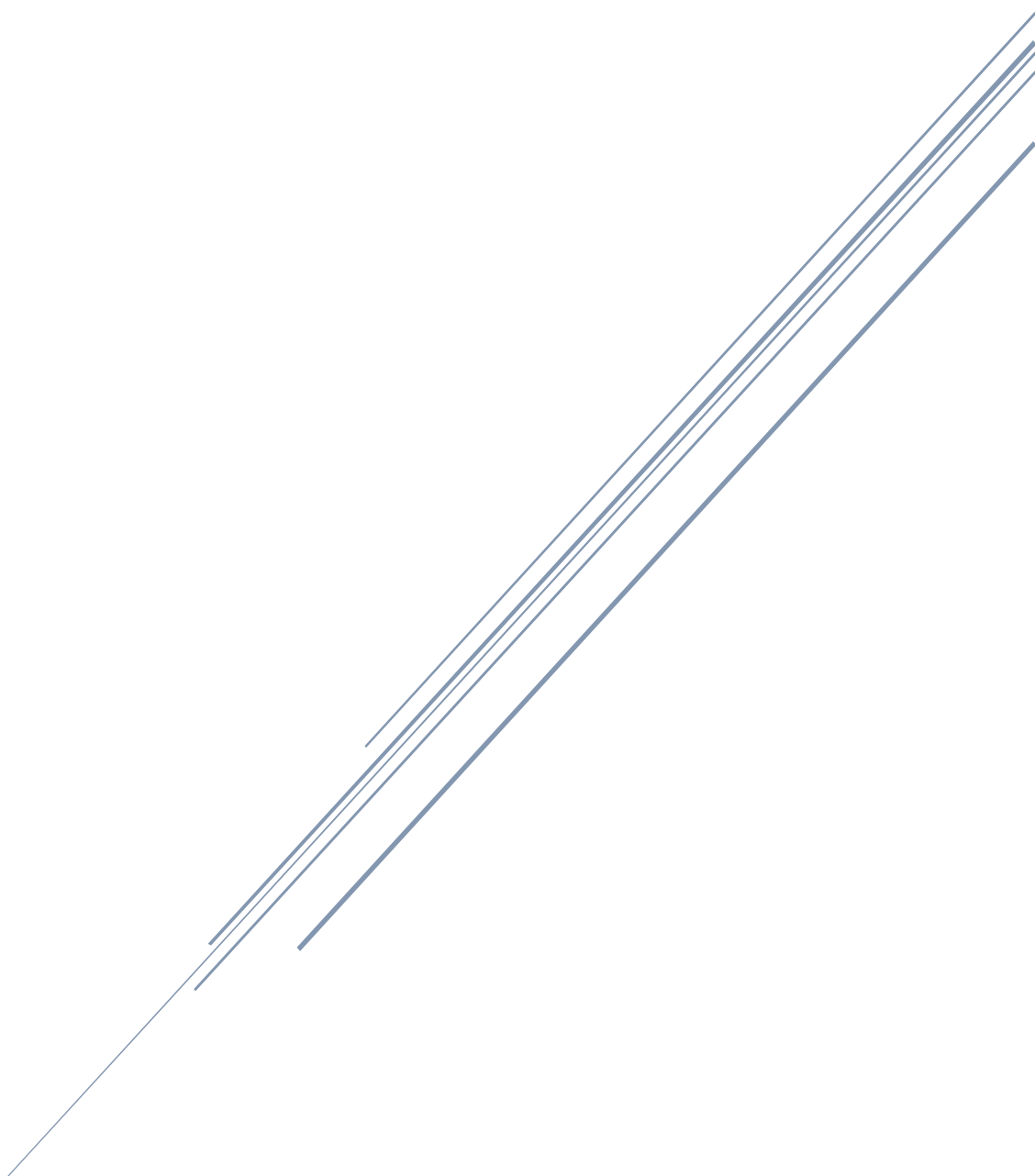


PROJETO JAVA

Sistema de Controle



Senai Almirante Tamandaré
Programação Java

Depois de criar o projeto no NetBeans, do lado esquerdo do NetBeans, expandir o nome do projeto, expandir Project Files, e abri o arquivo pom.xml

Você pode procurar a dependência do MySql no site abaixo (o que vamos usar é o trecho destacado em amarelo)

<https://mvnrepository.com/artifact/mysql/mysql-connector-java>

01) Incluir no arquivo pom.xml a dependência abaixo:

