
    Statement consulta;  
    ResultSet resultado = null;  
  
    try {  
        consulta = Conexion.createStatement();  
        resultado = consulta.executeQuery("select idriesgo, nombre, detalles from riesgo where clave =' " + C + " '");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return resultado;  
}
```

3.3 Asignación de Riesgos



Asignacion de Riesgos

Inicio Gestion de Riesgos Acciones de Mitigacion Acciones de Contingencia Matriz de Riesgos Salir

Clave De Proyecto:

Desarrollador a cargo

Botón de Asignar Riesgo

```

private void BTNasignarMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Riesgo mR = new Riesgo();
    mR.setID(ID);
    mR.setNombreUser(this.CBDesarrolladores.getSelectedItem().toString());
    //Verifica que los campos necesarios no esten vacios
    if (CBDesarrolladores.getSelectedIndex() == 0 || CBproyecto.getSelectedIndex() == 0 || ID == 0) {
        JOptionPane.showMessageDialog(null, "Seleccione un Riesgo y un Desarrollador");
    } else {
        if (mBD.ConectarAcciones()) {
            //Guarda los datos
            if (mBD.AgregarUsuario(mR)) {
                JOptionPane.showMessageDialog(null, "Se Asigno Desarrollador a Riesgo\ncon Exito");
                listo = false;
                listo2 = false;
                LlenarC();
                LlenarD();
                CBproyecto.setSelectedIndex(0);
                CBDesarrolladores.setSelectedIndex(0);
                Borrar();
                ID = 0;
            } else {
                JOptionPane.showMessageDialog(null, "Error al agregar Accion de Contingencia");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
    }
}

```

Método de Asignar Riesgo

```

public boolean AgregarUsuario(Riesgo mRiesgo) {
    Statement consulta;

    try {
        consulta = Conexion.createStatement();
        consulta.execute("update riesgo set " +
            "nombreuser = '" + mRiesgo.getNombreUser() + "'" +
            " where idriesgo = '" + mRiesgo.getID() + "';");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

3.4 Acciones de Mitigación

Botón de Agregar Acción de Mitigación

```
private void BTNagrearMouseClicked(java.awt.event.MouseEvent evt) {
    //Verifica que los campos no esten vacios
    if (ID == 0 || this.TXTmit.getText() == "" || this.CBProyecto.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(null, "Seleccione un riesgo y agregue la accion de Mitigacion");
    } else {
        Riesgo mR = new Riesgo();
        mR.setID(ID);
        mR.setMitigacion(this.TXTmit.getText());
        if (mBD.ConectarAcciones()) {
            //Da de alta la accion de mitigacion o la modifica
            if (mBD.AgregarMitigacion(mR)) {
                listo = false;
                LlenarC();
                JOptionPane.showMessageDialog(null, "Se agrego Accion de Mitigacion\n\tcon Exito");
                TXTmit.setText("");
                CBProyecto.setSelectedIndex(0);
                Borrar();
            } else {
                JOptionPane.showMessageDialog(null, "Error al agregar Accion de Mitigacion");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
    }
}
```

Método de Agregar Acción de Mitigación

```
public boolean AgregarMitigacion(Riesgo mRiesgo) {  
    Statement consulta;  
    String sql = "";  
    try {  
        consulta = Conexion.createStatement();  
        sql = "update riesgo set " +  
            "accmitigacion = '" + mRiesgo.getMitigacion() + "'" +  
            " where idriesgo = '" + mRiesgo.getID() + "';";  
        consulta.execute(sql);  
        return true;  
    } catch (Exception e) {  
        e.printStackTrace();  
        return false;  
    }  
}
```

3.5 Acciones de Contingencia

Asignar Accion de Contingencia

Inicio Gestion de Riesgos Asignacion de Riesgos Acciones de Mitigacion Matriz de Riesgos Salir

Clave De Proyecto:

Seleccione Un Riesgo:

Accion De Contingencia:

