

# **GUI Graphical User Interface**

## **Swing**

### **CRUD (Create, Read, Update e Delete)**

Jose.wellington@uniceub.br

# Calendário

◀ agosto de 2013 ▶

D	S	T	Q	Q	S	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
-	-	-	-	-	-	-

◀ setembro de 2013 ▶

D	S	T	Q	Q	S	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

◀ outubro de 2013 ▶

D	S	T	Q	Q	S	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

◀ novembro de 2013 ▶

D	S	T	Q	Q	S	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

◀ dezembro de 2013 ▶

D	S	T	Q	Q	S	S
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

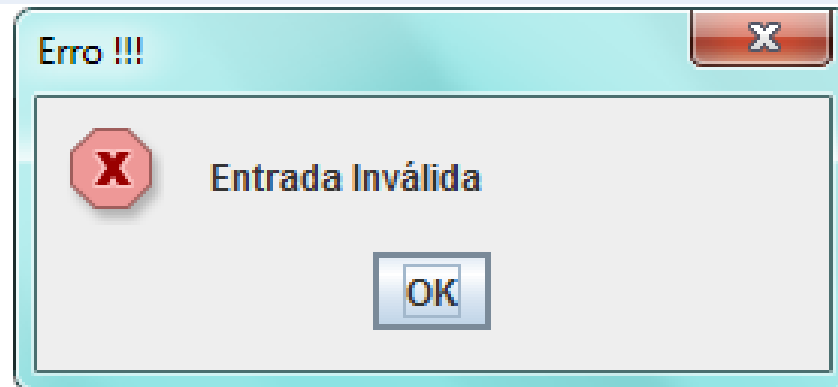
# Agenda

- **Revisão - GUI - Graphical User Interface**
  - Container
  - JOptionPane
  - JFrame
  - Layout Frame
  - Controles
  - Exercício
  - IDE – NetBeans
  - Navegação do Sistema
- **SQL Server**
  - Comandos - DDL
- **NetBeans - Biblioteca de acesso – SqlServer**
- **Comandos SQL (insert, update, delete, select)**
- **Exercício**

# JOptionPane

## JOptionPane – showMessageDialog - exemplo 1





















```
import javax.swing.*;  
public class Msg01 {  
  
    public static void main(String[] args) {  
  
        JOptionPane.showMessageDialog(null, "Entrada Inválida",  
            "Erro !!!", JOptionPane.ERROR_MESSAGE);  
    }  
}
```



# Controles

# Controles

## Controles javax.swing

 JLabel	 JTextPane	 JTextArea
 JTextField	 JEditorPane	 JFormattedTextField
 JComboBox	 JSpinner	 JPasswordField
 JButton	 JList	 JScrollBar
 JCheckBox	 JTable	 JSeparator
 JRadioButton	 JTree	 JSlider
 JToggleButton	 JProgressBar	

# IDE - NetBeans



# IDE - Integration Develop Enterprise

**NetBeans IDE é um AMBIENTE DE DESENVOLVIMENTO INTEGRADO (IDE)**

gratuito para desenvolvedores de software nas linguagens Java, C, C++, PHP, Ruby.

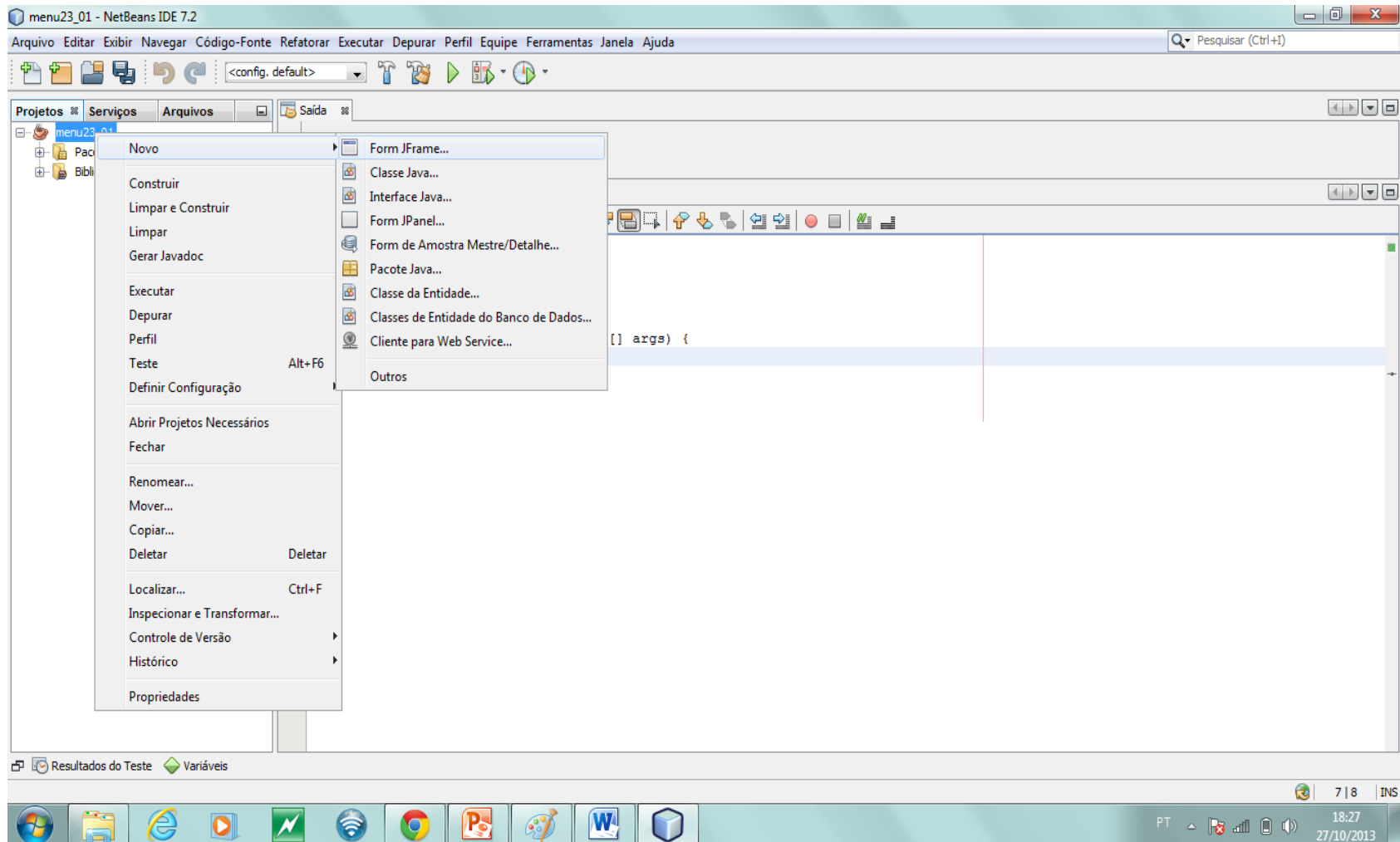


O IDE é executado em **muitas plataformas**, como Windows, Linux, Solaris e MacOS.