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.33 -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
  </dependency>
</dependencies>
```

Criação da TelaLogin

a) Criar um pacote Telas

Criar uma JFrame com nome TelaLogin

Na tela de login incluir label:

Usuário: lblUsuário

Senha: lblSenha

Na tela de login incluir Text Field:

Alterar a variável da caixa de texto para txtUsuário

Na tela de login incluir Password Field:

Alterar a variável da caixa de texto para txtSenha

Na tela de login incluir Button:

Alterar a variável do botão para btnLogin

b) Alterar as propriedades

b.1) Incluir nome na tela, em propriedades, title colocar Login

b.2) Em propriedades resize, desmarcar pra que o usuário não altere o tamanho da tela de login

b.3) Em Código, marcar centralizado, para ficar no meio da tela

=====

Agora vamos criar o módulo de conexão que vai fazer a ligação entre o código Java e o banco de dados MySQL

Vamos criar um package para organizar o projeto e separar o módulo de conexão dos demais código que serão desenvolvidos

- 1) Criar um pacote ModuloConexao
- 2) Criar uma classe ModuloConexao

Abaixo vamos criar o código para estabelecer a conexão com o Banco de Dados (destacado em amarelo, podemos copiar e incluir no classe criada acima chamada ModuloConexao

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.*;

public class ModuloConexao {

    //Método responsável por fazer a conexão com o banco

    public static Connection conector() {

        java.sql.Connection conexao = null;

        //a linha abaixo chama o Driver

        String driver = "com.mysql.jdbc.Driver";

        //armazenado informações referente ao banco

        String url = "jdbc:mysql://localhost:3306/dbempresa";

        String user = "root";

        String password = "Sen@i2025";

        //estabelecer conexão com banco

        try {

            Class.forName(driver);

            conexao = DriverManager.getConnection(url, user, password);

            return conexao;

        } catch (Exception e){

            System.out.println(e);

            return null;

        }

    }

}
```

```
}  
}
```

Vamos usar a TelaLogin e incluir o método para executar a conexão que criamos acima, assim garantimos que o Java consegue se comunicar com o banco de dados MySQL (usar o código destacado em amarelo)

```
import com.mycompany.sistemacontrole.ModuloConexao.ModuloConexao;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
public class TelaLogin extends javax.swing.JFrame {
```

```
    Connection conexao = null;
```

```
    PreparedStatement pst = null;
```

```
    ResultSet rs = null;
```

```
    /**
```

```
     * Creates new form TelaLogin
```

```
     */
```

```
    public TelaLogin() {
```

```
        initComponents();
```

```
        conexao = ModuloConexao.conector();
```

```
        //linha abaixo server de apoio para o desenvolvimento
```

```
        //System.out.println(conexao);
```

```
        if(conexao != null){
```

```
            lblStatus.setText("Conectado");
```

```
        } else {
```

```
            lblStatus.setText("Não conectado");
```

```
        }
```

```
    }
```

Depois de estabelecer a conexão entre o Java e o MySQL, vamos criar o método Logar, para validar no banco de dados se o usuário e senha existem, e para criar o método vamos usar o código abaixo destacado em amarelo

```
public void logar(){

    String sql = "select * from tbusuarios where login=? and senha=?";

    try {

        //as linhas abaixo preparam a consulta ao banco com base na digitação das caixas de
        texto

        pst = conexao.prepareStatement(sql);

        pst.setString(1, txtUsuario.getText());

        String captura = new String(txtSenha.getPassword());

        //pst.setString(2, txtSenha.getText());

        pst.setString(2, captura);


        //a linha abaixo executa a consulta

        rs = pst.executeQuery();


        //estrutura de decisão

        if(rs.next()) {

            TelaPrincipal principal = new TelaPrincipal();

            principal.setVisible(true);

            //linha abaixo fecha a tela de login

            this.dispose();

            //linha abaixo fecha a conexão com o banco de dados

            conexao.close();

        } else {

            JOptionPane.showMessageDialog(null,"usuário e ou senha inválido(s)");

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}
```

=====

Depois que fazemos o login em um sistema, teremos acesso ao que este sistema disponibiliza para ser usado, então vamos criar uma tela chamada TelaPrincipal, onde os usuários poderão interagir com o que o sistema se propõem para entregar.

Então, dentro do package Telas, criar um JFrame chamado TelaPrincipal

Em Propriedade:

Title: Sistema Ordem de Serviço

Resizable: desmarcar, para que o usuário não mexa no tamanho da tela

Em Código:

Generate Center: marcar para a tela ser gerada no centro

Form Size Policy: Generate Resize Code

Form Size: 935, 529

Com a TelaPrincipal criada, vamos criar o Menu para o usuário realizar seu trabalho

Em Paletas => Swing Menu (lado direito da tela está o menu Paletas, onde temos os objetos que podemos usar em nossa tela)

Menu Bar: arrastar para o JFrame

Depois de arrastar o Swing Menu para tela vamos realizar as etapas abaixo:

Alterar o nome da variável para Menu

File: alterar para Cadastro

Alterar o nome da variável para menCad

Edit: alterar para Relatório

Alterar o nome da variável para menRel

Botão direito na barra de menu e add menu,

Mudar o nome para Ajuda

Alterar o nome da variável para menAju

Botão direito na barra de menu e add menu,

Mudar o nome para Opções

Alterar o nome da variável para menOpc

Agora criar os itens de cada menu

Em Cadastro:

Botão direito em Cadastro => Add From Pallete => Menu Item => mudar o nome para Cliente

Clicar duas vezes na frente da palavra Cliente para incluir a tecla de atalho (ALT+C)

Alterar o nome da variável para menCadCli

Botão direito em Cadastro => Add From Pallete => Menu Item => mudar o nome para OS

Clicar duas vezes na frente da palavra OS para incluir a tecla de atalho (ALT+O)

Alterar o nome da variável para menCadOs

Botão direito em Cadastro => Add From Pallete => Menu Item => mudar o nome para Usuários

Clicar duas vezes na frente da palavra Usuários para incluir a tecla de atalho (ALT+U)

Alterar o nome da variável para menCadUsu

Em Relatórios:

Botão direito em Relatórios => Add From Pallete => Menu Item => mudar o nome para Serviços

Clicar duas vezes na frente da palavra Serviços para incluir a tecla de atalho (ALT+S)