Botón de Agregar Acción de Contingencia

```

private void BTNagrearMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Riesgo mR = new Riesgo();
    mR.setID(ID);
    mR.setContingencia(this.TXTmit.getText());
    //Verifica que los Campos necesarios no esten vacios
    if (this.TXTmit.getText() == "" || this.CBProyecto.getSelectedIndex() == 0 || ID == 0) {
        JOptionPane.showMessageDialog(null, "Seleccione un riesgo y agregue la accion de Contingencia");
    } else {
        if (mBD.ConectarAcciones()) {
            //Agrega o Modifica la Accion de mitigacion a el riesgo seleccionado
            if (mBD.AgregarContingencia(mR)) {
                JOptionPane.showMessageDialog(null, "Se agrego Accion de Contingencia\ncon Exito");
                listo = false;
                LlenarC();
                TXTmit.setText("");
                CBProyecto.setSelectedIndex(0);
                Borrar();
            } else {
                JOptionPane.showMessageDialog(null, "Error al agregar Accion de Contingencia");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
    }
}
}

```

Método de Agregar Acción de Contingencia

```

public boolean AgregarContingencia(Riesgo mRiesgo) {
    Statement consulta;

    try {
        consulta = Conexion.createStatement();
        consulta.execute("update riesgo set " +
            "acccontingencia = '" + mRiesgo.getContingencia() + "'" +
            " where idriesgo = '" + mRiesgo.getID() + "';");
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```


3.6 Matriz de Riesgos



ComboBox de Proyectos

```
private void CBProyectoItemStateChanged(java.awt.event.ItemEvent evt) {
    // Llena la tabla dependiendo del elemento seleccionado en el combobox
    if (listo) {
        if (mBD.ConectarRiesgos() ) {
            String C = "";
            C = CBProyecto.getSelectedItem().toString();
            ResultSet Lista = mBD.Matriz(C);
            this.TBProyecto.setModel(Desarrollador.Convertidor.convertir(Lista));
        }
        mBD.DesconectarRiesgos();
    }
}
```

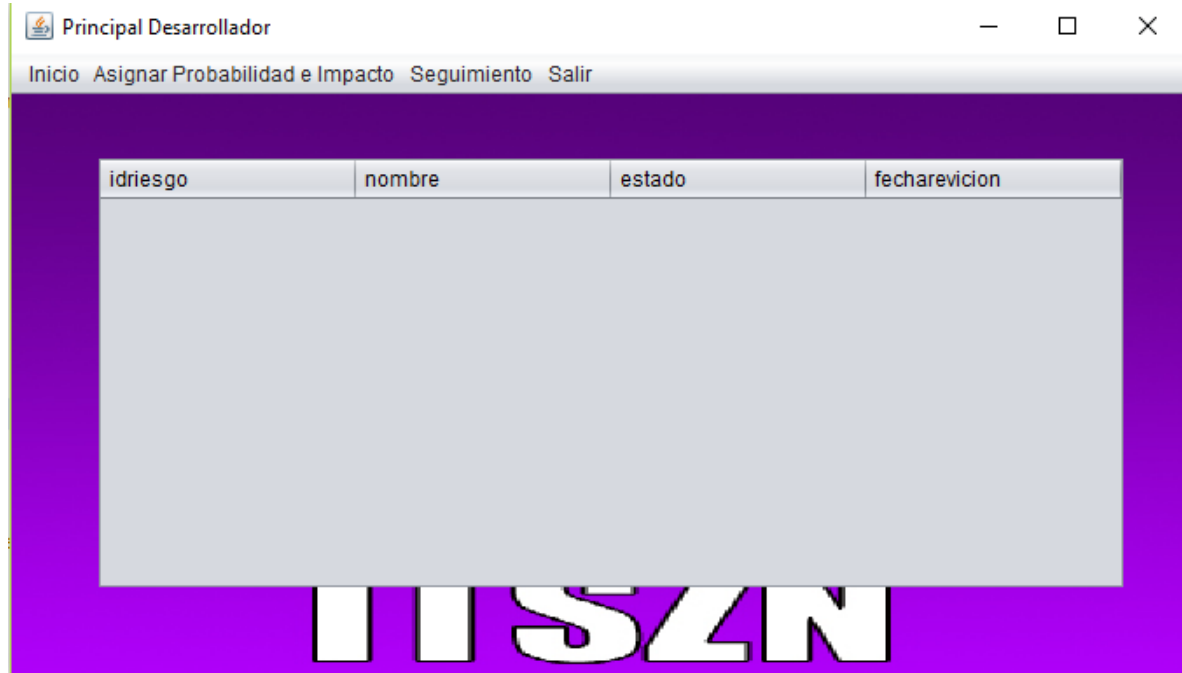
Método de Consulta