O NetBeans IDE oferece aos desenvolvedores ferramentas necessárias para **criar aplicativos** profissionais de desktop, empresariais, Web e móveis multiplataformas.

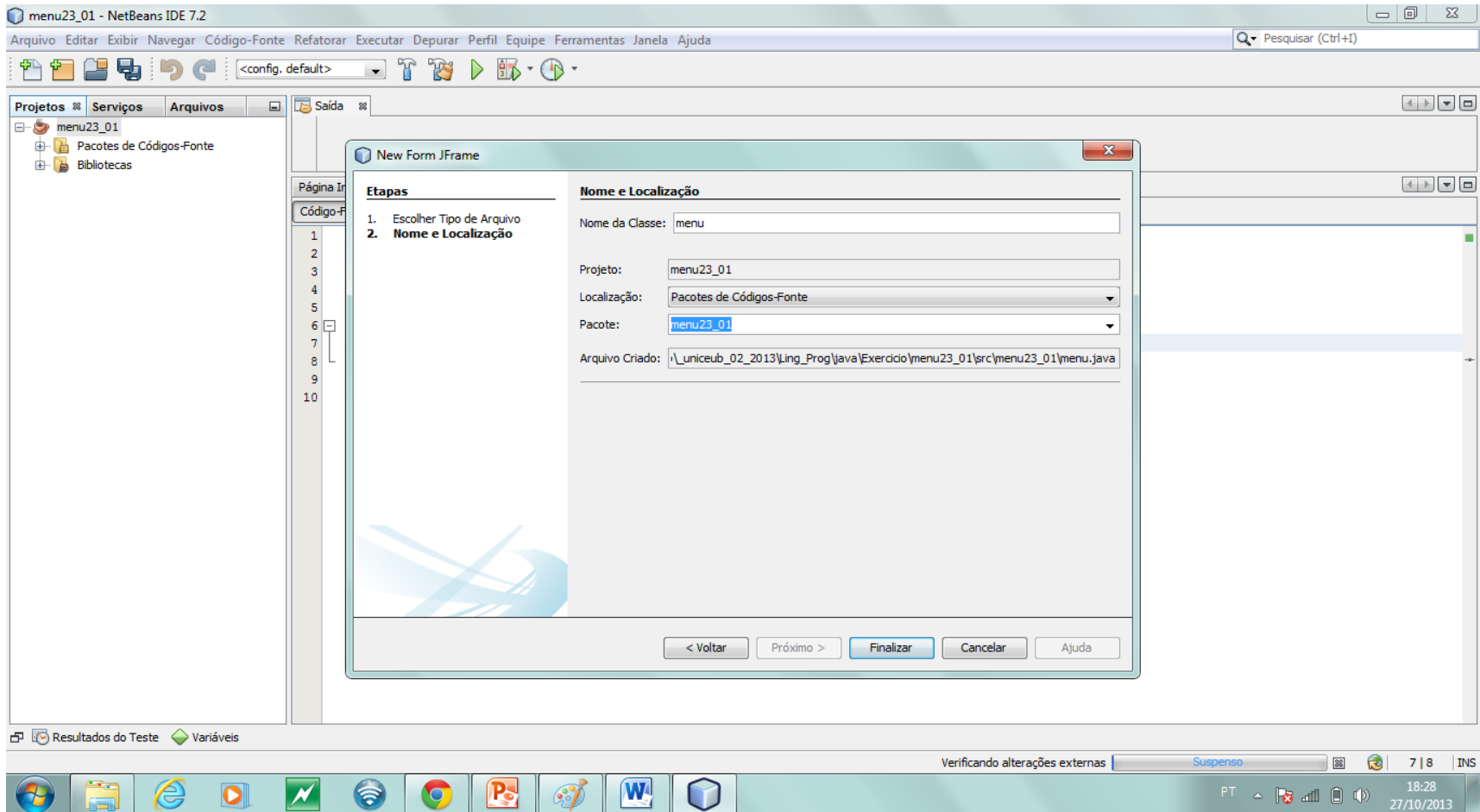
# Navegação do Sistema

## Criar formulário - Menu



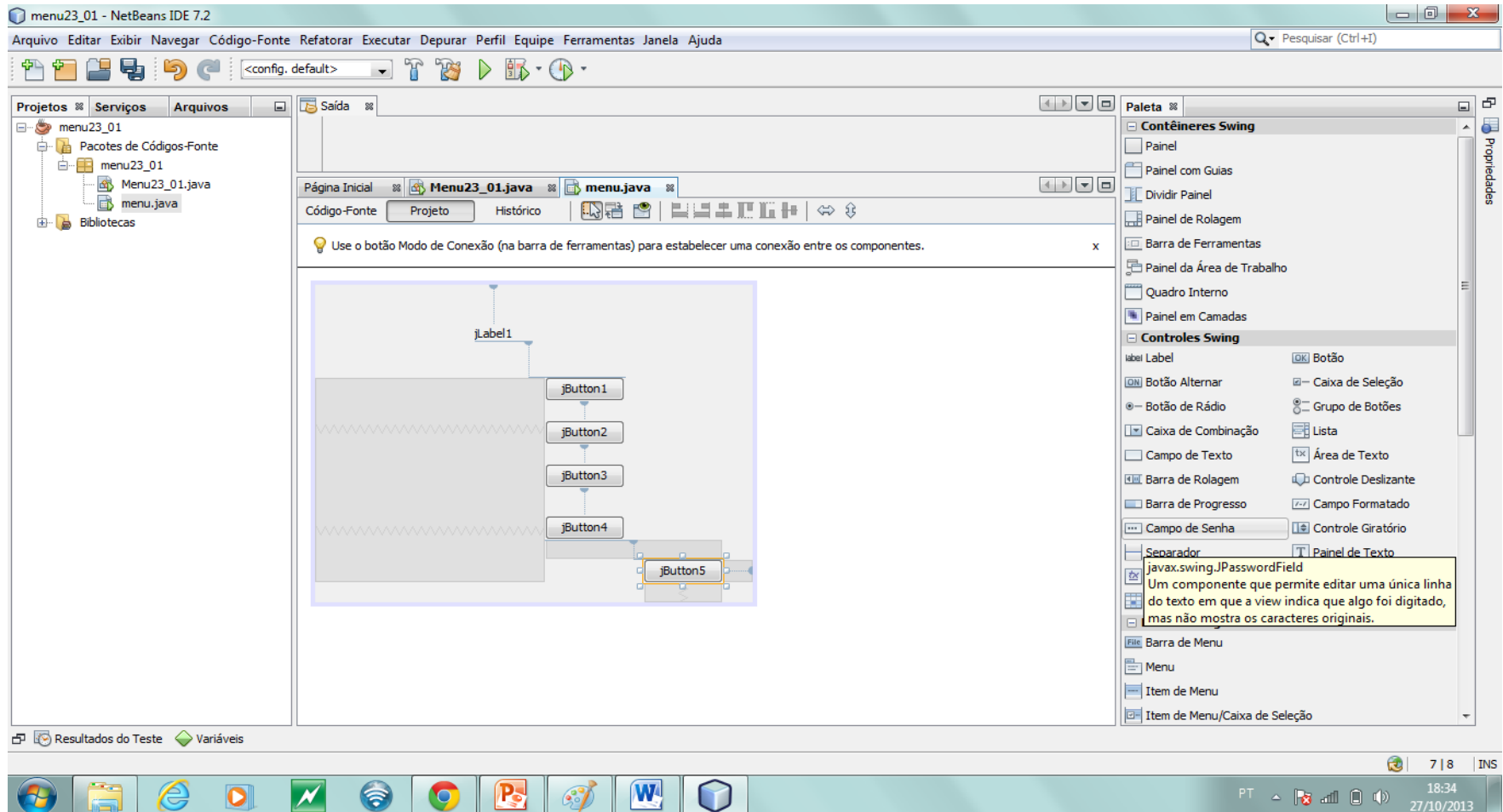
# Navegação do Sistema

## Criar formulário - Menu - NOME



# Navegação do Sistema

## Controles - Botões e Label



menu23\_01 - NetBeans IDE 7.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Pesquisar (Ctrl+I)

Projetos Serviços Arquivos

menu23\_01

Pacotes de Códigos-Fonte

menu23\_01

Menu23\_01.java

menu.java

Bibliotecas

Saída

Página Inicial Menu23\_01.java menu.java

Código-Fonte Projeto Histórico

Use o botão Modo de Conexão (na barra de ferramentas) para estabelecer uma conexão entre os componentes.

jLabel1

JButton1

JButton2

JButton3

JButton4

JButton5

**Paleta**

**Contêineres Swing**

- Painel
- Painel com Guias
- Dividir Painel
- Painel de Rolagem
- Barra de Ferramentas
- Painel da Área de Trabalho
- Quadro Interno
- Painel em Camadas

**Controles Swing**

- Label Label