Alterar o nome da variável para menRelSer

Em Ajuda:

Botão direito em Ajuda => Add From Pallete => Menu Item => mudar o nome para Sobre

Clicar duas vezes na frente da palavra Serviços para incluir a tecla de atalho (ALT+F1)

Alterar o nome da variável para menAjuSob

Em Opções:

Botão direito em Opções => Add From Pallete => Menu Item => mudar o nome para Sair

Clicar duas vezes na frente da palavra Serviços para incluir a tecla de atalho (ALT+F4)

Alterar o nome da variável para menOpcSai

Incluir no JFrame o Desktop Pane, (que está dentro de Palletas, Swing Containers), para que todos os formulários sejam abertos dentro da área de Desktop

Alterar o nome da variável para desktop

Em Propriedades:

PreferredSize: 640, 480

Horizontal Size: Default

Vertical Size: Default

Vamos incluir uma label para mostrar o nome do usuário que efetuou o login no sistema

Text: Usuário

Font: 18

Alterar o nome da variável para lblUsuario

E também incluir uma label para mostrar a data que o login foi efetuado

Text: Data

Font: 18 e Negrito

Alterar o nome da variável para lblData

=====

Agora que temos a label data em nossa TelaPrincipal, vamos escrever o código que mostra a data do login efetuado no sistema

Clicar na TelaPrincipal com o botão direito => Events => Window => Window Activated, será aberto o método, copiar o trecho destacado em amarelo e incluir no código, atenção ao local que será incluído o código copiado

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {  
    // o código abaixo substitui a label data com a data do sistema  
    Date data = new Date();  
    //lblData.setText(data.toString());  
    DateFormat formatador = DateFormat.getDateInstance(DateFormat.SHORT);  
    lblData.setText(formatador.format(data));  
}
```


=====

Voltamos para o menu da TelaPrincipal, vamos codificar a opção Sair

Clicar com o botão direito na Opção Sair, Eventos => Action => ActionPerformed, vamos copiar o trecho destacado em amarelo e copiar para o código, atenção ao local que será incluído o código copiado

```
private void MenOpcSaiActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// Exibe a caixa de dialogo
```

```
    int sair = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja sair  
?", "Atenção", JOptionPane.YES_NO_OPTION);
```

```
    if(sair == JOptionPane.YES_OPTION) {
```

```
        System.exit(0);
```

```
    }
```

```
}
```

=====

No menu Ajuda, vamos precisar criar uma tela para incluir a especificação do sistema que será criado, será necessário criar uma JFrame chamada TelaSobre

Title: Sobre

Label: Sistema de controle de ordem de serviço

Laber: Licença GPL

Em Propriedades => defaultCloseOperation, mudar para Dispose

Botão direito no Item de Menu Sobre => Eventos => Action => ActionPerfomed

Dentro do Método incluir o código abaixo

```
TelaSobre sobre = new TelaSobre();
```

```
sobre.setVisible(true);
```

=====

Vamos pensar nas permissões que um usuário pode ter em um sistema, se o usuário tem o perfil de admin, ele tem acesso ao menu completo, já se ele é um usuário com perfil user, a permissão será limitada, vamos preparar o menu para ser manipulado dentro do código para quando o usuário for admin e user.

No menu Cadastro => clicar em Usuários => Propriedades => Code => Modificador de Variáveis, trocar para público e marcar static, clicar em Ok

Em propriedades, Enabled, desmarcar a opção, clicar em Close

No menu Relatório => clicar em Relatório => Propriedades => Code => Modificador de Variáveis, trocar para público e marcar static, clicar em Ok

Em propriedades, Enabled, desmarcar a opção, clicar em Close

=====

Para fazer a verificação se o usuário é admin o user, o trecho abaixo destacado em amarelo deve ser incluído no método login()

```
public void login(){  
    String sql = "select * from tbusuarios where login=? and senha=?";  
    try {  
        //as linhas abaixo preparam a consulta ao banco com base na digitação das caixas de  
        texto  
        pst = conexao.prepareStatement(sql);  
        pst.setString(1, txtUsuario.getText());  
        String captura = new String(txtSenha.getPassword());  
        //pst.setString(2, txtSenha.getText());  
        pst.setString(2, captura);  
  
        //a linha abaixo executa a consulta  
        rs = pst.executeQuery();  
  
        //estrutura de decisão  
        if(rs.next()) {  
            //captura o perfil do usuário  
            String perfil = rs.getString(6);  
            //System.out.println(perfil);  
            //a estrutura abaixo faz a validação do perfil do usuário  
            if(perfil.equals("admin")){  
                TelaPrincipal principal = new TelaPrincipal();  
                principal.setVisible(true);  
                TelaPrincipal.menCadUsu.setEnabled(true);  
            }  
        }  
    }  
}
```

```

        TelaPrincipal.menRel.setEnabled(true);

        //linha abaixo fecha a tela de login
        this.dispose();

        //linha abaixo fecha a conexão com o banco de dados
        conexao.close();

    } else {

        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);

        //linha abaixo fecha a tela de login
        this.dispose();

    }

} else {

    JOptionPane.showMessageDialog(null,"usuário e ou senha inválido(s)");

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);

}

}

```

=====

Alterar a label usuário da tela principal para mostrar qual usuário está logado

Clicar na label usuário botão direito => propriedades => código, em modificador de variável trocar de private para public e marcar static

