

# **CURSO JAVA COM MYSQL**

## **aulaEAD (Professor José de Assis)**

<https://www.youtube.com/watch?v=eA4WjzkK3c&list=PLbEOwbQR9lqxsTusvu8wfkUECrmcV81MU>

Resumo do curso feito por Roberto Pinheiro

## **INTRODUÇÃO**

Cliente: X

Assistência técnica de microcomputadores, notebooks e periféricos.



Quadro de funcionários:

- 4 técnicos
- 2 atendentes
- 1 gestor (que também é o dono da empresa)

Necessidade:

Sistema para controle de OS (Ordem de Serviços)

### Análise de Requisitos:

- Cadastro de clientes
- Emissão de OS
- Relatório de serviços

### **Análise de Requisitos:**

- **Cadastro de Cliente**
- **Emissão de OS**
- **Relatório de serviços**



### Restrições:

- Não pode haver nenhuma OS sem cliente vinculado;
- O sistema deve gerar automaticamente data e hora na emissão da OS;
- Somente o gestor pode ter acesso ao relatório de serviços.

### **Restrições:**

**Não pode existir nenhuma OS  
sem cliente vinculado**

**O sistema deverá gerar  
automaticamente data e hora  
na emissão da OS**

**Somente o gestor pode ter  
acesso ao relatório de serviços**

# BANCO DE DADOS

-- Criando o banco de dados

```
create database dbinfox;
```

-- Selecionando o banco de dados a ser utilizado

```
use dbinfox;
```

-- Criando a tabela de usuários do sistema

```
create table tbusuarios(
iduser int primary key,
usuario varchar(50) not null,
fone varchar(15),
login varchar(15) not null unique,
senha varchar(15) not null
);
```

-- Descrevendo a estrutura da tabela

```
describe tbusuarios;
```

-- Inserindo dados na tabela

```
insert into tbusuarios(iduser,usuario,fone,login,senha)
values (1,'José de Assis','9999-9999','joseassis','123456');
```

-- Exibindo os dados da tabela

```
select * from tbusuarios;
```

-- Inserindo mais dois usuários

```
insert into tbusuarios(iduser,usuario,fone,login,senha)
values (2,'Administrador','9999-9999','admin','admin');
insert into tbusuarios(iduser,usuario,fone,login,senha)
values (3,'Bill Gates','9999-9999','bill','123456');
```

-- Modificando dados na tabela

```
update tbusuarios set fone='8888-8888' where iduser=2;
```

-- Excluindo um registro da tabela

```
delete from tbusuarios where iduser=3';
```

-- Criando a tabela de clientes

```
create table tbclientes(
idcli int primary key auto_increment,
nomecli varchar(50) not null,
endcli varchar(100),
fonecli varchar(50) not null,
emailcli varchar(50)
);
```

```
-- Descrevendo a estrutura da tabela  
describe tbclientes;
```

```
-- Inserindo dados na tabela
```

```
insert into tbclientes(nomecli,endcli,fonecli,emailcli)  
values ('Linus Torvalds','Rua Tux,2015','9999-9999','linus@linux.com');
```

```
-- Exibindo os dados na tabela
```

```
select * from tbclientes;
```

```
-- Criando a tabela de ordem de serviços (OS)
```

```
use dbinfox;
```

```
create table tbos(  
os int primary key auto_increment,  
data_os timestamp default current_timestamp,  
equipamento varchar(150) not null,  
defeito varchar(150) not null,  
servico varchar(150),  
tecnico varchar(30),  
valor decimal(10,2),  
idcli int not null,  
foreign key (idcli) references tbclientes(idcli)  
);
```

```
-- Descrevendo a estrutura da tabela
```

```
describe tbos;
```

```
-- Inserindo dados na tabela
```

```
insert into tbos (equipamento,defeito,servico,tecnico,valor,idcli)  
values ('Notebook','Não liga','Troca da fonte','Zé',87.50,1);
```

```
-- Exibindo os dados da tabela
```

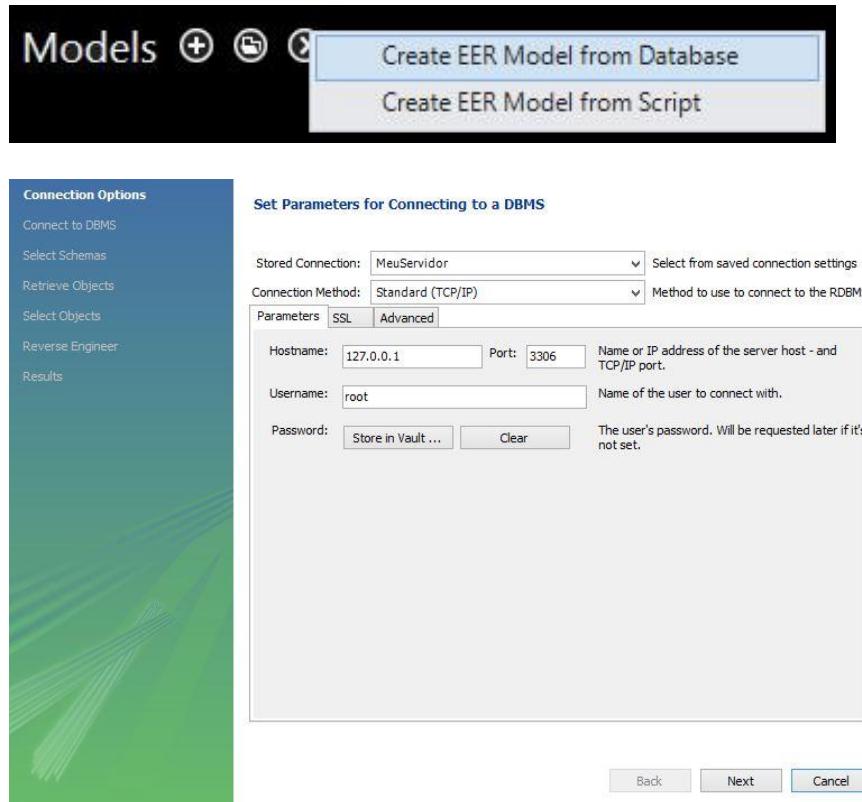
```
select * from tbos;
```

```
-- Relacionando a tabela de clientes com a tabela de ordem de serviços
```

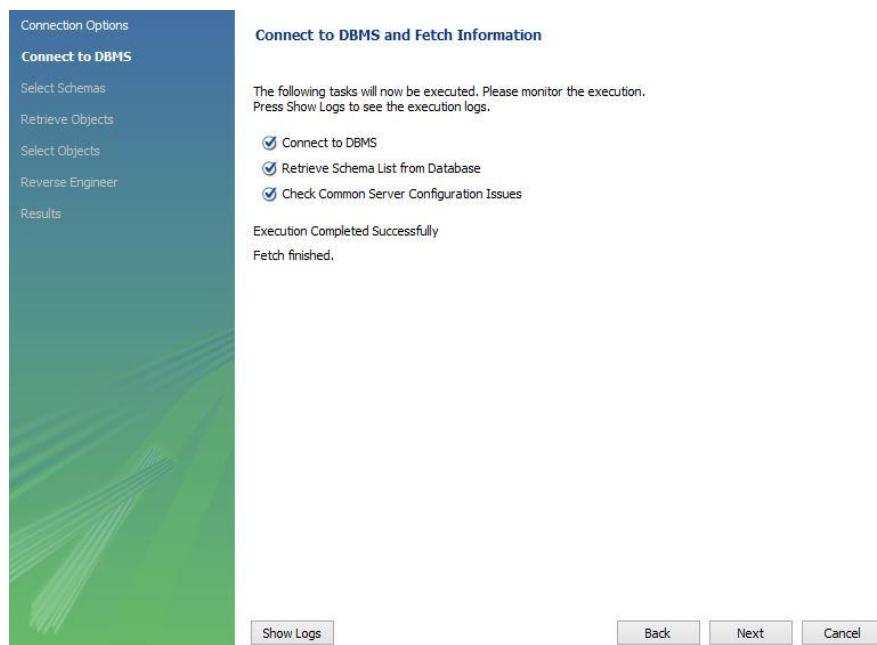
```
select  
O.os,equipamento,defeito,servico,valor,  
C.nomecli,fonecli  
from tbos as O  
inner join tbclientes as C  
on (C.idcli = O.idcli);
```

# DOCUMENTAÇÃO

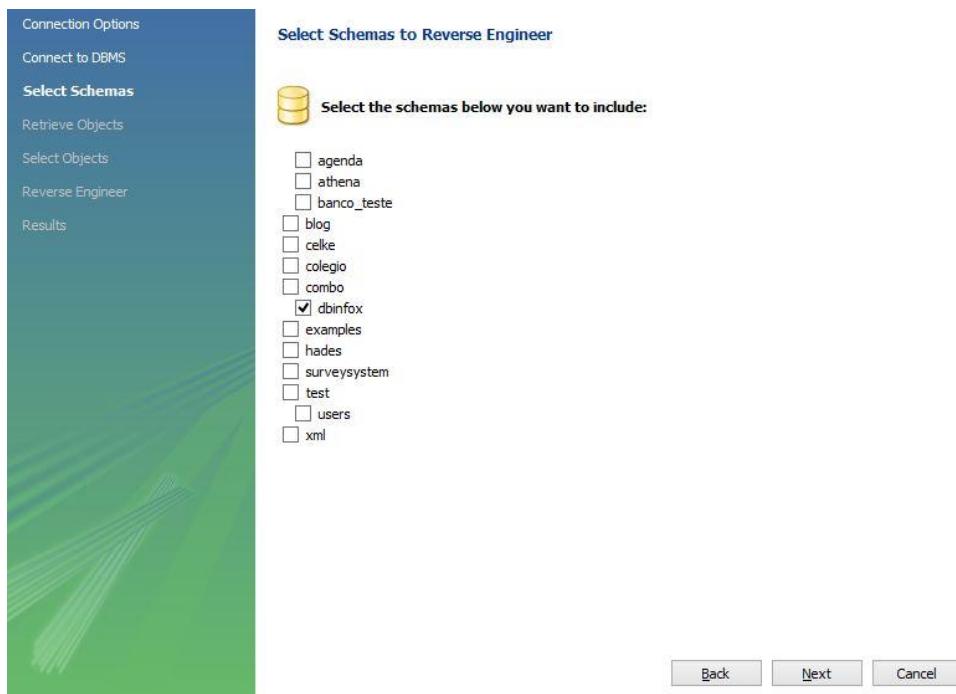
No MySQL Workbench, em Models, clicar no terceiro botão da esquerda para a direita e selecionar a opção "Create EER Model from Database".



Clique no botão "Next" para realizar a conexão.