- Botão
- Botão Alternar
- Caixa de Seleção
- Botão de Rádio
- Grupo de Botões
- Caixa de Combinação
- Lista
- Campo de Texto
- Área de Texto
- Barra de Rolagem
- Controle Deslizante
- Barra de Progresso
- Campo Formatado
- Campo de Senha
- Controle Giratório
- Separador
- Painel de Texto
- javax.swing.JPasswordField
- Um componente que permite editar uma única linha do texto em que a view indica que algo foi digitado, mas não mostra os caracteres originais.
- Barra de Menu
- Menu
- Item de Menu
- Item de Menu/Caixa de Seleção

Resultados do Teste Variáveis

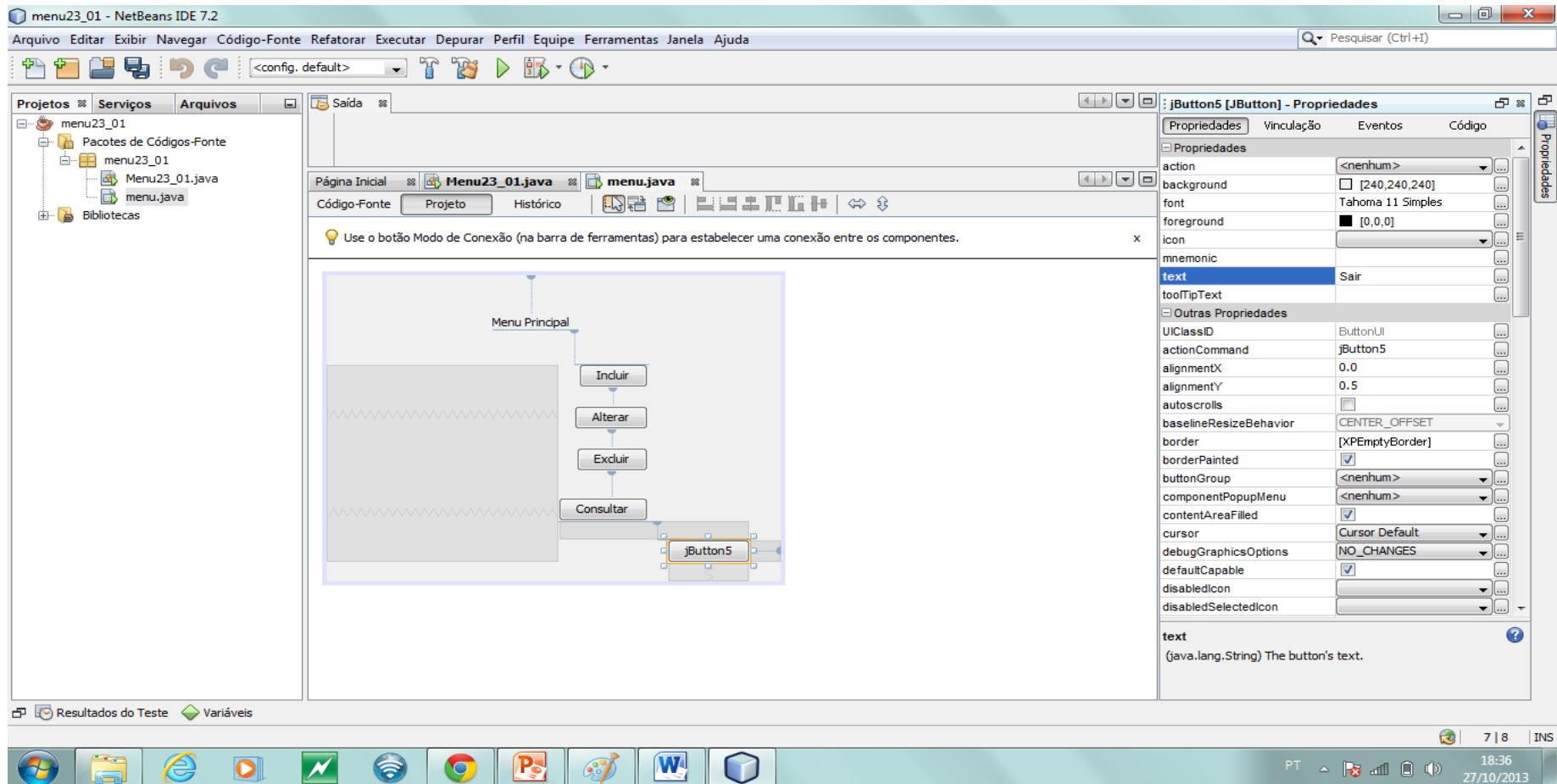
7 | 8 INS

18:34

27/10/2013

# Navegação do Sistema

## Propriedade dos Objetos (Botões, Label)



menu23\_01 - NetBeans IDE 7.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projeto: menu23\_01

Pacotes de Código-Fonte: menu23\_01

Arquivos: Menu23\_01.java, menu.java

Bibliotecas

Página Inicial: Menu23\_01.java, menu.java

Código-Fonte Projeto Histórico

Use o botão Modo de Conexão (na barra de ferramentas) para estabelecer uma conexão entre os componentes.