```

public void logar(){

    String sql = "select * from tbusuarios where login=? and senha=?";

    try {

        //as linhas abaixo preparam a consulta ao banco com base na digitação das caixas de
        texto

        pst = conexao.prepareStatement(sql);

        pst.setString(1, txtUsuario.getText());

        String captura = new String(txtSenha.getPassword());

        //pst.setString(2, txtSenha.getText());
    }
}

```

```
pst.setString(2, captura);

//a linha abaixo executa a consulta
rs = pst.executeQuery();

//estrutura de decisão
if(rs.next()) {
    //captura o perfil do usuário
    String perfil = rs.getString(6);
    //System.out.println(perfil);
    //a estrutura abaixo faz a validação do perfil do usuário
    if(perfil.equals("admin")){
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
        TelaPrincipal.MenCadUsu.setEnabled(true);
        TelaPrincipal.MenRel.setEnabled(true);
        TelaPrincipal.lblUsuario.setText(rs.getString(2));
        TelaPrincipal.lblUsuario.setForeground(Color.red);
        //linha abaixo fecha a tela de login
        this.dispose();
        //linha abaixo fecha a conexão com o banco de dados
        conexao.close();
    } else {
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
        TelaPrincipal.lblUsuario.setText(rs.getString(2));
        TelaPrincipal.lblUsuario.setForeground(Color.blue);
        //linha abaixo fecha a tela de login
        this.dispose();
    }
} else {
```

```

        JOptionPane.showMessageDialog(null,"usuário e ou senha inválido(s)");
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}

```

=====

Vamos criar um JInternal frame para cadastro de usuários

Clicar com o botão direito no pacote Telas => Outros => Swing GUI Forms => Form

JInternalFrame Form => Clicar em Next => Nome da classe: TelaUsuario => Clicar em Finish

Em Propriedades:

Title: Usuários

PreferredSize: 640, 480

Closeable: marcar

Iconifiable: marcar

Maximizable: marcar

Em Código:

Form Size Policy: Generate Resize Code

Designer Size: 640, 480

Depois de criar a TelaUsuario, voltar na TelaPrincipal => Cadastro => Usuários => Events => Action => ActionPerformed

```

private void menCadUsuActionPerformed(java.awt.event.ActionEvent evt) {
    //Abre a TelaUsuario dentro do desktop
    TelaUsuario usuario = new TelaUsuario();
    usuario.setVisible(true);
    desktop.add(usuario);
}

```

=====

Agora que a TelaUsuario já está associada ao menu, vamos criar seus campos e botões

Incluir na TelaUsuario as labels:

Id

Nome

Fone

Login

Senha

Perfil

Incluir na TelaUsuario os campos de texto:

Uma caixa de texto para Id

Alterar a variável da caixa de texto id para txtUsuld

Uma caixa de texto para Nome

Alterar a variável da caixa de texto Nome para txtUsuNome

Uma caixa de texto para Fone

Alterar a variável da caixa de texto Nome para txtUsuFone

Uma caixa de texto para Login

Alterar a variável da caixa de texto Login para txtUsuLogin

Uma caixa de texto para Senha

Alterar a variável da caixa de texto Senha para txtUsuSenha

Incluir na TelaUsuario um Combo Box para Perfil:

Alterar a variável do Combo Box Perfil para cboUsuPerfil

Em propriedades do Combo Box => Model => Retirar os itens defaults e incluir admin e user

Incluir 4 botões na TelaUsuario

No Primeiro botão

Em propriedades:

Title: Adicionar

Alterar nome da variável para btnUsuCreate

No Segundo botão

Em propriedades:

Title: Consultar

Alterar nome da variável para btnUsuRead

No Terceiro botão

Em propriedades:

Title: Alterar

Alterar nome da variável para btnUsuUpdate

No Quarto botão

Em propriedades:

Title: Apagar

Alterar nome da variável para btnUsuDelete

=====

Vamos codificar o botão Consultar, este código deve ser criado na TelaUsuario

```
import java.sql.*;
```

```
import com.mycompany.sistemacontrole.ModuloConexao.ModuloConexao;
```

```
import javax.swing.JOptionPane;
```

```
public class TelaUsuario extends javax.swing.JInternalFrame {
```

```
    Connection conexao = null;
```

```
    PreparedStatement pst = null;
```

```
    ResultSet rs = null;
```

```
/**
```

```
 * O código abaixo é um construto, é usado para inicializar o Módulo de Conexão
```

```
 */
```

```
public TelaUsuario() {
```

```
    initComponents();
```

```
    conexao = ModuloConexao.conector();
```

```
}
```

```
//Método Consultar
```

```
public void consultar(){
```

```
    String sql = "select * from tbusuarios where iduser=?";
```

```
    try {
```

```
        pst=conexao.prepareStatement(sql);
```

```
        pst.setString(1, txtUsuld.getText());
```

```
        rs=pst.executeQuery();
```

```
        if(rs.next()){
```

```
            txtUsuNome.setText(rs.getString(2));
```

```
            txtUsuFone.setText(rs.getString(3));
```

```
            txtUsuLogin.setText(rs.getString(4));
```

```
            txtUsuSenha.setText(rs.getString(5));
```

```
            cboUsuPerfil.setSelectedItem(rs.getString(6));
```

```
        } else {
```

```
            JOptionPane.showMessageDialog(null, "Usuário não cadastrado.");
```