```
public ResultSet Matriz(String C) {
    ArrayList mLista = new ArrayList();
    Statement consulta;
    ResultSet resultado = null;

    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select idriesgo, clave, nombre, npro, nimp, "
            + "exposicion, accmitigacion, acccontingencia, fecharevision, estado, nombreuser from riesgo"
            + " where clave =" + C + " ");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return resultado;
}
```

IV. Desarrollador

4.1 Página Principal de Desarrollador



Botón de Menú-Asignar-Inicio

```
private void jMenuItem1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new PrincipalDesarrollador().setVisible(true);
    this.dispose();
}
```

Botón de Menú-Asignar-Probabilidad e Impacto

```
private void jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new AsignarProbabilidad().setVisible(true);
    this.dispose();
}
```

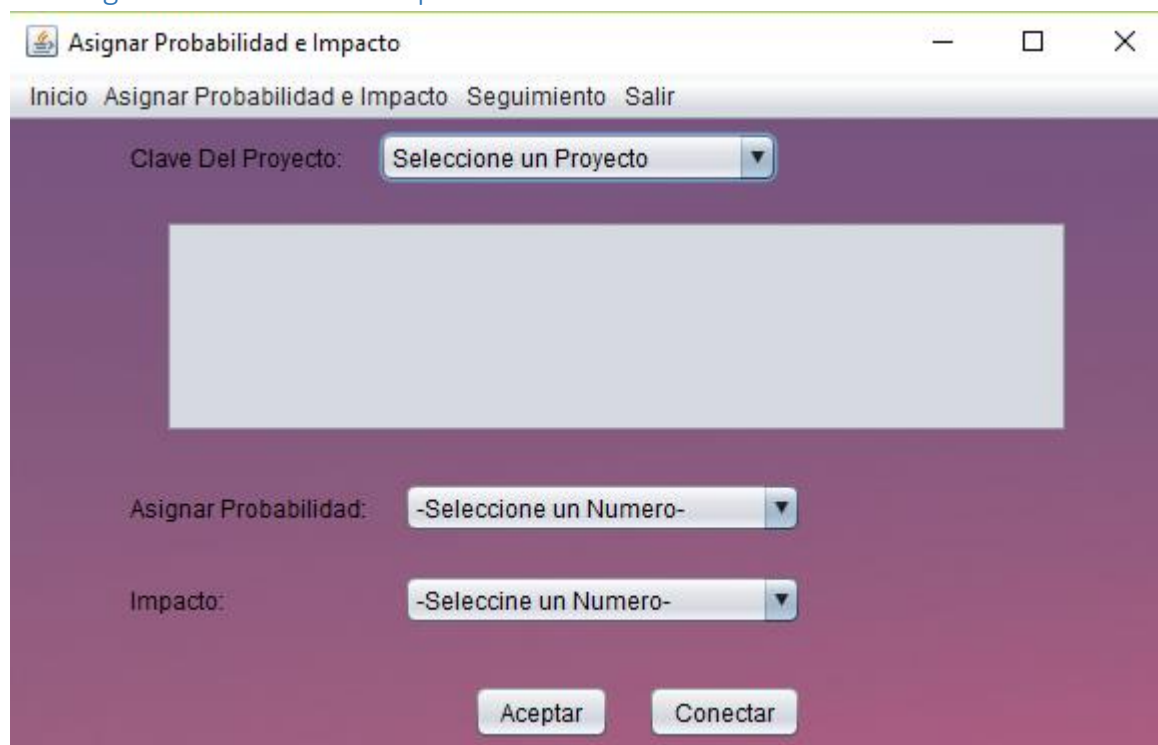
Botón de Menú-Seguimiento

```
private void jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new SeguimientoDeRiesgo().setVisible(true);
    this.dispose();
}
```

Botón de Menú-Salir

```
private void jMenu5MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    Login mL = new Login();  
    mL.setVisible(true);  
    this.dispose();  
}
```

4.2 Asignar Probabilidad e Impacto



Asignar Probabilidad e Impacto

Inicio Asignar Probabilidad e Impacto Seguimiento Salir

Clave Del Proyecto: Seleccione un Proyecto ▼

Asignar Probabilidad: -Seleccione un Numero- ▼

Impacto: -Seleccine un Numero- ▼

Aceptar Conectar

Botón de Agregar Probabilidad e Impacto