Menu Principal

Incluir

Alterar

Excluir

Consultar

jButton5

jButton5 [JButton] - Propriedades

Propriedades	Vinculação	Eventos	Código
action	<nenhum>		
background	[240,240,240]		
font	Tahoma 11 Simples		
foreground	[0,0,0]		
icon			
mnemonic			
text	Sair		
toolTipText			
Outras Propriedades			
UIClassID	ButtonUI		
actionCommand	jButton5		
alignmentX	0.0		
alignmentY	0.5		
autoscrolls			
baselineResizeBehavior	CENTER_OFFSET		
border	[XPEmptyBorder]		
borderPainted	<input checked="" type="checkbox"/>		
buttonGroup	<nenhum>		
componentPopupMenu	<nenhum>		
contentAreaFilled	<input checked="" type="checkbox"/>		
cursor	Cursor Default		
debugGraphicsOptions	NO_CHANGES		
defaultCapable	<input checked="" type="checkbox"/>		
disabledIcon			
disabledSelectedIcon			
text (java.lang.String) The button's text.			

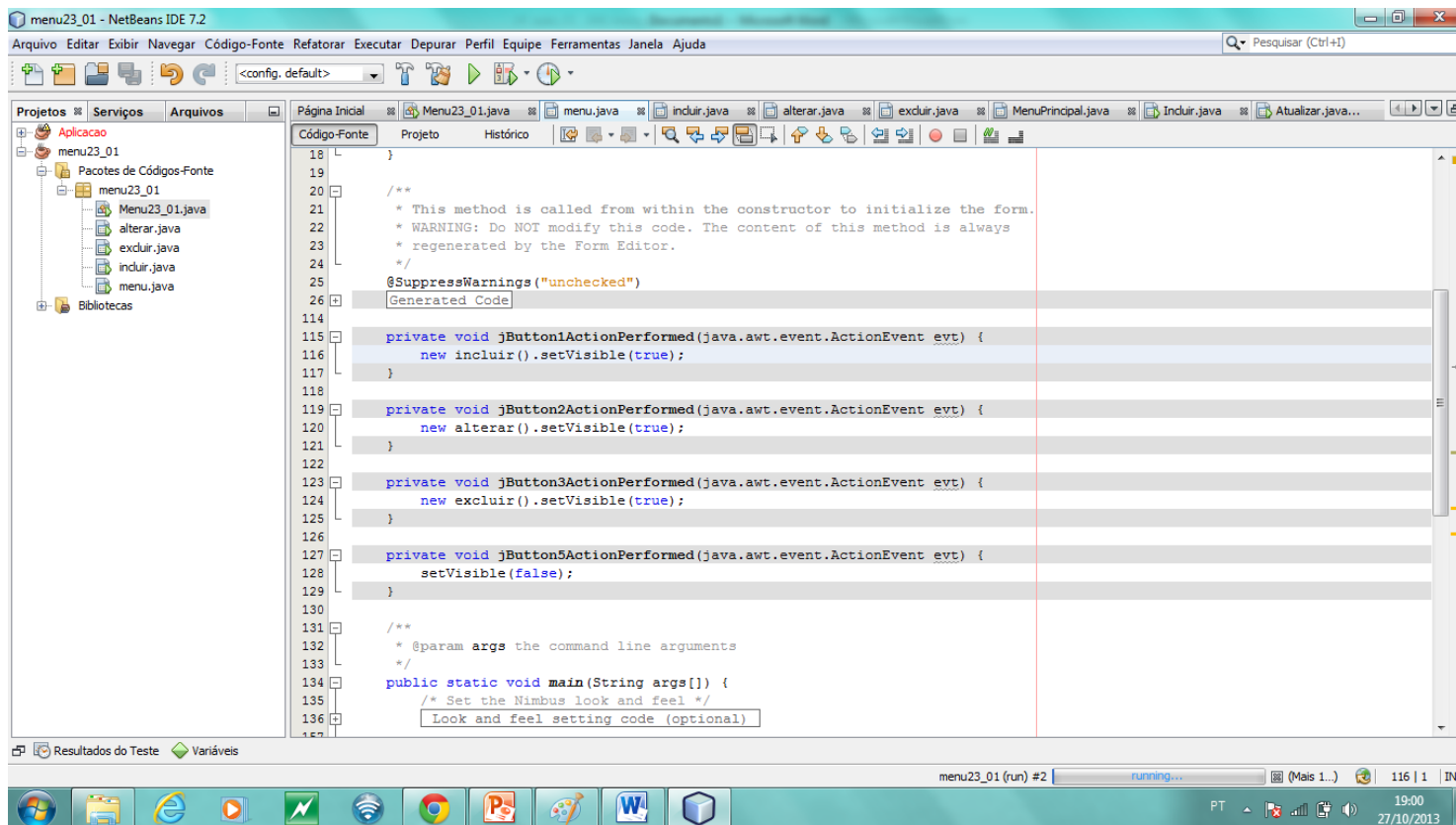
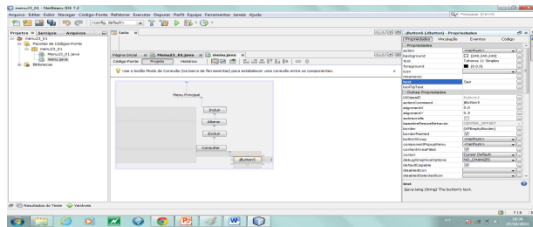
Resultados do Teste Variáveis