```
            //as linhas abaixo limpam os campos
```

```
            txtUsuNome.setText(null);
```

```
            txtUsuFone.setText(null);
```

```
            txtUsuLogin.setText(null);
```

```
            txtUsuSenha.setText(null);
```

```
            cboUsuPerfil.setSelectedItem(null);
```

```
        }
```



```

    } catch (Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

}

```

=====

Vamos codificar o botão Adicionar, deve ser criado no código da TelaUsuario

```

//Método Adicionar

private void adicionar(){

    String sql = "insert into tbusuarios (iduser, usuario, fone, login, senha, perfil) values
    (?, ?, ?, ?, ?, ?)";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtUsuld.getText());

        pst.setString(2, txtUsuNome.getText());

        pst.setString(3, txtUsuFone.getText());

        pst.setString(4, txtUsuLogin.getText());

        pst.setString(5, txtUsuSenha.getText());

        pst.setString(6, cboUsuPerfil.getSelectedItem().toString());

        //validação dos campos obrigatórios

        if((txtUsuld.getText().isEmpty()) || (txtUsuNome.getText().isEmpty()) ||
        txtUsuLogin.getText().isEmpty() || txtUsuSenha.getText().isEmpty()) {

            JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");

        } else {

            int adicionado = pst.executeUpdate();

            if(adicionado > 0) {

                JOptionPane.showMessageDialog(null, "Usuário adicionado com sucesso !");

                txtUsuld.setText(null);

                txtUsuNome.setText(null);

                txtUsuFone.setText(null);

                txtUsuLogin.setText(null);
            }
        }
    }
}

```

```

        txtUsuSenha.setText(null);

        //cboUsuPerfil.setSelectedItem(null);

    }

}

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);

}

}

```

=====

Vamos codificar o botão Alterar, deve ser criado no código da TelaUsuario

//Método Alterar

```

private void alterar() {

    String sql = "update tbusuarios set usuario=?, fone=?, login=?, senha=?, perfil=? where iduser=?";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtUsuNome.getText());

        pst.setString(2, txtUsuFone.getText());

        pst.setString(3, txtUsuLogin.getText());

        pst.setString(4, txtUsuSenha.getText());

        pst.setString(5, cboUsuPerfil.getSelectedItem().toString());

        pst.setString(6, txtUsuld.getText());

    }

    //validação dos campos obrigatórios

    if((txtUsuld.getText().isEmpty()) || (txtUsuNome.getText().isEmpty()) || txtUsuLogin.getText().isEmpty() || txtUsuSenha.getText().isEmpty()) {

        JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");

    } else {

        int adicionado = pst.executeUpdate();

        if(adicionado > 0) {

            JOptionPane.showMessageDialog(null, "Dados o usuário alterados com sucesso!");

```

```

        txtUsuld.setText(null);
        txtUsuNome.setText(null);
        txtUsuFone.setText(null);
        txtUsuLogin.setText(null);
        txtUsuSenha.setText(null);
    }
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}
}

```

=====

Vamos codificar o botão Apagar, deve ser criado no código da TelaUsuario

//Método Remover

```

private void remover() {
    int confirma=JOptionPane.showConfirmDialog(null, "Tem certeza que deseja remover o
usuário ? ", "Atenção", JOptionPane.YES_NO_OPTION);

    if(confirma==JOptionPane.YES_OPTION) {
        String sql = "delete from tbusuarios where iduser=?";

        try {
            pst=conexao.prepareStatement(sql);
            pst.setString(1, txtUsuld.getText());
            int apagado = pst.executeUpdate();

            if(apagado > 0) {
                JOptionPane.showMessageDialog(null, "Usuário removido com sucesso !");

                txtUsuld.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        }
    }
}

```

```

    }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
}
}

```

Depois de criado todos os códigos dos botões:

Clicar com o botão direito no botão Adicionar => Events => Action = actionPerformed , e incluir adicionar();

Clicar com o botão direito no botão Consultar => Events => Action = actionPerformed , e incluir consultar();

Clicar com o botão direito no botão Alterar => Events => Action = actionPerformed , e incluir alterar();

Clicar com o botão direito no botão Apagar => Events => Action = actionPerformed , e incluir remover();

=====

Dentro do menu Cadastro, temos o submenu Clientes, vamos precisar criar um JInternal frame com o nome TelaClientes

Clicar com o botão direito no pacote Telas => Outros => Swing GUI Forms => Form
JInternalFrame Form => Clicar em Next => Nome da classe: TelaClientes => Clicar em Finish

Em Propriedades:

Title: Clientes

PreferredSize: 640, 480

Closeable: marcar

Iconifiable: marcar

Maximizable: marcar

Em Código:

Form Size Policy: Generate Resize Code

Designer Size: 640, 480

Vamos associar a TelaClientes ao menu Cadastro => Clientes, clicar com o botão direito no menu Clientes => Events => Action => actionPerformed

Incluir o trecho abaixo no código