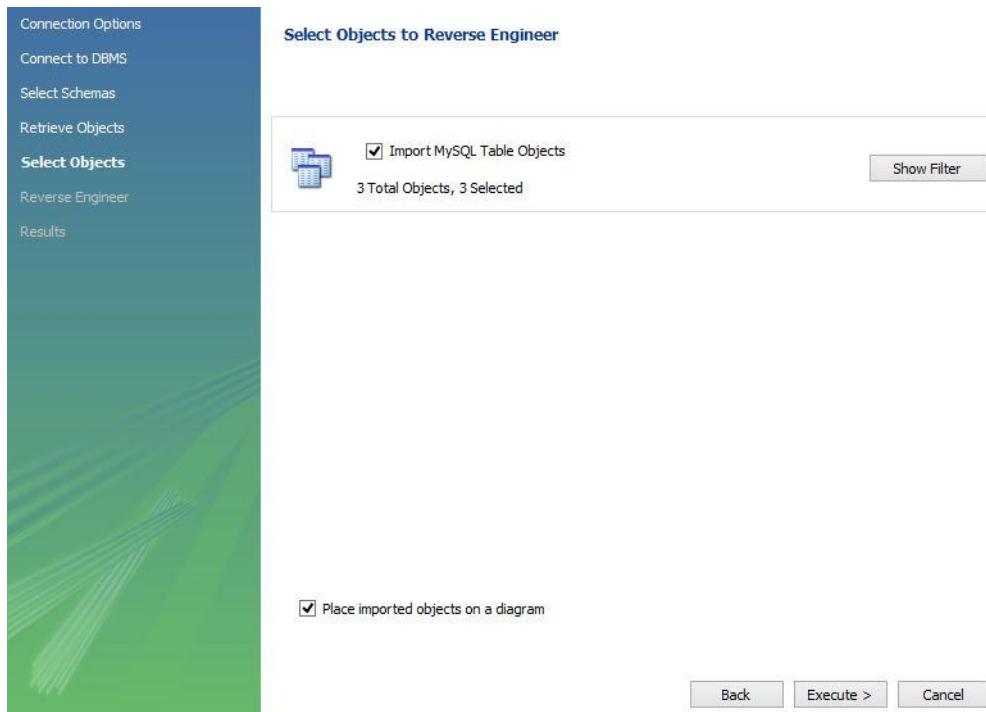


Clique novamente no botão "Next"



Selecione o banco de dados dbinfox e clique no botão "Next".

Em "Retrieve Objects" clique em "Next"

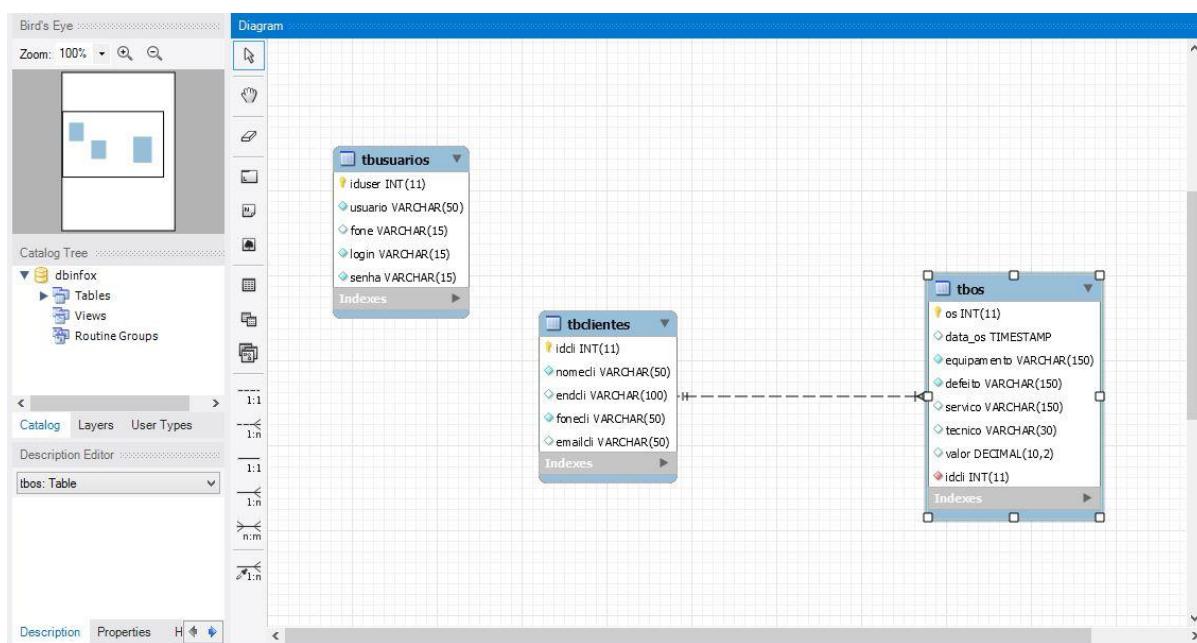


Clique no botão "Execute"

Em "Reverse Engineer" clique no botão "Next".

Em "Results" clique no botão "Finish"

# Diagrama Entidade-Relacionamento



Esse é o modelo Entidade-Relacionamento do Projeto.

- Salve o modelo como [mer\\_infox.mwb](#).
- Para exportar como um arquivo PDF, clique em **File → Export → Export as Single Page PDF** com o nome "[modeloER\\_infoX.pdf](#)".

## Script SQL do Banco de Dados

- Salve o script do banco de dados com o nome [infox.sql](#):  
**File → Save Script → Infox**

## Documentação das tabelas do banco de dados

Para fazer a documentação das tabelas do banco de dados, acesse o GitHub em <https://github.com/rsn86/MWB-DBDocPy> e baixe o plugin [MWB-DBDocPy](#).

Clique em **Scripting → Install Plugin/Module → DBDocPy\_grt** e em seguida no botão "Abrir" e reinicie o MySQL Workbench.

Abra o modelo criado.

Clique em **Tools → Catalog** e selecione **DBDoc As HTML File**

Salve o arquivo como **infoxDOC** e abra-o com o navegador:

The screenshot shows a web browser window with three tabs at the top: 'Menu', 'facebook - Pesquisa Google', and 'Data dictionary: dbinfox'. The main content area displays three tables:

**Table: tbclientes -**

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
idcli	INT(11)	Yes	Yes	No		
nomecli	VARCHAR(50)	Yes	No	No		
endcli	VARCHAR(100)	No	No	No	NULL	
fonecli	VARCHAR(50)	Yes	No	No		
emailcli	VARCHAR(50)	No	No	No	NULL	

**Table: tbos -**

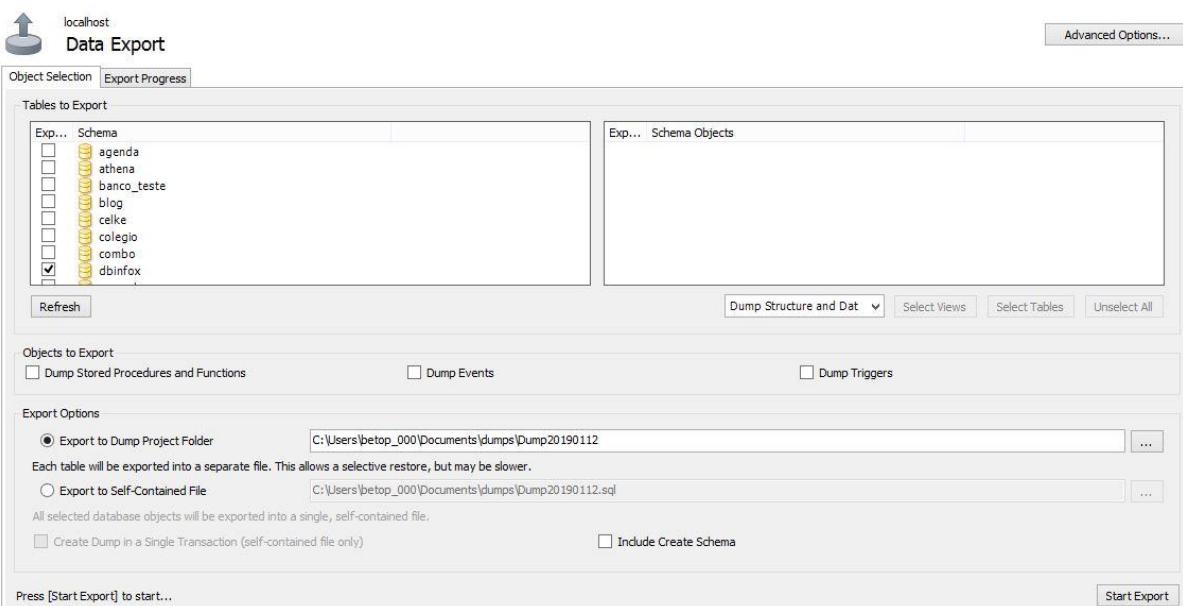
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
os	INT(11)	Yes	Yes	No		
data_os	TIMESTAMP	No	No	No	CURRENT_TIMESTAMP	
equipamento	VARCHAR(150)	Yes	No	No		
defeito	VARCHAR(150)	Yes	No	No		
servico	VARCHAR(150)	No	No	No	NULL	
technico	VARCHAR(30)	No	No	No	NULL	
valor	DECIMAL(10,2)	No	No	No	NULL	
idcli	INT(11)	Yes	No	Yes		

**Table: tbusuarios -**

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
iduser	INT(11)	Yes	Yes	No		
usuario	VARCHAR(50)	Yes	No	No		
fone	VARCHAR(15)	No	No	No	NULL	
login	VARCHAR(15)	Yes	No	No		
senha	VARCHAR(15)	Yes	No	No		

## Backup (dump) do banco de dados

No MySQL Workbench, na aba **Management** e clicar em "**Data Export**" e selecionar o banco de dados **dbinfox**. Em seguida, clique no botão "**Start Export**".



O backup será feito na pasta: **documentos/dumps/Dump20190112**, onde o número representa a data do backup.

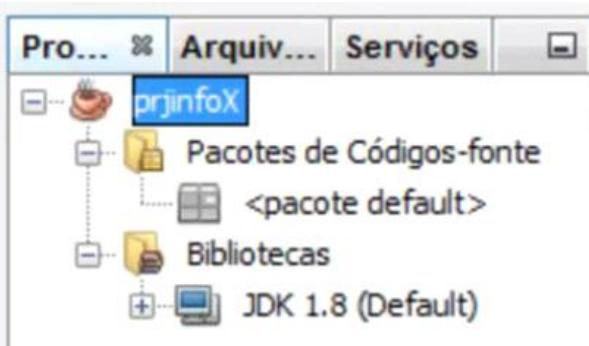


# CRIANDO O PROJETO NO NETBEANS

Arquivo → Novo Projeto → Java → Aplicação Java

Nome do Projeto: **prjinfoX**

Desmarcar a caixa: Criar classe principal



Criar 3 pacotes Java:



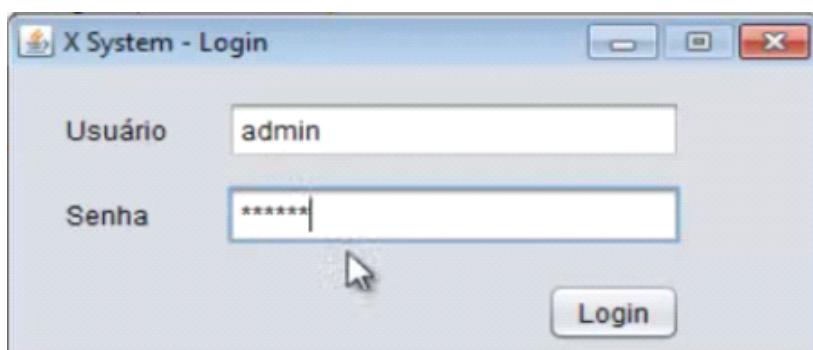
# TELA DE LOGIN

Clicar com o botão direito do mouse no pacote "br.com.infox.telas" e criar um Form JFrame:

Nome da Classe: TelaLogin



Inserir dois labels, um campo de texto, um campo de senha e um botão para que a tela fique como mostrado abaixo:



Botão

Nome da variável: btnLogin

Campo de texto

Nome da variável: txtUsuario

Campo de senha

Nome da variável: txtSenha

- Clicar com o botão direito do mouse sobre a tela de login. Selecionar a opção "**Propriedades**".

- Em "**Title**" digitar "**X System - Login**"

- Desmarcar a caixa "**Resizable**".