7 | 8 | INS

18:36  
27/10/2013

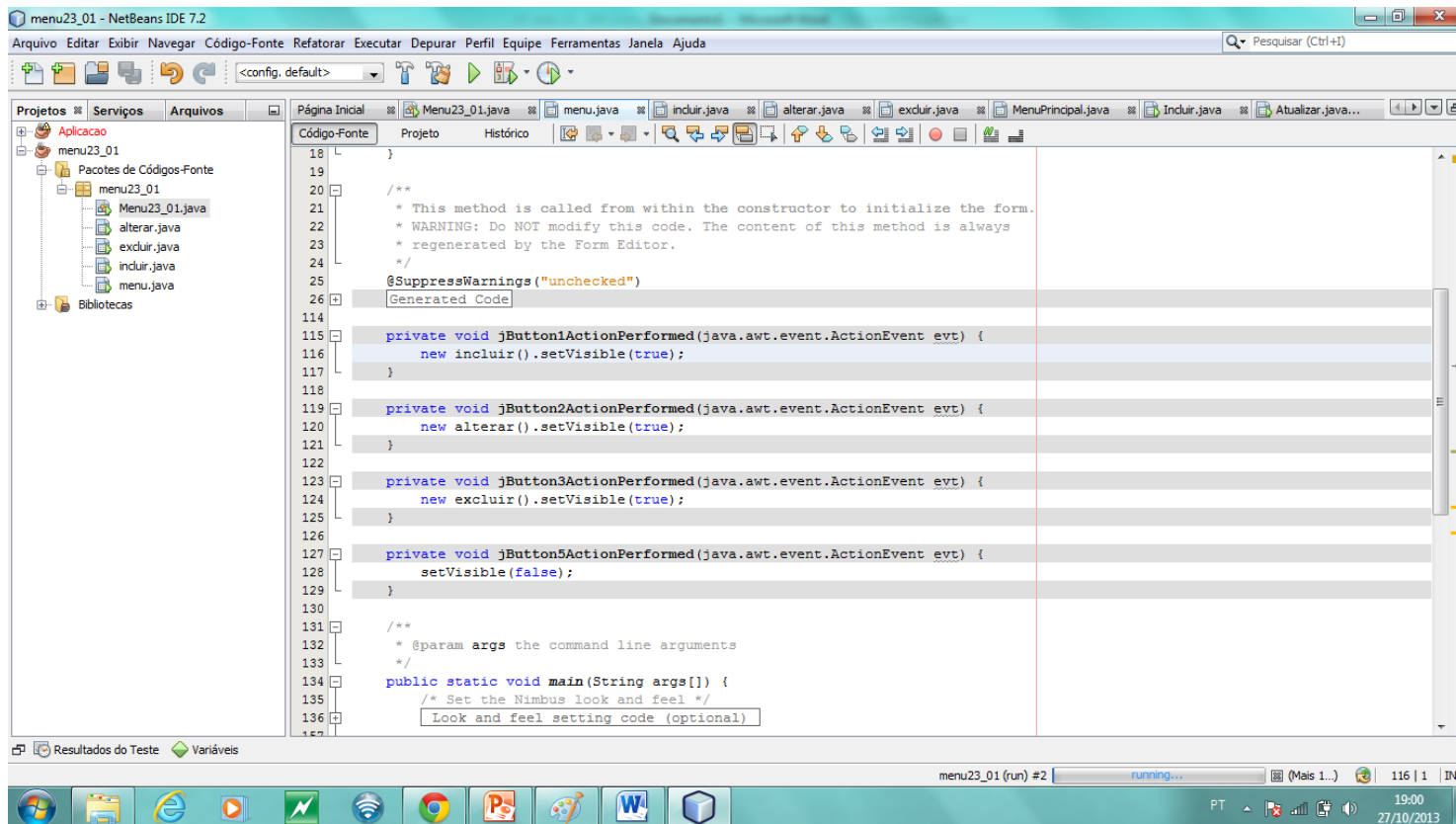
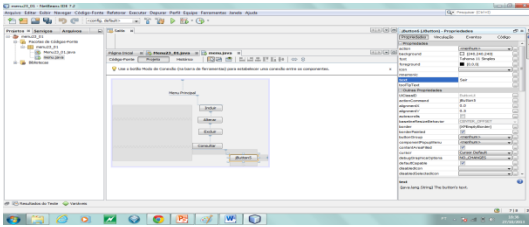
# Navegação do Sistema

## Executar os formulários (incluir, alterar, excluir, consultar)



# Navegação do Sistema

## Executar os formulários (incluir, alterar, excluir, consultar)





# Navegação do Sistema

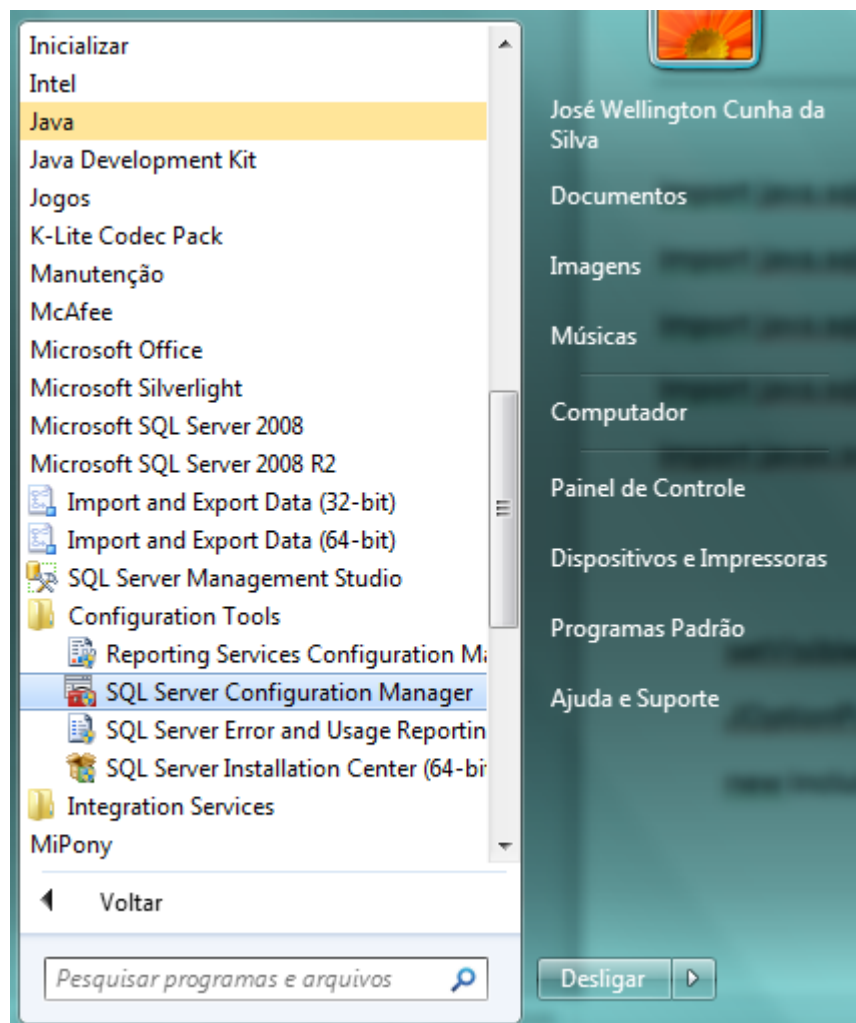
## Main

```
package menu23_01;  
  
public class Menu23_01 {  
  
    public static void main(String[] args) {  
  
        new menu().setVisible(true);  
    }  
}
```