```
//Abre a Tela Clientes dentro do desktop
```

```
TelaClientes clientes = new TelaClientes();
```

```
clientes.setVisible(true);
```

```
desktop.add(clientes);
```

Vamos criar os objetos que serão necessários na TelaClientes

Na TelaClientes incluir os labels:

a) Procure o Nome

b) *Nome

c) Endereço

d) *Telefone

e) *Email

f) *Campos Obrigatórios

Na TelaClientes incluir as caixas de texto:

Alterar a variável da caixa de texto para txtCliPesquisar

Alterar a variável da tabela para tblClientes

Alterar a variável da caixa de texto para txtCliNome

Alterar a variável da caixa de texto para txtCliEndereco

Alterar a variável da caixa de texto para txtCliFone

Alterar a variável da caixa de texto para txtCliEmail

Na TelaClientes incluir 3 botões:

Primeiro botão, em propriedades:

Title: Adicionar

Alterar a variável da caixa de texto para btnAdicionar

Segundo botão, em propriedades:

Title: Alterar

Alterar a variável da caixa de texto para btnAlterar

Terceiro botão, em propriedades:

Title: Remover

Alterar a variável da caixa de texto para btnRemover

Em propriedades da TelaCliente:

Marcar as propriedades:

Closable

Iconifiable

Maximizable

Vamos criar um método para cadastrar os Clientes

```
//Método para cadastrar Clientes
```

```
private void adicionar(){
```

```
String sql = "insert into tbclientes (nomecli, endcli, fonecli, emailcli) values (?, ?, ?, ?)";
```

```
try {
```

```
pst=conexao.prepareStatement(sql);
```

```
pst.setString(1, txtCliNome.getText());
```

```
pst.setString(2, txtCliEndereco.getText());
```

```
pst.setString(3, txtCliFone.getText());
```

```
pst.setString(4, txtCliEmail.getText());
```

```

        //validação dos campos obrigatórios
        if((txtCliNome.getText().isEmpty()) || (txtCliFone.getText().isEmpty())) {
            JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");
        } else {
            int adicionado = pst.executeUpdate();
            if(adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Cliente adicionado com sucesso !");
                txtCliNome.setText(null);
                txtCliEndereco.setText(null);
                txtCliFone.setText(null);
                txtCliEmail.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Vamos criar um método para consultar os Clientes

```

//método para consultar clientes
public void consultar(){
    String sql = "select * from tbclientes where nomecli=?";
    try {
        pst=conexao.prepareStatement(sql);
        pst.setString(1, txtCliPesquisa.getText());
        rs=pst.executeQuery();
        if(rs.next()){
            txtCliId.setText(rs.getString(1));
            txtCliNome.setText(rs.getString(2));

```

```

        txtCliEndereco.setText(rs.getString(3));
        txtCliFone.setText(rs.getString(4));
        txtCliEmail.setText(rs.getString(5));
    } else {
        JOptionPane.showMessageDialog(null, "Cliente não cadastrado.");
        //as linhas abaixo limpam os campos
        txtCliId.setText(null);
        txtCliNome.setText(null);
        txtCliEndereco.setText(null);
        txtCliFone.setText(null);
        txtCliEmail.setText(null);
    }
} catch (Exception e){
    JOptionPane.showMessageDialog(null, e);
}
}

```

Vamos criar um método para consultar os Clientes e mostrar dentro da tabela os clientes cadastrados

```

//método para pesquisar clientes e exibir na tabela
private void pesquisar_cliente(){
    String sql = "select * from tblclientes where nomecli like ?";
    try {
        pst=conexao.prepareStatement(sql);
        pst.setString(1,txtCliPesquisa.getText()+"%");
        rs=pst.executeQuery();

        //tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
        DefaultTableModel modelo = (DefaultTableModel) tblClientes.getModel();
        modelo.setRowCount(0);
    }
}

```



```

        ResultSetMetaData metaData = rs.getMetaData();

        int colunas = metaData.getColumnCount();

        modelo.setColumnCount(0);

        for(int i = 1; i <= colunas; i++){

            modelo.addColumn(metaData.getColumnName(i));

        }

        while (rs.next()){

            Object[] linha = new Object[colunas];

            for(int i=1; i<=colunas; i++){

                linha[i-1]=rs.getObject(i);

            }

            modelo.addRow(linha);

        }

        rs.close();

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

```

Vamos criar um método para alterar os Clientes

```

//Método Alterar Clientes

private void alterar() {

    String sql = "update tbclientes set nomecli=?, endcli=?, fonecli=?, emailcli=? where nomecli=?";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtCliNome.getText());

        pst.setString(2, txtCliEndereco.getText());

        pst.setString(3, txtCliFone.getText());

        pst.setString(4, txtCliEmail.getText());

    }
}

```

```

        pst.setString(5, txtCliNome.getText());

        //validação dos campos obrigatórios
        if((txtCliNome.getText().isEmpty()) || (txtCliFone.getText().isEmpty()) ) {
            JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");
        } else {
            int adicionado = pst.executeUpdate();
            if(adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Dados do cliente alterados com sucesso!");
                txtCliId.setText(null);
                txtCliNome.setText(null);
                txtCliEndereco.setText(null);
                txtCliFone.setText(null);
                txtCliEmail.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Vamos criar um método para remover os Clientes

```

//Método Remover
private void remover() {
    int confirma=JOptionPane.showConfirmDialog(null, "Tem certeza que deseja remover o cliente ? ","Atenção", JOptionPane.YES_NO_OPTION);
    if(confirma==JOptionPane.YES_OPTION) {
        String sql = "delete from tbclientes where nomecli=?";
        try {
            pst=conexao.prepareStatement(sql);

```