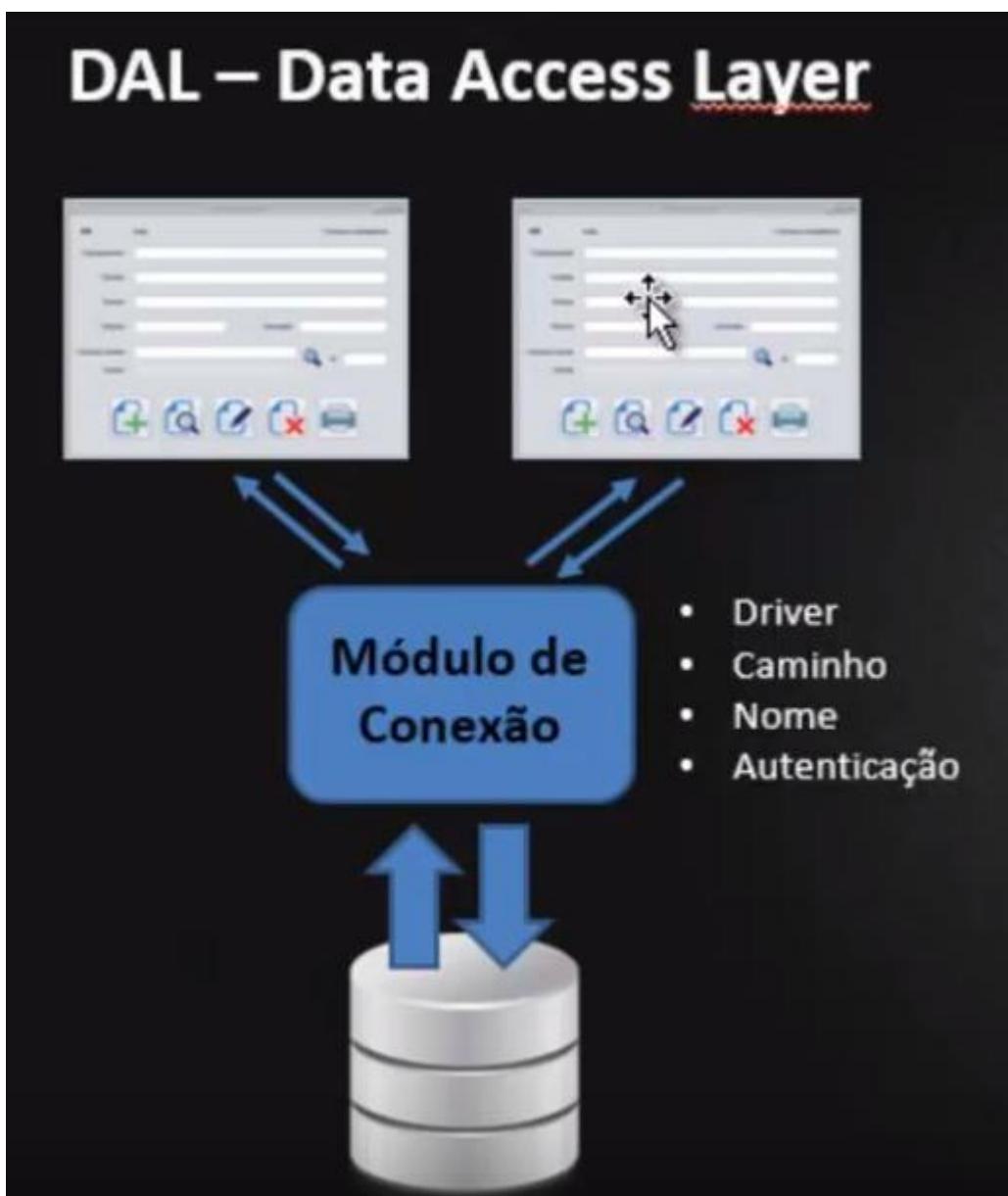
- Na aba "**Código**" marcar a caixa "**Gerar centralizado**"

- Política de tamanho de Form → **Gerar código de redimensionamento**

# CONEXÃO DO JAVA COM O BANCO DE DADOS

## Data Access Layer (DAL)

- Driver
- Caminho
- Nome
- Autenticação



# Driver para MySQL

(Biblioteca para conexão do Java com o banco de dados)

Acessar <https://www.mysql.com/> → Downloads → MySQL Community Edition (GPL) → Community (GPL) Downloads → MySQL Connectors → **Connector/J**

The screenshot shows the MySQL Downloads page. The navigation bar includes links for MySQL.COM, DOWNLOADS (which is highlighted), DOCUMENTATION, and DEVELOPER ZONE. Below the navigation bar, there are tabs for Enterprise, Community (which is selected), Yum Repository, APT Repository, SUSE Repository, Windows, and Archives. The main content area is titled "MySQL Connectors". It lists several connectors: Connector/ODBC, Connector/.NET, Connector/J (highlighted in blue), Connector/Python, and New! Connector/Node.js. A sidebar on the right contains a message about MySQL open source software being provided under the GPL License, and another message for OEMs, ISVs, and VARs about purchasing commercial licenses.

Selecionar:

Platform Independent (Architecture Independent), ZIP Archive

No thanks, just start my download.

E fazer o download do arquivo:

[mysql-connector-java-5.1.40.zip](#)

Descompactá-lo na área de trabalho. Será descompactada a pasta:

[mysql-connector-java-5.1.40](#)

Dentro dela há o arquivo:

[mysql-connector-java-5.1.40-bin.jar](#)

- No pacote "Bibliotecas" clicar com o botão direito do mouse e selecionar "Adicionar JAR/Pasta" e adicionar o arquivo "[mysql-connector-java-5.1.40-bin.jar](#)".

- Criar uma classe Java no pacote "[br.com.infox.dal](#)"

Nome da classe: ModuloConexao

Deve ter o seguinte conteúdo:

```
package br.com.infox.dal;
import java.sql.*;
public class ModuloConexao {
    // Método responsável por estabelecer a conexão com o banco de dados
    public static Connection conector() {
        java.sql.Connection conexao = null;
        // Chamando o driver de conexão
        String driver = "com.mysql.jdbc.Driver";
        // Armazenando informações referentes ao banco
        String url = "jdbc:mysql://localhost:3306/dbinfox";
        String user = "root";
        String password = "";
        // Estabelecendo a conexão com o banco de dados
        try {
            Class.forName(driver);
            conexao = DriverManager.getConnection(url,user,password);
            return conexao;
        } catch (Exception e) {
            // a linha abaixo serve de apoio para esclarecer o erro
            // System.out.println(e);
            return null;
        }
    }
}
```

- Inserir um novo label na tela de Login.

Nome da variável: lblStatus

- Baixar os ícones **dbok.png** e **dberror.png** em [www.iconfinder.com](http://www.iconfinder.com) (procurar por database).
- Importar os arquivos **dbok.png** e **dberror.png** para o pacote "**br.com.infox.icones**".
- No arquivo TelaLogin.Java acessar a aba "**Código-Fonte**" e acrescentar o seguinte conteúdo:

```
package br.com.infox.telas;
import java.sql.*;
import br.com.infox.dal.ModuloConexao;

public class TelaLogin extends javax.swing.JFrame {
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;
```

```

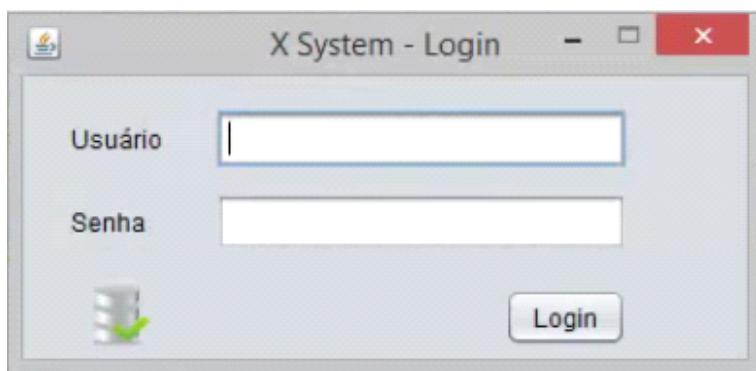
public TelaLogin() {
    initComponents();
    conexao = ModuloConexao.conector();
    // A linha abaixo serve de apoio ao status da conexão
    // System.out.println(conexao);
    if (conexao != null) {
        lblStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/br/com/infox/icones/dbok.png")));
    } else {
        lblStatus.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/br/com/infox/icones/dberror.png")));
    }
}

```

Tela de Login - sem conexão com o banco de dados:



Tela de Login - há conexão com o banco dados;



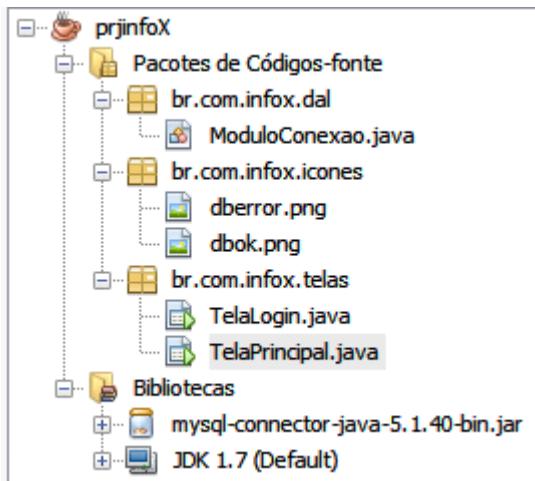
String de conexão:



# LOGANDO NO SISTEMA

- Clicar com o botão direito do mouse no pacote "br.com.infox.telas" e criar um Form JFrame:

Nome da Classe: TelaPrincipal



- Inserir as seguintes linhas no Código Fonte da Tela de Login:

```
public class TelaLogin extends javax.swing.JFrame {
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public void logar(){
        String sql = "select * from tbusuarios where login=? and senha=?";
        try {
            // Preparando a consulta ao banco em função do que foi digitado
            // nas caixas de texto. O ? é substituído pelo conteúdo das variáveis
            pst = conexao.prepareStatement(sql);
            pst.setString(1, txtUsuario.getText());
            pst.setString(2, txtSenha.getText());
            // Executando a query
            rs = pst.executeQuery();
            // Se existir usuário e senha correspondente
            if (rs.next()){
                TelaPrincipal principal = new TelaPrincipal();
                principal.setVisible(true);
            } else {
                JOptionPane.showMessageDialog(null,"Usuário e/ou senha inválido(s)!");
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null,e);
        }
    }
}
```

- Clicar com o botão direito do mouse sobre o botão da Tela de login. Selecionar **Eventos → Action → actionPerformed** e inserir o seguinte conteúdo:

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // Chamando o método logar()  
    logar();  
}
```

- Entrar no MySQL Workbench, abrir o banco de dados dbinfox:

```
use dbinfox;
```

- Inserir mais um usuário na tabela tbusuarios:

```
insert into tbusuarios (iduser,usuario,fone,login,senha)  
values (3,'Leandro Ramos','9999-9999','leandro','123');
```

# CONSTRUINDO A TELA PRINCIPAL

## Fechando a tela de login ao ser exibida a tela principal

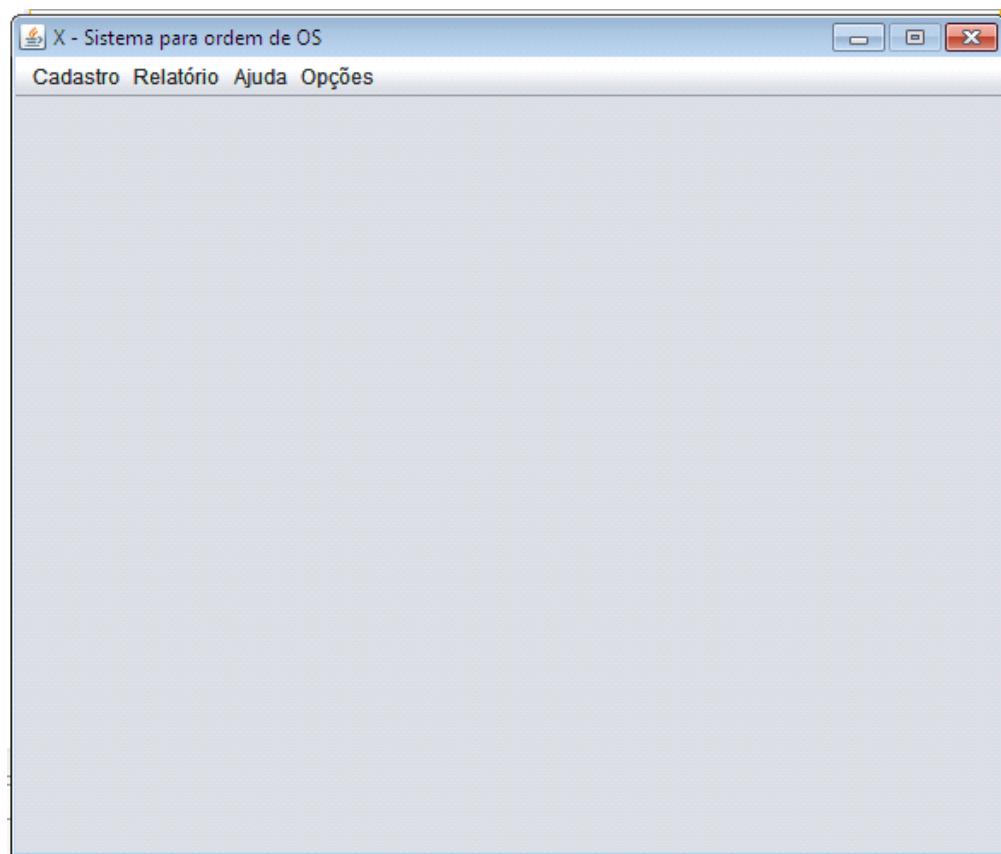
```
public class TelaLogin extends javax.swing.JFrame {  
    Connection conexao = null;  
    PreparedStatement pst = null;  
    ResultSet rs = null;  
  
    public void logar(){  
        String sql = "select * from tbusuarios where login=? and senha=?";  
        try {  
            // Preparando a consulta ao banco em função do que foi digitado  
            // nas caixas de texto. O ? é substituído pelo conteúdo das variáveis  
            pst = conexao.prepareStatement(sql);  
            pst.setString(1, txtUsuario.getText());  
            pst.setString(2, txtSenha.getText());  
            // Executando a query  
            rs = pst.executeQuery();  
            // Se existir usuário e senha correspondente  
            if (rs.next()){  
                TelaPrincipal principal = new TelaPrincipal();  
                principal.setVisible(true);  
                this.dispose();  
                conexao.close();  
            } else {  
                JOptionPane.showMessageDialog(null,"Usuário e/ou senha inválido(s)!");  
            }  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(null,e);  
        }  
    }  
}
```

## Configurando as propriedades da tela principal

- Clicar com o botão direito do mouse sobre a tela principal. Selecionar a opção "Propriedades".
- Em "Title" digitar "X - Sistema para ordem de OS"
- Desmarcar a caixa "Resizable".
- Na aba "Código" marcar a caixa "Gerar centralizado"
- Política de tamanho de Form → Gerar código de redimensionamento

## Adicionando uma barra de menu horizontal

- Na tela principal inserir uma "barra de menu" na parte superior da tela.
- Alterar a opção "File" por "Cadastro" e "Edit" por "Relatório".
- Clicar com o botão direito do mouse no menu e "adicionar menu". Alterar o texto do menu para "Ajuda".
- Clicar com o botão direito do mouse no menu e "adicionar menu". Alterar o texto do menu para "Opções".



## Adicionando sub-menus

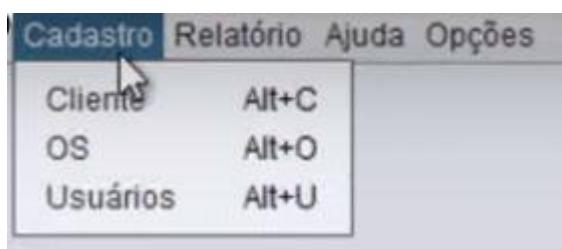
- Clicar com o botão direito do mouse sobre o item de menu "**Cadastro**" e selecionar a opção "Adicionar da paleta" → "Item de Menu".

"Editar texto" → Cliente

- Dar 2 cliques no quadrado existente no lado direito e em "**Toque de tecla**" usar como tecla de atalho a combinação "**Alt-C**".

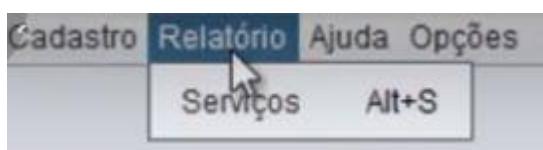
- Repetir o processo acima e criar o sub-menu "**OS**" e tecla de atalho "**Alt-O**".

- Repetir o processo acima e criar o sub-menu "**Usuários**" e tecla de atalho "**Alt-U**".



- Clicar com o botão direito do mouse sobre o item de menu "**Relatório**" e selecionar a opção: "Adicionar da paleta" → "Item de Menu".

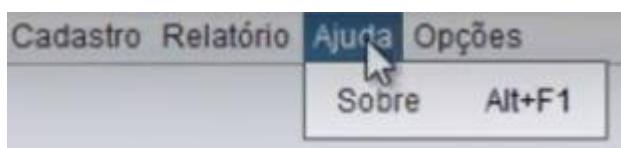
"Editar texto" → Serviços



- Dar 2 cliques no quadrado existente no lado direito e em "**Toque de tecla**" usar como tecla de atalho a combinação "**Alt-S**".

- Clicar com o botão direito do mouse sobre o item de menu "**Ajuda**" e selecionar a opção: "Adicionar da paleta" → "Item de Menu".

"Editar texto" → Sobre

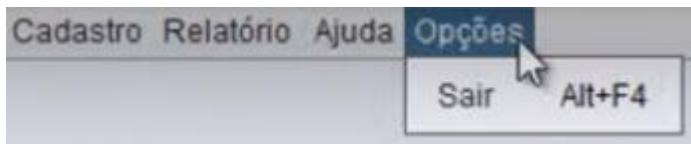


- Dar 2 cliques no quadrado existente no lado direito e em "**Toque de tecla**" usar como tecla de atalho a combinação "**Alt-F1**".

- Clicar com o botão direito do mouse sobre o item de menu "**Opções**" e selecionar a opção: "Adicionar da paleta" → "Item de Menu".

"**Editar texto**" → Sair

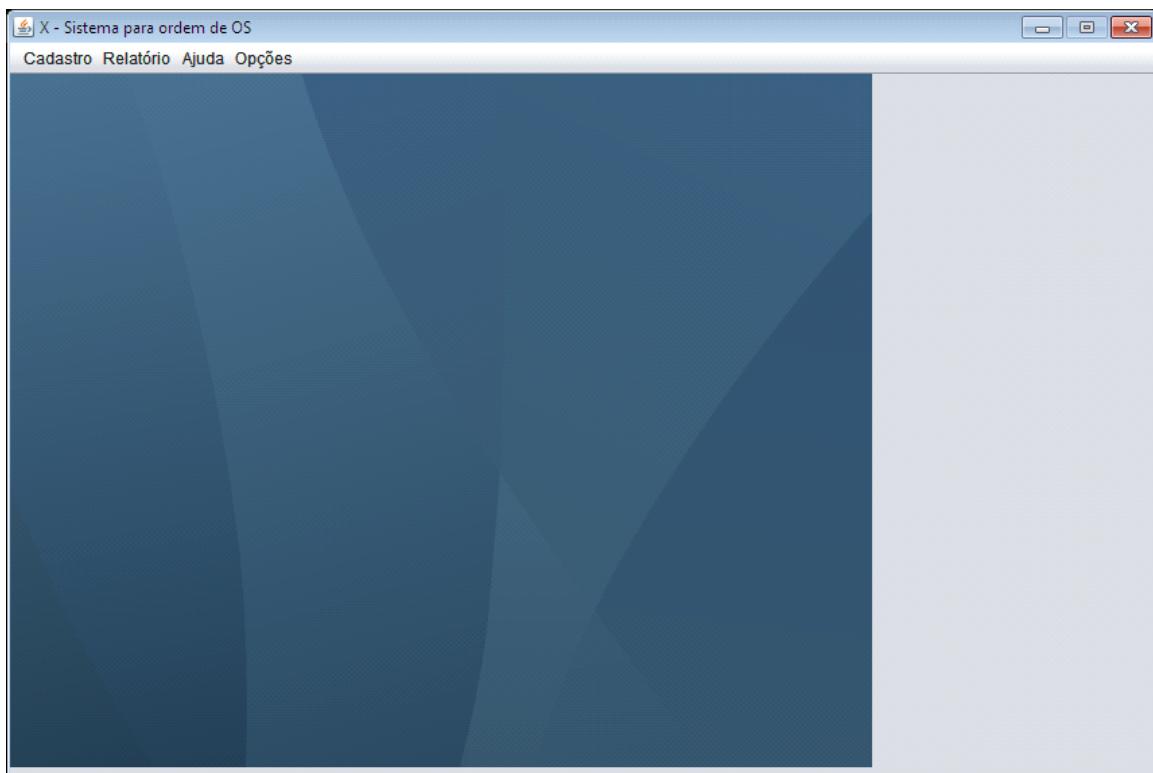
- Dar 2 cliques no quadrado existente no lado direito e em "**Toque de tecla**" usar como tecla de atalho a combinação "**Alt-F4**".



## Adicionando um Painel da Área de Trabalho

- Inserir, na Tela Principal, um "Painel da Área de Trabalho" (de 640 x 480 px)

Todos os formulários serão inseridos dentro desta área de trabalho.



- Inserir 3 labels do lado direito do painel da área de trabalho:

- 1 - Nome do usuário
- 2 - Data do sistema
- 3 - Ícone com o logo da empresa X

## **Nomeando os elementos da tela principal**

Label "Usuário" → lblUsuario  
Label "Data" → lblData  
Painel da área de trabalho → desktop  
Barra de Menu → menu

Menu "Cadastro" → menCad  
Submenu Cliente → menCadCli  
Submenu "OS" → menCadOs  
Submenu "Usuários" → menCadUsu

Menu "Relatório" → menRel  
Submenu "Serviços" → menRelSer

Menu "Ajuda" → menAju  
Submenu "Sobre" → menAjuSob

Menu "Opções" → MenOpc  
Submenu "Sair" → MenOpcSai

## **Exibindo a data do sistema**

- Clicar com o botão direito do mouse em qualquer parte do formulário. Selecionar **Eventos** → **Window** → **windowActivated**

Inserir o seguinte código:

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {  
    // As linhas abaixo substituem a label lblData pela data atual do sistema  
    // ao inicializar o formulário  
    Date data = new Date();  
    DateFormat formatador = DateFormat.getDateInstance(DateFormat.SHORT);  
    lblData.setText(formatador.format(data));  
}
```

## **Codificando a opção "Sair"**

- Clicar com o botão direito do mouse sobre "Sair" no menu "Opções", Selecionar **Eventos** → **Action** → **actionPerformed**.

Inserir o código abaixo:

```
private void MenOpcSaiActionPerformed(java.awt.event.ActionEvent evt) {  
    // Exibe uma caixa de diálogo  
    int sair = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja  
    sair?", "Atenção!", JOptionPane.YES_NO_OPTION);  
    if (sair == JOptionPane.YES_OPTION){  
        System.exit(0);  
    }  
}
```

### Construindo e chamando a tela "Sobre"

- Clicar com o botão direito do mouse no pacote "br.com.infox.telas" e criar um Form JFrame.

Nome da classe: TelaSobre

- Selecionar "Propriedades" e em Title digitar "Sobre".
- Inserir quatro labels e editar texto.

label1 → Sistema para controle de ordem de serviços

label2 → Desenvolvido por José de Assis

label3 → Sob a licença GPL

label4 → Apagar texto e inserir o ícone GNU.png no pacote br.com.infox.icones.