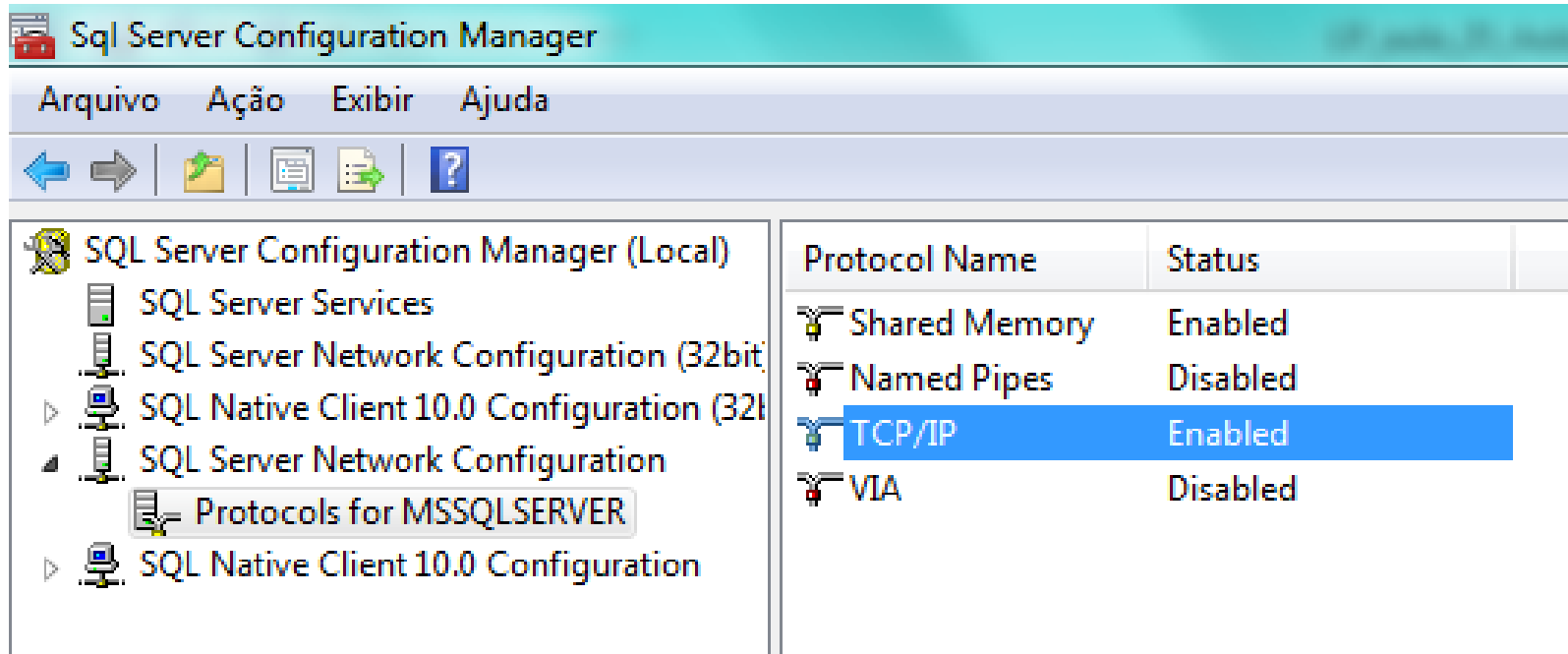


# SQL Server

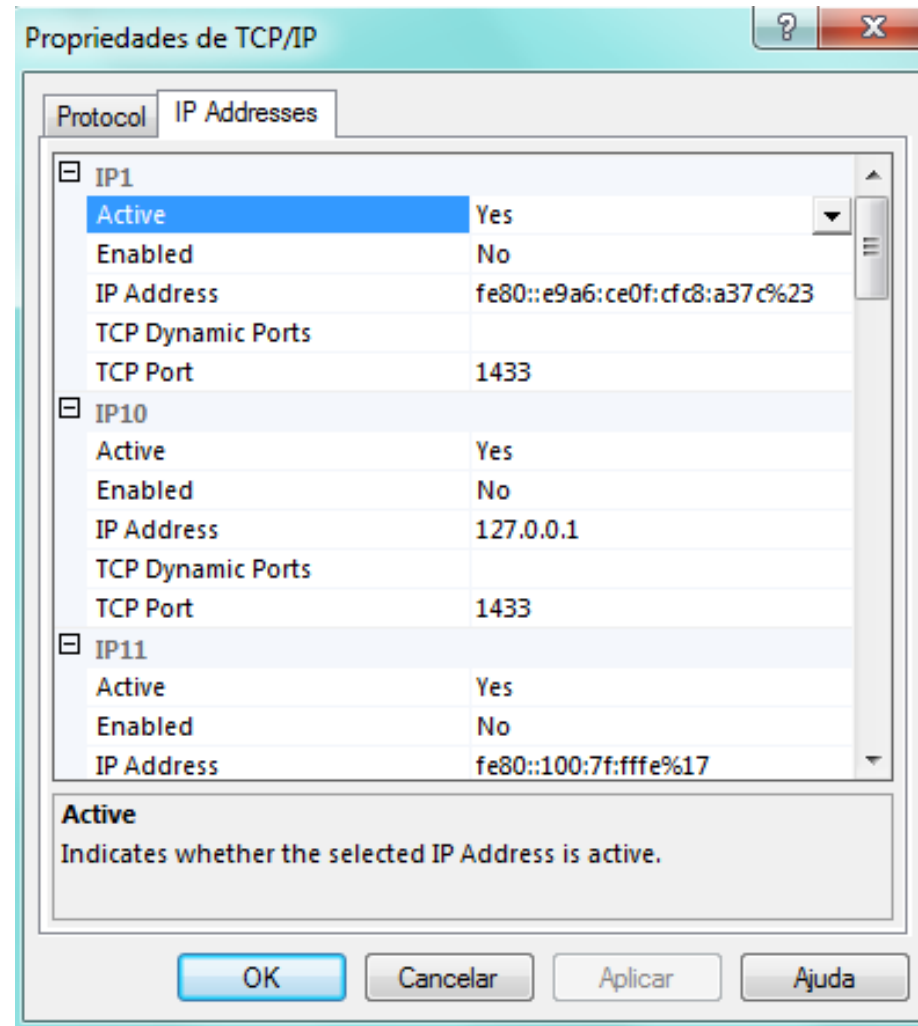
# Ativar - porta TCP/IP



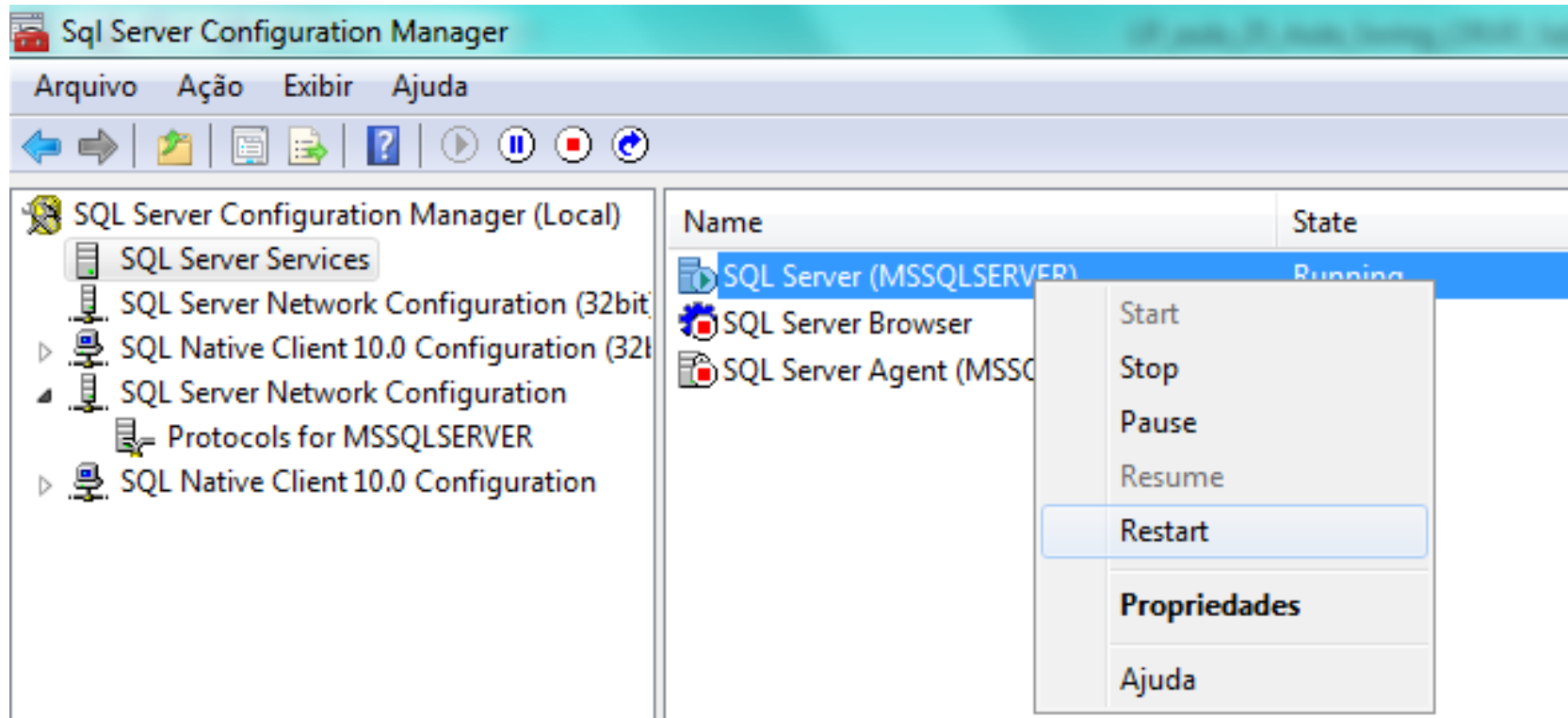
# SQL Server habilitar tcp/ip



# SQL Server habilitar tcp/ip



# SQL Server habilitar tcp/ip



# CRUD

```
// Localiza o driver utilizado
Class.forName("com.microsoft.sqlserver.jdbc.SQL ServerDriver").newInstance();
// conectar Banco
String url = "jdbc:sqlserver://localhost:1433;databaseName=bd_rh10;";
Connection con = DriverManager.getConnection(url, "sa","hu07pe21");
// Execução do banco
Statement stmt = (Statement) con.createStatement();

// As variaveis a,b e c recebendo os dados digitados nos campos nome,endereco e sexo.
String a = campoMatricula.getText();
String b= campoFuncionario.getText();
String c= campoSexo.getModel().getSelectedItem().toString();
String d;
if (c.equals("Masculino")) {d = "M";}
else { d = "F";}
String sql = "insert into tb_funcionario ( matricula,funcionario,cd_sexo) ";
    sql = sql + "values (" + a + "," + b + "," + d + ") ";

stmt.executeUpdate( sql ); //Executa o SQL
// Exibe a mensagem depois da gravação
JOptionPane.showMessageDialog( this, b+" foram incluidos com sucesso ");
//limpando campos
campoMatricula.setText("");
campoFuncionario.setText("");
con.close(); // fecha conexão do banco de dados.
```

## // ALTERAÇÃO

```
Class.forName("com.microsoft.sqlserver.jdbc.SQL ServerDriver").newInstance();

// conectar Banco

String url = "jdbc:sqlserver://localhost:1433;databaseName=bd_rh10;";

Connection con = DriverManager.getConnection(url, "sa", "hu07pe21");

// Execução do banco

Statement stmt = (Statement) con.createStatement();

// As variaveis a,b e c recebendo os dados digitados nos campos nome, endereco e sexo.

String a = campoMatricula.getText();

String b= campoFuncionario.getText();

String c= campoSexo.getModel().getSelectedItem().toString();

if (c.equals("Masculino")) {c = "M";}

else {c = "F";}

String sql = "UPDATE tb_funcionario set funcionario='"+b+"', cd_sexo='"+c+"

Sql = sql + " WHERE matricula='"+a;

stmt.executeUpdate( sql); //Executa o SQL