```

        pst.setString(1, txtCliPesquisa.getText());

        int apagado = pst.executeUpdate();

        if(apagado > 0) {

            JOptionPane.showMessageDialog(null, "Cliente removido com sucesso !");

            txtCliPesquisa.setText(null);

            txtCliId.setText(null);

            txtCliNome.setText(null);

            txtCliEndereco.setText(null);

            txtCliFone.setText(null);

            txtCliEmail.setText(null);

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

}

}

```

=====

Vamos criar uma tela para criar as OSs a TelaOS

Clicar com o botão direito no pacote Telas => Outros => Swing GUI Forms => Form
 JInternalFrame Form => Clicar em Next => Nome da classe: TelaOs => Clicar em Finish

Em Propriedades:

Marcar as propriedades:

Title: OS

Closable

Iconifiable

Maximizable

Em Code:

Designer Size: 640 x 480

Incluir duas labels

Nº OS

Data

Incluir as caixas de texto para Nº OS e Data

Primeira caixa de texto

Alterar a variável para txtOs

Em propriedades

Editable: desmarcar

Segunda caixa de texto

Alterar a variável para txtData

Em propriedades

Editable: desmarcar

No código da TelaOs incluir os imports, a conexão e o construtor

```
import java.sql.*;
```

```
import com.mycompany.sistemacontrole.ModuloConexao.ModuloConexao;
```

```
import javax.swing.JOptionPane;
```

```
public class TelaOS extends javax.swing.JInternalFrame {
```

```
    Connection conexao = null;
```

```
    PreparedStatement pst = null;
```

```
    ResultSet rs = null;
```

```
    public TelaOS() {
```

```
        initComponents();
```

```
        conexao = ModuloConexao.conector();
```

```
    }
```

Criar um método para consultar o cliente

```
//Método Consultar

public void consultar_cliente(){

    //String sql = "select b.os, b.data_os, b.equipamento, b.defeito, b.servico, b.tecnico,
b.valor, a.idcli, a.nomecli, a.fonecli from tbclientes a, tbos b where nomecli=? and a.idcli =
b.idcli";

    String sql = "select a.idcli, a.nomecli, a.fonecli from tbclientes a where nomecli=?";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtCliPesquisa.getText());

        rs=pst.executeQuery();

        if(rs.next()){

            txtCliId.setText(rs.getString(1));

        } else {

            JOptionPane.showMessageDialog(null, "Cliente não possui ordem de serviço");

            //as linhas abaixo limpam os campos

            txtOs.setText(null);

            txtData.setText(null);

            //txtCliId.setText(null);

            txtOsEquip.setText(null);

            txtOsDef.setText(null);

            txtOsServ.setText(null);

            txtOsTec.setText(null);

            txtOsValor.setText(null);

            //txtCliPesquisa.setText(null);

        }

    } catch (Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

}
```

Criar um método para Adicionar OS

```
//Método Adicionar OS

private void adicionar_os(){

    String sql = "insert into tbos (equipamento, defeito, servico, tecnico, valor, idcli) values
    (?, ?, ?, ?, ?, ?)";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtOsEquip.getText());

        pst.setString(2, txtOsDef.getText());

        pst.setString(3, txtOsServ.getText());

        pst.setString(4, txtOsTec.getText());

        pst.setString(5, txtOsValor.getText());

        pst.setString(6, txtClild.getText());


        //validação dos campos obrigatórios

        if((txtClild.getText().isEmpty()) || (txtOsEquip.getText().isEmpty()) ||
        txtOsDef.getText().isEmpty() ) {

            JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");

        } else {

            int adicionado = pst.executeUpdate();

            if(adicionado > 0) {

                JOptionPane.showMessageDialog(null, "OS adicionada com sucesso !");

                txtOsEquip.setText(null);

                txtOsDef.setText(null);

                txtOsServ.setText(null);

                txtOsTec.setText(null);

                txtOsValor.setText(null);

                txtClild.setText(null);

                txtCliPesquisa.setText(null);

            }

        }

    }

}
```

```

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

```

Criar um método para Consultar OS

```

//Método Consultar OS

public void consultar_os(){

    String sql = "select a.os, a.data_os, a.equipamento, a.defeito, a.servico, a.tecnico, a.valor,
a.idcli, b.nomecli from tbos a, tbclientes b where os=? and a.idcli = b.idcli";

    try {

        pst=conexao.prepareStatement(sql);

        pst.setString(1, txtOs.getText());

        rs=pst.executeQuery();

        if(rs.next()){

            txtOs.setText(rs.getString(1));

            txtData.setText(rs.getString(2));

            txtOsEquip.setText(rs.getString(3));

            txtOsDef.setText(rs.getString(4));

            txtOsServ.setText(rs.getString(5));

            txtOsTec.setText(rs.getString(6));

            txtOsValor.setText(rs.getString(7));

            txtCliId.setText(rs.getString(8));

            txtCliPesquisa.setText(rs.getString(9));

        } else {

            JOptionPane.showMessageDialog(null, "OS não cadastrado.");

            //as linhas abaixo limpam os campos

            txtOs.setText(null);

            txtData.setText(null);

            txtOsEquip.setText(null);

            txtOsDef.setText(null);

```

```

        txtOsServ.setText(null);
        txtOsTec.setText(null);
        txtOsValor.setText(null);
        txtCliId.setText(null);
        txtCliPesquisa.setText(null);
    }
} catch (Exception e){
    JOptionPane.showMessageDialog(null, e);
}
}

```

Criar um método para Alterar OS