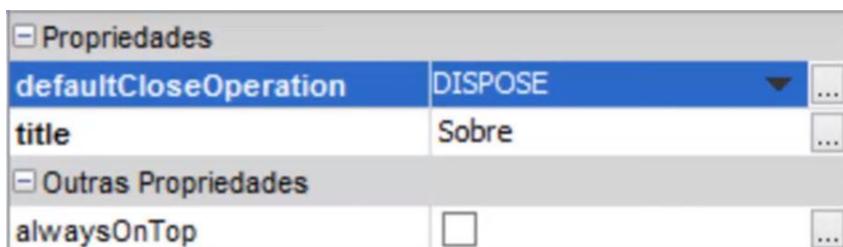


- Clicar com o botão direito do mouse sobre a opção "Sobre" no menu "Ajuda", Selecionar "Eventos → Action → actionPerformed".

- Inserir o código abaixo:

```
private void MenAjuSobActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    TelaSobre sobre = new TelaSobre();  
    sobre.setVisible(true);  
}
```

Ao fechar esta janela, o sistema também é automaticamente fechado. Para evitar que isso ocorra, entre em "propriedades" da Tela Sobre e altere a propriedade defaultCloseOperation de EXIT\_ON\_CLOSE para DISPOSE.



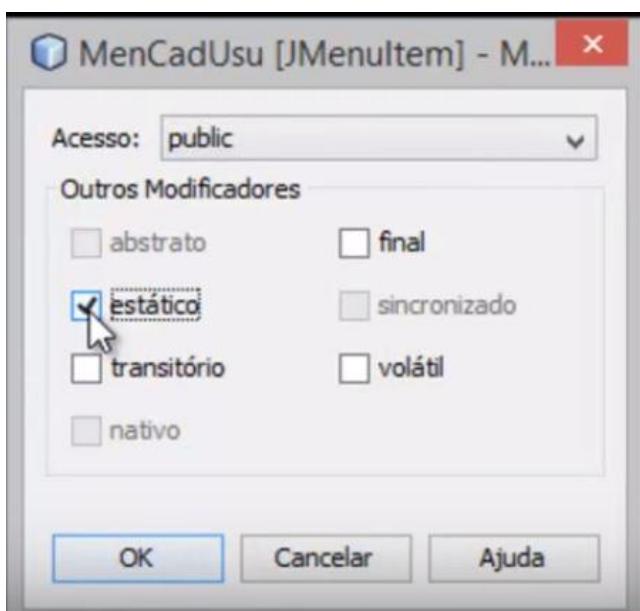
## Desabilitando opções do menu

Se o usuário não for um administrador, ele não pode acessar as opções de menu:

- Cadastro → Usuários;
- Relatórios → Serviços

Portanto, a princípio essas opções devem ser desativadas. Para isso:

- Clicar com o botão direito do mouse sobre a opção "Usuários" do menu "Cadastro". Seleccionar "Propriedades". Clicar na aba "Código".
- Em "Modificadores de variáveis" substituir "private" por "public" e marcar a caixa "estático".



- Na aba "Propriedades" desmarcar a caixa "Enabled".
- Repetir o processo acima para a opção "Relatórios" do menu.

# PERFIL DE USUÁRIO

**Adicionando o campo "perfil" na tabela "tbusuarios"**

```
use dbinfox;
alter table tbusuarios add column perfil varchar(20) not null;
describe tbusuarios;
```

**Removendo um campo da tabela**

```
alter table tbusuarios drop column perfil;
```

**Adicionando novamente o campo "perfil" na tabela "tbusuarios"**

```
alter table tbusuarios add column perfil varchar(20) not null;
```

**Preenchendo o campo perfil**

```
update tbusuarios set perfil='admin' where iduser=1;
update tbusuarios set perfil='admin' where iduser=2;
update tbusuarios set perfil='user' where iduser=3;
```

## Exibindo o menu de acordo com o perfil de usuário

**Arquivo TelaLogin.java**

```
if (rs.next()){
    // Obtendo o conteúdo do campo perfil da tabela tbusuarios
    String perfil = rs.getString(6);
    // A estrutura abaixo faz o tratamento do perfil do usuário
    if (perfil.equals("admin")){
        // A linha abaixo exibe o conteúdo do campo da tabela
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
        TelaPrincipal.MenRel.setEnabled(true);
        TelaPrincipal.MenCadUsu.setEnabled(true);
    } else {
        TelaPrincipal principal = new TelaPrincipal();
        principal.setVisible(true);
    }
    this.dispose();
    conexao.close();
} else {
    JOptionPane.showMessageDialog(null,"Usuário e/ou senha inválido(s)!");
}
} catch (Exception e) {
    JOptionPane.showMessageDialog(null,e);
}
```

## Exibindo o nome do Usuário na Tela Principal

- Clicar com o botão direito do mouse sobre o label lblUsuario. Selecionar "Propriedades". Clicar na aba "Código".
- Em "Modificadores de variáveis" substituir "private" por "public" e marcar a caixa "estático".

Alterar o código para:

```
if (rs.next()){\n    // Obtendo o conteúdo do campo perfil da tabela tbusuarios\n    String perfil = rs.getString(6);\n    // A estrutura abaixo faz o tratamento do perfil do usuário\n    TelaPrincipal principal = new TelaPrincipal();\n    principal.setVisible(true);\n    if (perfil.equals("admin")){\n        // A linha abaixo exibe o conteúdo do campo da tabela\n        TelaPrincipal.MenRel.setEnabled(true);\n        TelaPrincipal.MenCadUsu.setEnabled(true);\n        TelaPrincipal.lblUsuario.setText(rs.getString(2));\n        TelaPrincipal.lblUsuario.setForeground(Color.red);\n    } else {\n        TelaPrincipal.lblUsuario.setText(rs.getString(2));\n    }\n    this.dispose();\n    conexao.close();\n} else {\n    JOptionPane.showMessageDialog(null,"Usuário e/ou senha inválido(s)!");\n}\n} catch (Exception e) {\n    JOptionPane.showMessageDialog(null,e);\n}
```

# CONSTRUINDO A TELA DE USUÁRIOS

- Clicar com o botão direito do mouse no pacote "br.com.infox.telas". Selecionar a opção **Novo → Outros**.
- Em Categorias selecionar "**Forms Gui Swing**" e em tipo de arquivos selecionar "**FormJInternalFrame**". Isto permite criar um formulário interno.

Nome da Classe: TelaUsuario

- Em Propriedades inserir o título:

Title: Usuários

- Alterar o tamanho do formulário para o mesmo tamanho dado para o Painel Desktop (640 x 480).

preferredSize: 640 x 480

Aba Código → Política de Tamanho de Form → Gerar código de redimensionamento

- Em "**Propriedades**" marcar as caixas "**closable**", "**iconifiable**", "**maximizable**".
- Em TelaPrincipal, clicar com o botão direito do mouse sobre o submenu "**Usuários**" e selecionar **Eventos → Action → ActionPerformed**
- Em TelaUsuario inserir 6 labels com os seguintes textos:

- id (txtUsul)
- Nome (txtUsuNome)
- Telefone (txtUsuFone)
- Login (txtUsuLogin)
- Senha (txtUsuSenha)
- Perfil (cboUsuPerfil)

- Inserir um campo de texto para os cinco primeiros labels e um combobox para o label Perfil.

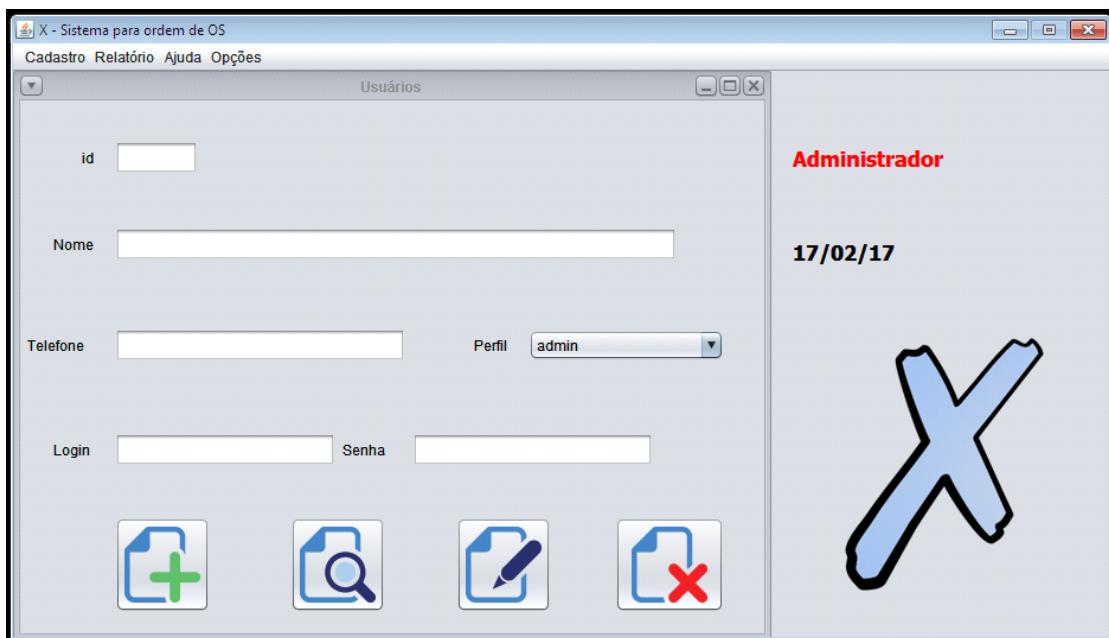
- Acessar o site [www.iconfinder.com](http://www.iconfinder.com) e fazer o download de quatro ícones (64 px X 64 px) que serão responsáveis pelo CRUD.

- Renomear os ícones e inseri-los em forma de botão no formulário:

- create (btnUsuCreate)
- read (btnUsuRead)
- update (btnUsuUpdate)
- delete (btnUsuDelete)

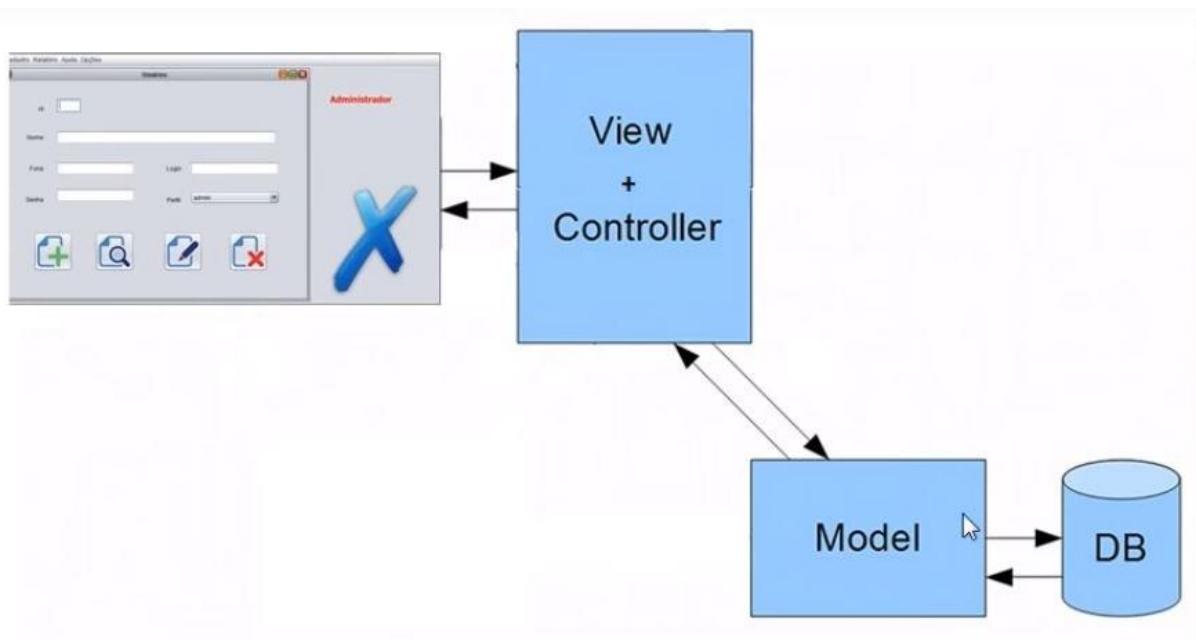
- Na propriedade:

- "Preferred Size" alterar para 80 x 80.
- "toolTipText" preencher com "Adicionar", "Consultar", "Alterar", "Remover".
- "Cursor" selecionar "Cursor em forma de mão".