// Exibe a mensagem depois da gravação

JOptionPane.showMessageDialog( this, b+" foram Alterados com Sucesso");
```



## // EXCLUSÃO

// Inicio

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
```

// conectar Banco

```
String url = "jdbc:sqlserver://localhost:1433;databaseName=bd_rh10;";
```

```
Connection con = DriverManager.getConnection(url, "sa", "hu07pe21");
```

// Execução do banco

```
Statement stmt = (Statement) con.createStatement();
```

// As variaveis a,b e c recebendo os dados digitados nos campos nome,endereco e sexo.

```
String a = campoMatricula.getText();
```

// Comando para a exclusão do dados do cliente, pelo o codigo.

```
String sql = "delete from funcionario where matricula=" + a ;
```

```
int resultado = stmt.executeUpdate( sql ); //Executa o SQL
```

```
if (resultado != 0 ){
```

```
    JOptionPane.showMessageDialog( this, a +"" foram Alterados com Sucesso "); }
```

```
else { JOptionPane.showMessageDialog( this, "Matricula = " + a +"" não encontrada "); }
```

## // CONSULTA

### // Inicio

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
```

### // conectar Banco

```
String url = "jdbc:sqlserver://localhost:1433;databaseName=bd_rh10;";
```

```
Connection con = DriverManager.getConnection(url, "sa", "hu07pe21");
```

### // Execução do banco

```
Statement stmt = (Statement) con.createStatement();
```

// As variáveis a,b e c recebendo os dados digitados nos campos nome,endereco e sexo.

```
String a = campoMatricula.getText();
```

```
String sql = "Select * from tb_funcionario where matricula="+ a;
```

```
ResultSet RS =stmt.executeQuery(sql);
```

// Comando para exibir todos os dados do cliente.

```
while (RS.next()){
```

```
    String msg = "\n Matricula:" +RS.getString("matricula")+";
```

```
    msg = msg +  "\n Funcionario:" +RS.getString("funcionario");
```

```
    msg = msg +  "+" "\n Sexo:" +RS.getString("cd_sexo")""
```

```
    JOptionPane.showMessageDialog(null, msg ,JOptionPane.INFORMATION_MESSAGE); }
```

```
RS.close(); // fecha resultSet
```

```
con.close(); // fecha conexão do banco de dados.
```