```

//Método Alterar OS
private void alterar_os() {
    String sql = "update tbos set equipamento=?, defeito=?, servico=?, tecnico=?, valor=?
where os=?";
    try {
        pst=conexao.prepareStatement(sql);
        pst.setString(1, txtOsEquip.getText());
        pst.setString(2, txtOsDef.getText());
        pst.setString(3, txtOsServ.getText());
        pst.setString(4, txtOsTec.getText());
        pst.setString(5, txtOsValor.getText());

        //validação dos campos obrigatórios
        if((txtOsEquip.getText().isEmpty()) || (txtCliId.getText().isEmpty()) ||
txtOsDef.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Preencher todos os campos obrigatórios");
        } else {
            int adicionado = pst.executeUpdate();
            if(adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Dados da OS alterados com sucesso !");
            }
        }
    }
}

```



```

        txtOs.setText(null);
        txtData.setText(null);
        txtOsEquip.setText(null);
        txtOsDef.setText(null);
        txtOsServ.setText(null);
        txtOsTec.setText(null);
        txtOsValor.setText(null);
        txtCliId.setText(null);
        txtCliPesquisa.setText(null);
    }
}
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}

```

Método para remover OS cadastrada

```

//Método Remover
private void remover_os() {
    int confirma=JOptionPane.showConfirmDialog(null, "Tem certeza que deseja remover a OS ? ", "Atenção", JOptionPane.YES_NO_OPTION);
    if(confirma==JOptionPane.YES_OPTION) {
        String sql = "delete from tbos where os=?";
        try {
            pst=conexao.prepareStatement(sql);
            pst.setString(1, txtOs.getText());
            int apagado = pst.executeUpdate();
            if(apagado > 0) {
                JOptionPane.showMessageDialog(null, "OS removida com sucesso !");
                txtOs.setText(null);
            }
        }
    }
}

```

```

        txtData.setText(null);
        txtCliId.setText(null);
        txtCliPesquisa.setText(null);
        txtOsEquip.setText(null);
        txtOsDef.setText(null);
        txtOsServ.setText(null);
        txtOsTec.setText(null);
        txtOsValor.setText(null);
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}
}
}

```

Atualizar tabela com os dados das OS cadastradas

```

//método para pesquisar OS
private void pesquisar_os(){
    String sql = "select * from tbos where os like ?";
    try {
        pst=conexao.prepareStatement(sql);
        pst.setString(1,txtCliPesquisa.getText()+"%");
        rs=pst.executeQuery();

        //tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
        DefaultTableModel modelo = (DefaultTableModel) tblOs.getModel();
        modelo.setRowCount(0);

        ResultSetMetaData metaData = rs.getMetaData();
        int colunas = metaData.getColumnCount();
    }
}

```

```

        modelo.setColumnCount(0);
        for(int i = 1;i <= colunas; i++){
            modelo.addColumn(metaData.getColumnName(i));
        }
        while (rs.next()){
            Object[] linha = new Object[colunas];
            for(int i=1;i<=colunas;i++){
                linha[i-1]=rs.getObject(i);
            }
            modelo.addRow(linha);
        }
        rs.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Criar um método para limpar a pesquisa

```

private void limpa_pesquisa(){
    txtOs.setText(null);
    txtData.setText(null);
    txtOsEquip.setText(null);
    txtOsDef.setText(null);
    txtOsServ.setText(null);
    txtOsTec.setText(null);
    txtOsValor.setText(null);
    txtCliId.setText(null);
    txtCliPesquisa.setText(null);
}

```

Vamos criar uma tela para exibir Relatório, então, dentro do package Telas, criar um JInternal Frame chamado TelaRelatorio

Incluir um botão

Alterar a variável para btnAtualizar

Incluir uma tabela

Alterar a variável para tblRelatorio

Segue o código para realizar a consulta do faturamento no banco de dados MySql

```
import java.sql.*;

import com.mycompany.sistemacontrole.ModuloConexao.ModuloConexao;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class TelaRelatorio extends javax.swing.JInternalFrame {

    Connection conexao = null;

    PreparedStatement pst = null;

    ResultSet rs = null;

    public TelaRelatorio() {

        initComponents();

        conexao = ModuloConexao.conector();

    }

    //método para consultar o faturamento

    private void consultar_faturamento(){

        String sql = "select substr(data_os,1,7) as Data, sum(valor) as Faturamento_Mensal from tbos group by data";

        try {

            pst=conexao.prepareStatement(sql);
```

```
rs=pst.executeQuery();
```

```
DefaultTableModel modelo = (DefaultTableModel) tblRelatorio.getModel();
```

```
modelo.setRowCount(0);
```

```
ResultSetMetaData metaData = rs.getMetaData();
```

```
int colunas = metaData.getColumnCount();
```

```
modelo.setColumnCount(0);
```

```
for(int i = 1;i <= colunas; i++){
```

```
    modelo.addColumn(metaData.getColumnName(i));
```

```
}
```

```
while (rs.next()){
```

```
    Object[] linha = new Object[colunas];
```

```
    for(int i=1;i<=colunas;i++){
```

```
        linha[i-1]=rs.getObject(i);
```

```
    }
```

```
    modelo.addRow(linha);
```

```
}
```

```
rs.close();
```

```
} catch (Exception e) {
```

```
    JOptionPane.showMessageDialog(null, e);
```

```
}
```

```
}
```