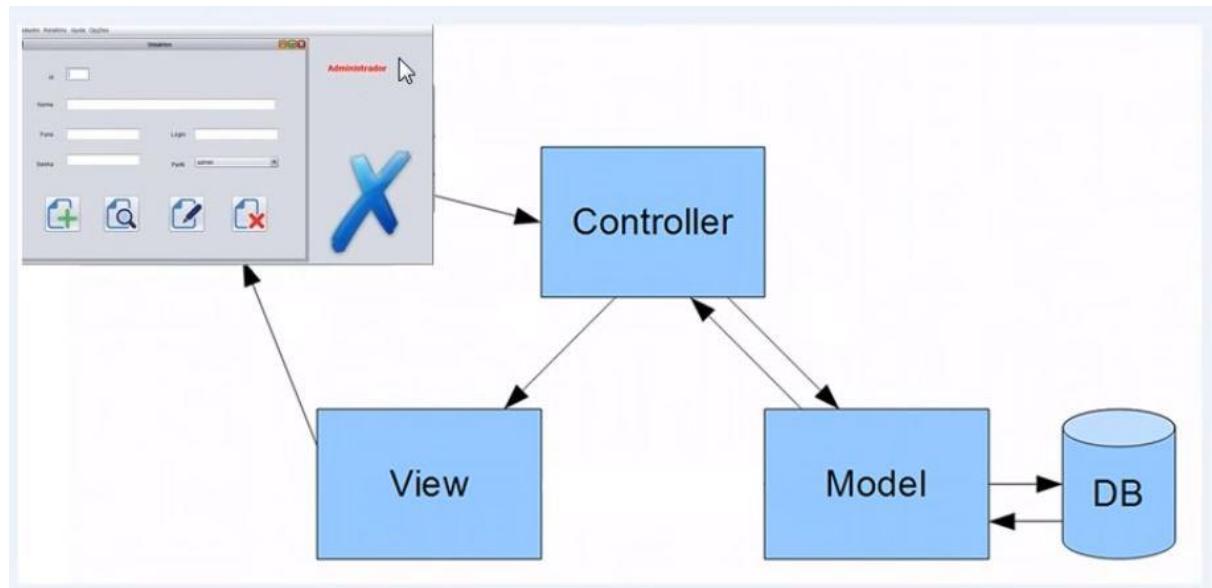
# CAMADAS DE UM SISTEMA

## Sistema de 2 camadas



Estamos construindo um sistema de 2 camadas.

## Sistema de 3 camadas (MVC)



# Consulta de usuários

## Método consultar()

No código do formulário "**TelaUsuario**" inserir as seguintes linhas:

```
package br.com.infox.telas;

import java.sql.*;
import br.com.infox.dal.ModuloConexao;
import javax.swing.JOptionPane;

public class TelaUsuario extends javax.swing.JInternalFrame {
    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    /**
     * Creates new form TelaUsuario
     */
    public TelaUsuario() {
        initComponents();
        conexao = ModuloConexao.conector();
    }

    private void consultar(){
        String sql = "select * from tbusuarios where iduser=?";
        try {
            pst=conexao.prepareStatement(sql);
            pst.setString(1,txtUsulId.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));
                // A linha abaixo se refere ao combobox
                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);
        }
    }
}
```

Digite o id e clique no botão "Consultar":

X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

Usuários

\* id  \* Campos obrigatórios

\* Nome

Fone

\* Login  \* Senha

\* Perfil

**Administrador**  
**14/01/19**

**X**

# Criando um novo usuário

## Método adicionar()

Abaixo do método consultar() inserir as seguintes linhas:

```
// Método para adicionar usuários
private void adicionar() {
    String sql = "insert into tbusuarios (iduser,usuario,fone,login,senha,perfil) values
(?,?,?,?,?,?)";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, txtUsuid.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 (((txtUsuid.getText().isEmpty()) || (txtUsuNome.getText().isEmpty()) ||
(txtUsuLogin.getText().isEmpty()) || (txtUsuSenha.getText().isEmpty())))
            JOptionPane.showMessageDialog(null, "Preencha todos os campos
obrigatórios!");
        } else {
            // A linha abaixo atualiza a tabela tbusuarios com os dados do formulário
            // A estrutura abaixo é usada para confirmar os dados na tabela
            int adicionado = pst.executeUpdate();
            // A linha abaixo serve de apoio ao entendimento da lógica
            // System.out.println(adicionado);
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Usuário adicionado com sucesso!");
                // As linhas abaixo limpam os campos
                txtUsuid.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Preencha os dados do novo usuário e clique no botão "Adicionar":

X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

Usuários

\* id: 4 \* Campos obrigatórios

\* Nome: Bill Gates

Fone: 4444-4444

\* Login: bill \* Senha: 123

\* Perfil: user

**Administrador**  
14/01/19



Usuários

\* id: 4 \* Campos obrigatórios

\* Nome: Bill Gates

Fone: 4444-4444

\* Login: bill

\* Perfil: user

Mensagem

usuário adicionado com sucesso!

OK

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	iduser	usuario	fone	login	senha	perfil
▶	1	José de Assis	9999-9999	joseassis	123456	admin
	2	Administrador	8888-8888	admin	admin	admin
	3	Leandro Ramos	9999-9999	leandro	123	user
*	4	Bill Gates	4444-4444	bill	123	user
	NULL	NULL	NULL	NULL	NULL	NULL

# Alterando os dados de um usuário

## Método alterar()

Abaixo do método adicionar() inserir as seguintes linhas:

```
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, "Preencha todos os campos
obrigatórios!");
        } else {
            // A linha abaixo atualiza a tabela tbusuarios com os dados do formulário
            // A estrutura abaixo é usada para confirmar a alteração dos dados na tabela
            int adicionado = pst.executeUpdate();
            // A linha abaixo serve de apoio ao entendimento da lógica
            // System.out.println(adicionado);
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Dados do usuário alterados com
sucesso!");
                // As linhas abaixo limpam os campos
                txtUsuld.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Altere a senha do usuário de id 4 de 123 para 12345 e clique no botão "Alterar":

X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

Usuários

\* id: 4 \* Campos obrigatórios

\* Nome: Bill Gates

Fone: 4444-4444

\* Login: bill \* Senha: 12345

\* Perfil: user

Administrador  
14/01/19

A large blue 'X' is displayed on the right side of the window.

X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

Usuários

\* id: 4 \* Campos obrigatórios

\* Nome: Bill Gates

Fone: 4444-4444

\* Login: bill

\* Senha: 12345

\* Perfil: user

Mensagem

Dados do usuário alterados com sucesso!

OK

Administrador  
14/01/19

A large blue 'X' is displayed on the right side of the window.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	iduser	usuario	fone	login	senha	perfil
▶	1	José de Assis	9999-9999	joseassis	123456	admin
	2	Administrador	8888-8888	admin	admin	admin
	3	Leandro Ramos	9999-9999	leandro	123	user
*	4	Bill Gates	4444-4444	bill	12345	user
*		NULL	NULL	NULL	NULL	NULL

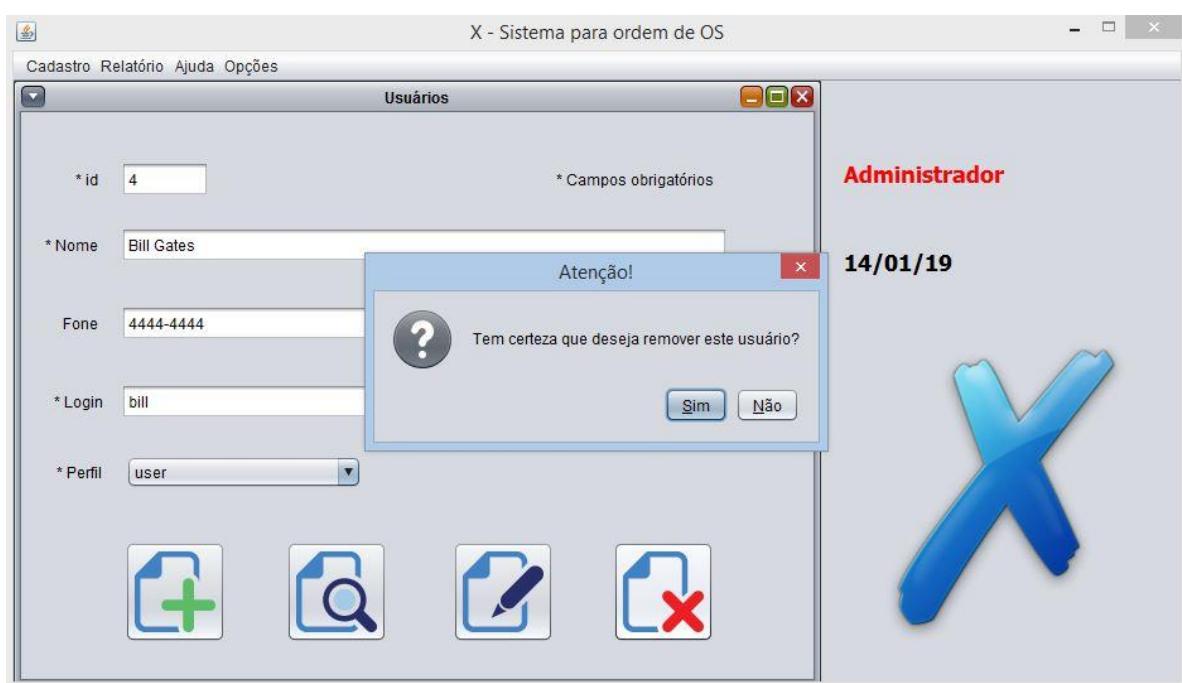
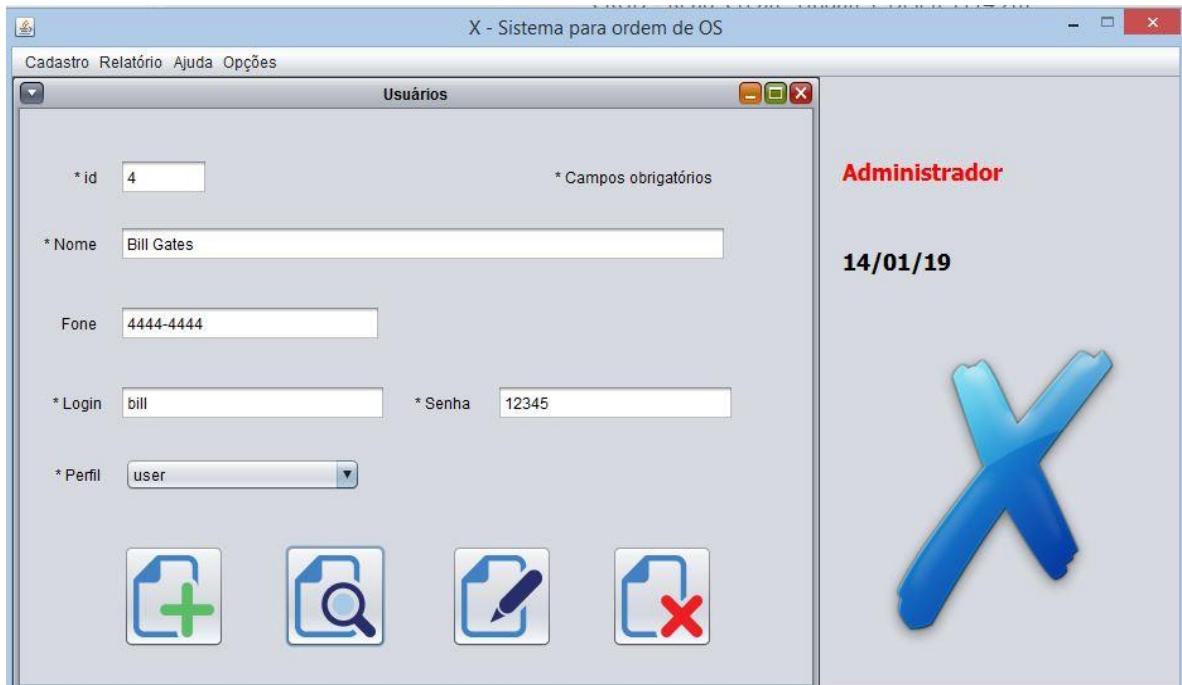
# Excluindo um usuário