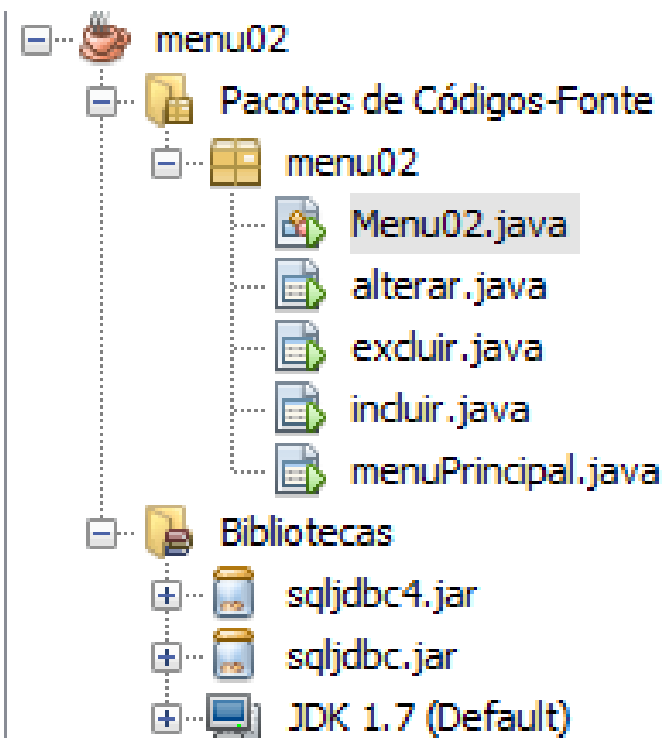
# NetBeans

## Biblioteca de acesso

### SQL Server

## Jar - Biblioteca de acesso

O driver de conexão com o banco de dados SQL Server tem uma biblioteca jar (**compactada**)





Try  
catch

## Exceção - try catch

```
try{  
    Comandos}  
  
catch( SQLException e){  
  
    JOptionPane.showMessageDialog(this,"Erro Cmdo SQL " + e.getMessage() ); }  
  
catch( ClassNotFoundException e){  
  
    JOptionPane.showMessageDialog( this, " Driver não encontrado "); }
```

# Exercício

# Exercício 25

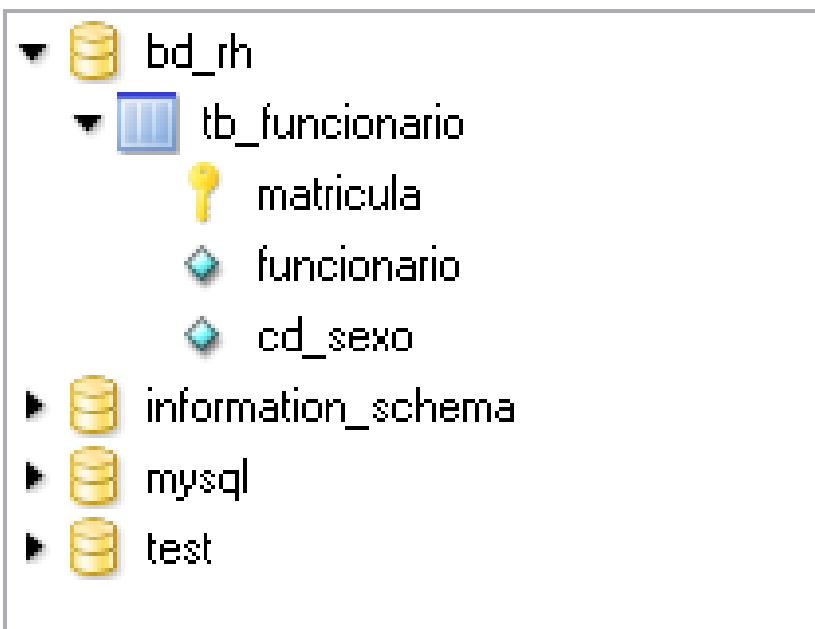
**Fazer CRUD – Create, Read, Update e Delete**

**No banco de dados bd\_rh10 – tb\_funcionario os Campos**

**matricula – int**

**funcionario – varchar(60)**

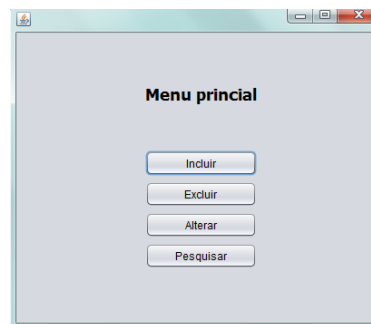
**cd\_sexo char(01)**





# Exercício 25

## Formulários



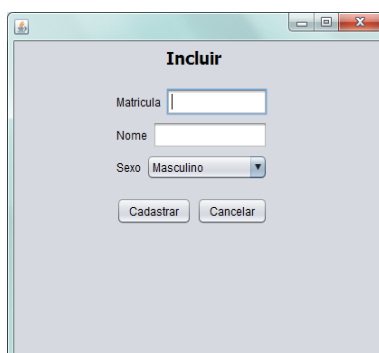
Menu principal

Incluir

Excluir

Alterar

Pesquisar



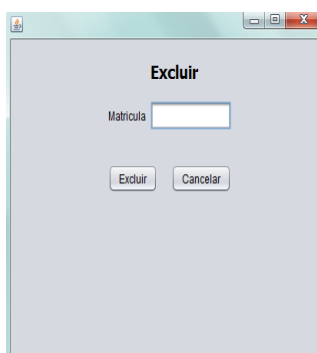
Incluir

Matricula

Nome

Sexo

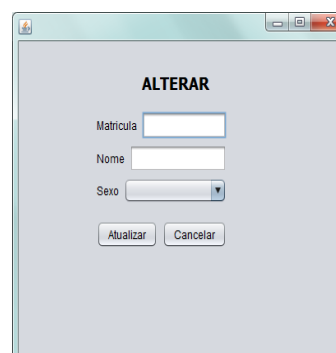
Cadastrar Cancelar



Excluir

Matricula

Excluir Cancelar



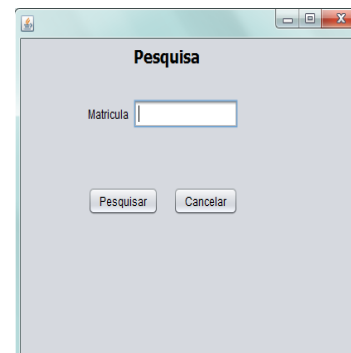
ALTERAR

Matricula

Nome

Sexo

Atualizar Cancelar



Pesquisa

Matricula

Pesquisar Cancelar

# Agenda

- **Revisão - GUI - Graphical User Interface**
  - Container
  - JOptionPane
  - JFrame
  - Layout Frame
  - Controles
  - Exercício
  - IDE – NetBeans
  - Navegação do Sistema
- **SQL Server**
  - Comandos - DDL
- **NetBeans - Biblioteca de acesso – SqlServer**
- **Comandos SQL (insert, update, delete, select)**
- **Exercício**