```

private void BTNAceptarMouseClicked(java.awt.event.MouseEvent evt) {
    // Verifica que los campos no esten vacios
    if (ID == 0 || CBProbabilidad.getSelectedIndex() == 0 || CBImpacto.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(null, "Seleccione un riesgo y llene los datos que se piden");
    } else {
        mR.setID(ID);
        mR.setNpro(this.CBProbabilidad.getSelectedIndex());
        mR.setNimp(this.CBImpacto.getSelectedIndex());
        mR.setNombreUser(Usuario.User);
        if (mBD.Conectar()) {
            //Verificacion de que el usuario no ala dado de alta valores a e mismo riesgo anteriormente
            if (mBD.Duplicidad(mR)) {
                JOptionPane.showMessageDialog(null, "Ya se a asignado una probabilidad y un impacto con este usuario");
            } else {
                //Se dan de alta los valores en la tabla para sacar probabilidad e impacto
                if (mBD.AltaPIE(mR)) {
                    JOptionPane.showMessageDialog(null, "Valores agregados exitosamente");
                    this.CBProbabilidad.setSelectedIndex(0);
                    this.CBImpacto.setSelectedIndex(0);

                    //Agregar campos a tabla riesgos
                    mR = new Riesgo();
                    mR.setID(ID);
                    mR.setNpro(Math.round(mBD.Probabilidad(mR)));
                    mR.setNimp(Math.round(mBD.Impacto(mR)));
                    mR.setExp((mR.getNpro() * mR.getNimp()));
                    mBD.ValoresRiesgo(mR);

                    //Actualiza la tabla
                    Borrar();
                    if (listo) {
                        if (mBD.Conectar()) {
                            String C = "";
                            C = CBProyecto.getSelectedItem().toString();
                            ResultSet Lista = mBD.ConsultaRiesgos2(C);
                            this.TBRiesgos.setModel(Convertidor.convertir(Lista));
                        }
                    }
                    mBD.Desconectar();
                } else {
                    JOptionPane.showMessageDialog(null, "Error al agregar");
                }
            }
        } else {
            JOptionPane.showMessageDialog(null, "Error en BD");
        }
        mBD.Desconectar();
    }
}

```

Método de Agregar Probabilidad e Impacto

```
//Alta de valores para tabla valores
public boolean AltaPIE(Riesgo mR) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "INSERT INTO riesgos.valores "
            + "(idvalor, idriesgo, probabilidad, impacto, nombreuser) "
            + "VALUES ("
            + "null" + ","
            + mR.getID() + ","
            + mR.getNpro() + ","
            + mR.getNimp() + ","
            + mR.getNombreUser() + ")";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}
```

Método para obtener Promedio de la Probabilidad de un Riesgo

```
public float Probabilidad (Riesgo mR){
    float valor = 0;
    Statement consulta;
    ResultSet resultado = null;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select avg(probabilidad) AS promedio from valores " +
            "where idriesgo = ' " + mR.getID() + "';");
        while (resultado.next()) {
            valor = resultado.getFloat("promedio");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return valor;
}
```

Método para obtener Promedio del Impacto de un Riesgo

```

public float Impacto(Riesgo mR) {
    float impacto = 0;
    Statement consulta;
    ResultSet resultado = null;
    try {
        consulta = Conexion.createStatement();
        resultado = consulta.executeQuery("select avg(impacto) AS promedio from valores " +
            "where idriesgo =" + mR.getID() + " ");

        while (resultado.next()) {
            impacto = resultado.getFloat("promedio");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return impacto;
}

```

Método para Agregar los Valores a la Tabla Principal

```

public boolean ValoresRiesgo(Riesgo mR) {
    Statement consulta;
    try {
        consulta = Conexion.createStatement();
        String SQL = "update riesgo set "
            + "npro = " + mR.getNpro() + ","
            + "nimp = " + mR.getNimp() + ","
            + "exposicion = " + mR.getExp()
            + " where idriesgo = " + mR.getID() + " ";
        consulta.execute(SQL);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```