## Método remover()

Abaixo do método alterar() inserir as seguintes linhas:

```
// Método responsável pela remoção de usuários
private void remover() {
    // A estrutura abaixo confirma a remoção do usuário
    int confirma = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja remover
este 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, txtUsuid.getText());
            int apagado = pst.executeUpdate();
            if (apagado > 0) {
                JOptionPane.showMessageDialog(null, "Usuário removido com sucesso!");
                txtUsuid.setText(null);
                txtUsuNome.setText(null);
                txtUsuFone.setText(null);
                txtUsuLogin.setText(null);
                txtUsuSenha.setText(null);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
```

Com o usuário de id 4 selecionado, clique no botão "Remover" e confirme a remoção:



X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

### Usuários

\* id: 4 \* Campos obrigatórios

\* Nome: Bill Gates

Fone: 4444-4444

\* Login: bill

\* Perfil: user

**Administrador**

**14/01/19**

**Mensagem**

Usuário removido com sucesso!



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	iduser	usuario	fone	login	senha	perfil
▶	1	José de Assis	9999-9999	joseassis	123456	admin
	2	Administrador	8888-8888	admin	admin	admin
*	3	Leandro Ramos	9999-9999	leandro	123	user
*	HULL	HULL	HULL	HULL	HULL	HULL

# CONSTRUINDO A TELA DE CLIENTES

- Clicar com o botão direito do mouse no pacote "br.com.infox.telas". Selecionar a opção Novo → "FormJInternalFrame". Isto permite criar um formulário interno.

Nome da Classe: TelaCliente

- Em Propriedades inserir o título:

Title: Clientes

- Alterar o tamanho do formulário para o mesmo tamanho dado para o Painel Desktop (640 x 480).

preferredSize: 640 x 480

Aba Código → Política de Tamanho de Form → Gerar código de redimensionamento

- Em "Propriedades" marcar as caixas "closable", "iconifiable", "maximizable".

- Em TelaPrincipal, clicar com o botão direito do mouse sobre o submenu "Clientes" e selecionar Eventos → Action → ActionPerformed

- Em TelaCliente inserir 4 labels com os seguintes textos:

- Nome (txtCliNome)
- Endereço (txtCliEndereco)
- Telefone (txtCliFone)
- email (txtCliEmail)

- Inserir um campo de texto para cada label.

- Inserir um campo de texto (txtCliPesquisar).

- Inserir uma tabela (tblClientes).

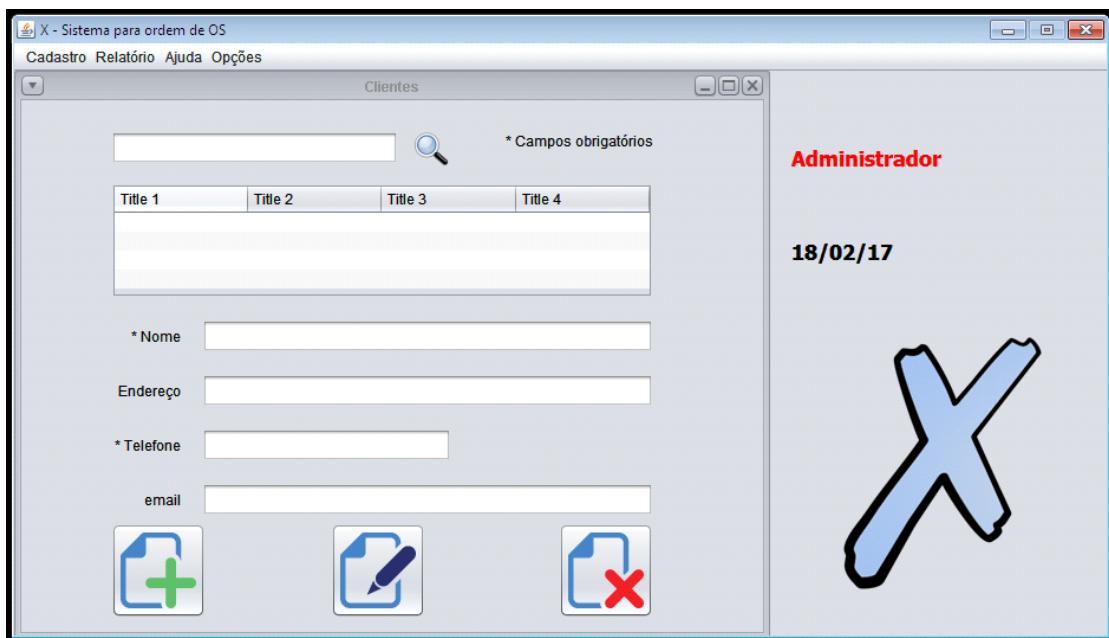
- Acessar o site [www.iconfinder.com](http://www.iconfinder.com) e fazer o download de um ícone (32 px X 32 px) para "Pesquisar".

- Renomear os ícones e inseri-los em forma de botão no formulário:

- create (btnCliAdicionar)
- pesquisar (btnUsuPesquisar)
- update (btnCliAlterar)
- delete (btnCliRemover)

- Na propriedade:

- "Preferred Size" alterar para 80 x 80.
- "toolTipText" preencher com "Adicionar", "Alterar", "Remover".
- "Cursor" selecionar "Cursor em forma de mão".



# CADASTRANDO CLIENTES

Inserir as seguintes linhas em TelaCliente:

```
package br.com.infox.telas;

import java.sql.*;
import br.com.infox.dal.ModuloConexao;
import javax.swing.JOptionPane;

public class TelaCliente extends javax.swing.JInternalFrame {

    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public TelaCliente() {
        initComponents();
        conexao = ModuloConexao.conector();
    }
}
```

Pelo MySQL Workbench, inserir os seguintes clientes:

-- Selecionando o banco de dados a ser utilizado  
use dbinfox;

-- Inserir clientes na tabela tbclientes:

```
insert into tbclientes(nomecli,endcli,fonecli,emailcli) values ('José de Assis Filho','Rua do Gasometro, 1234','9999-9999','professorjoseassis@gmail.com');
insert into tbclientes(nomecli,endcli,fonecli,emailcli) values ('João da Silva','Rua Vitória, 56','5678-9009','joao@hotmail.com');
insert into tbclientes(nomecli,endcli,fonecli,emailcli) values ('Leandro Ramos','Rua Tuiuti, 2900','3900-0000','professorjoseassis@gmail.com');
insert into tbclientes(nomecli,endcli,fonecli,emailcli) values ('Juliana Paes','Rua Santa Luzia, 456','7890-0987','ju@globo.com');
insert into tbclientes(nomecli,endcli,fonecli,emailcli) values ('Joaquim Jorge','Av. Sumaré, 2009 - casa 2','2908-0987','joaquim@yahoo.com');

select * from tbclientes;
```

	iddi	nomecli	endcli	fonecli	emailcli
▶	1	Linus Torvalds	Rua Tux, 2015	9999-9999	linus@linux.com
	2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
	3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
	4	Leandro Ramos	Rua Tuiuti, 2900	3900-0000	professorjoseassis@gmail.com
	5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
	6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
	NULL	NULL	NULL	NULL	NULL

## Método adicionar()

Inserir as seguintes linhas:

```
// Método para adicionar clientes
private void adicionar() {
    String sql = "insert into tbclientes (nomecli,endcli,fonecli,emailcli) values (?,?,?,?,?)";
    try {
        pst = conexao.prepareStatement(sql);
        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, "Preencha todos os campos
obrigatórios!");
        } else {
            // A linha abaixo atualiza a tabela tbclientes com os dados do formulário
            // A estrutura abaixo é usada para confirmar os dados na tabela
            int adicionado = pst.executeUpdate();
            // A linha abaixo serve de apoio ao entendimento da lógica
            // System.out.println(adicionado);
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "Cliente adicionado com sucesso!");
                // As linhas abaixo limpam os campos
                txtCliNome.setText(null);
                txtCliFone.setText(null);
                txtCliEndereco.setText(null);
                txtCliEmail.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Preencha os campos, conforme indicado abaixo e clique no botão "Adicionar":

**Administrador**

**14/01/19**

**X**

**Administrador**

**14/01/19**

**X**

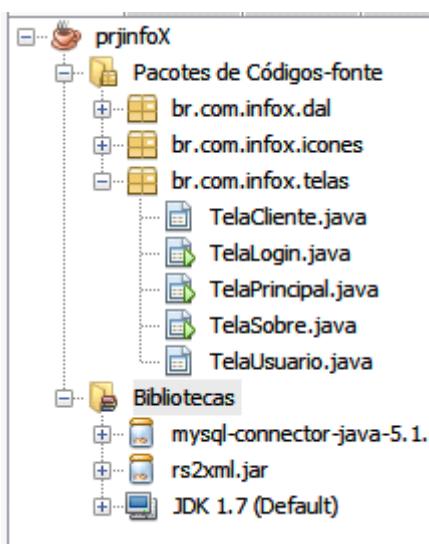
**Result Grid** | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

	idcli	nomedcli	endcli	fonedi	emailcli
▶	1	Linus Torvalds	Rua Tux, 2015	9999-9999	linus@linux.com
	2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
	3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
	4	Leandro Ramos	Rua Tuiuti, 2900	3900-0000	professorjoseassis@gmail.com
	5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
	6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
	7	Leonardo Teixeira	Rua das Acácas, 175	9872-3344	leoteixeira@hotmail.com

# CONSULTA AVANÇADA DE CLIENTES

Adicionar a biblioteca rs2xml (arquivo rs2xml.jar) no link abaixo:  
<https://sourceforge.net/projects/finalangelsanddemons/>

- No pacote "Bibliotecas" clicar com o botão direito do mouse e selecionar "Adicionar JAR/Pasta" e adicionar o arquivo "rs2xml.jar".



Inserir o código abaixo:

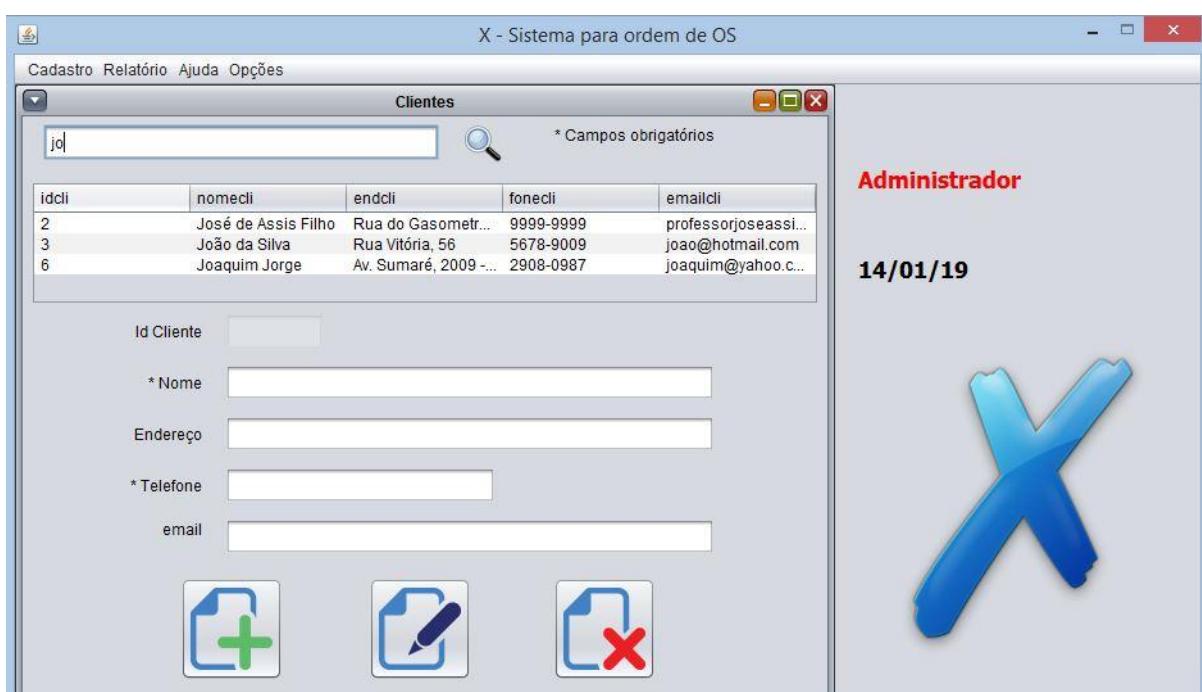
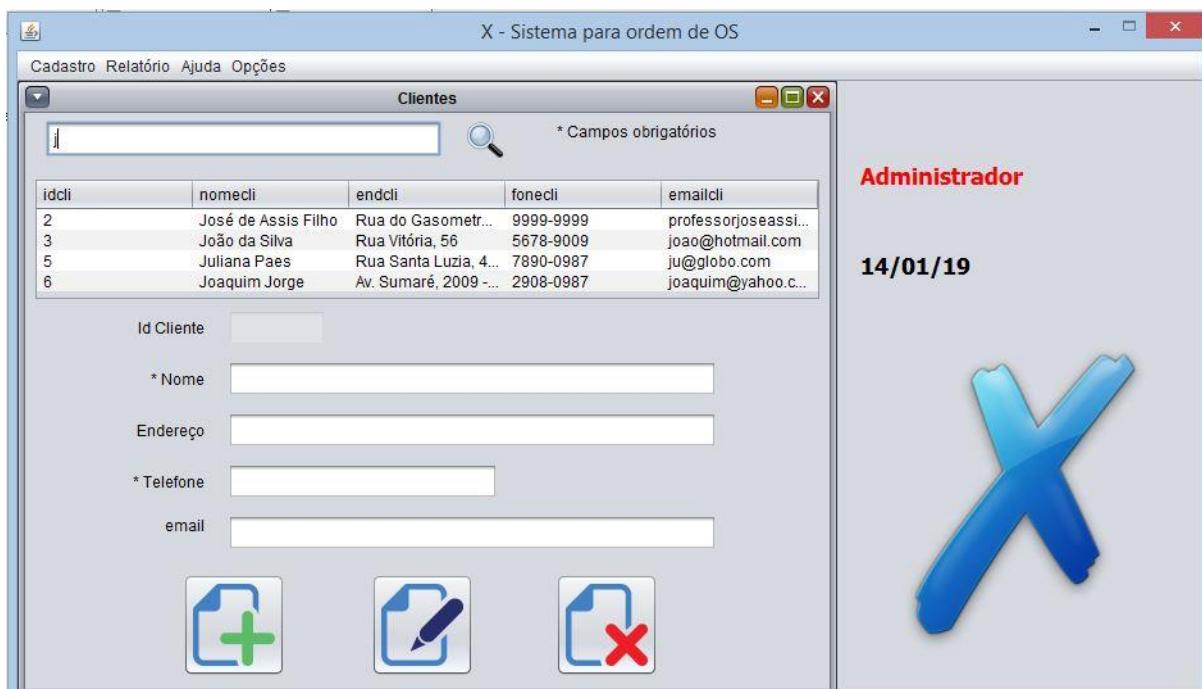
```
// Método para pesquisar clientes pelo nome com filtro
private void pesquisar_cliente() {
    String sql = "select * from tbclientes where nomecli like ?";
    try {
        pst = conexao.prepareStatement(sql);
        // Passando o conteúdo da caixa de pesquisa para o ?
        // Atenção ao "%" - continuação da String sql
        pst.setString(1, txtCliPesquisar.getText() + "%");
        rs = pst.executeQuery();
        // A linha abaixo usa a biblioteca rs2xml.jar para preencher a tabela
        tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

// Método para setar os campos do formulário com o conteúdo da tabela
public void setar_campos() {
    int setar = tblClientes.getSelectedRow();
    txtCliNome.setText(tblClientes.getModel().getValueAt(setar, 1).toString());
    txtCliEndereco.setText(tblClientes.getModel().getValueAt(setar, 2).toString());
    txtCliFone.setText(tblClientes.getModel().getValueAt(setar, 3).toString());
    txtCliEmail.setText(tblClientes.getModel().getValueAt(setar, 4).toString());
}
```

- Clicar com o botão direito do mouse sobre o campo de texto "txtCliPesquisar" e selecionar Action → Eventos → Key → KeyReleased

- Inserir o código:

```
// O evento abaixo é do tipo "enquanto for digitando"
private void txtCliPesquisarKeyReleased(java.awt.event.KeyEvent evt) {
    // chamar o método pesquisar_cliente()
    pesquisar_cliente();
}
```



## Setando campos

- Clicar com o botão direito do mouse sobre a tabela "tblClientes" e selecionar Action → Eventos → Mouse → MouseClicked

- Inserir o código:

```
// Evento que será usado para setar os campos da tabel (clicando com o botão esquerdo do mouse)
```

```
private void tblClientesMouseClicked(java.awt.event.MouseEvent evt) {  
    // Chamando o método para setar os campos  
    setar_campos();  
}
```

- Inserir um label e um campo de texto (txtClId)

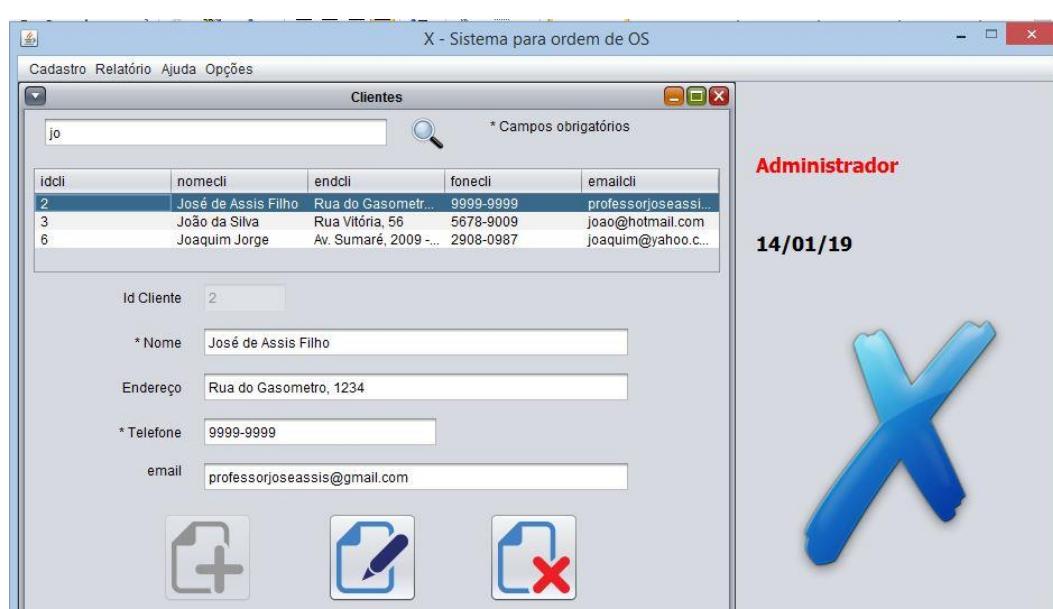
- Em "Propriedades" do campo de texto "txtClId" desmarcar a caixa Enable.

- Alterar o código do método setar\_campos() para:

```
// Método para setar os campos do formulário com o conteúdo da tabela  
public void setar_campos() {
```

```
    int setar = tblClientes.getSelectedRow();  
    txtClId.setText(tblClientes.getModel().getValueAt(setar, 0).toString());  
    txtCliNome.setText(tblClientes.getModel().getValueAt(setar, 1).toString());  
    txtCliEndereco.setText(tblClientes.getModel().getValueAt(setar, 2).toString());  
    txtCliFone.setText(tblClientes.getModel().getValueAt(setar, 3).toString());  
    txtCliEmail.setText(tblClientes.getModel().getValueAt(setar, 4).toString());  
    // A linha abaixo desabilita o botão Adicionar  
    btnAdicionar.setEnabled(false);  
}
```

Clique no registro de id=2:



# ATUALIZANDO CLIENTES

- Alterar as seguintes linhas de código do método alterar() para:

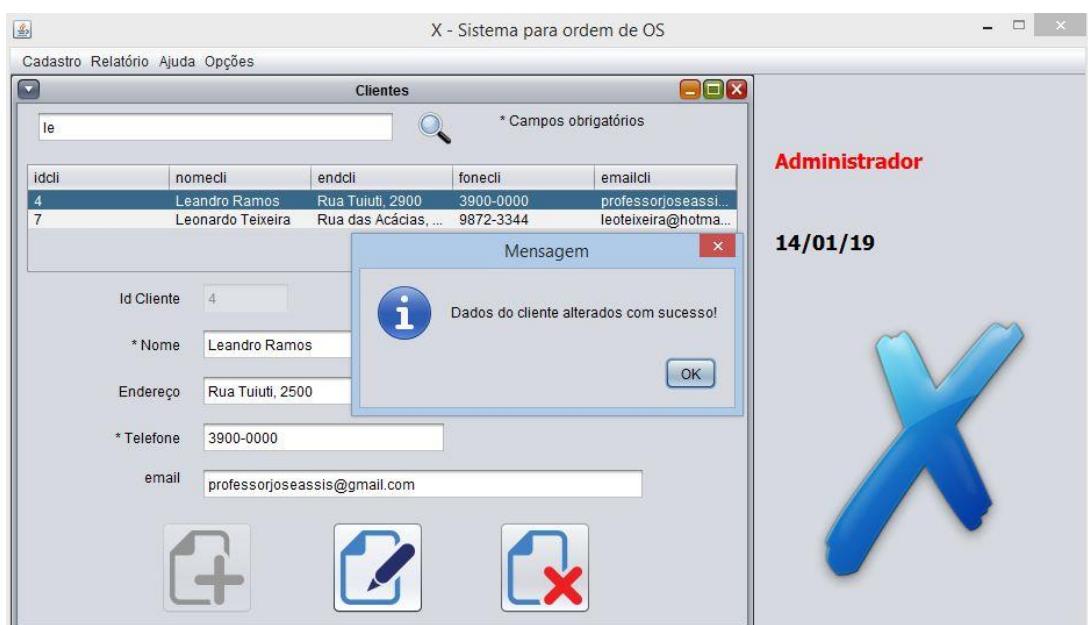
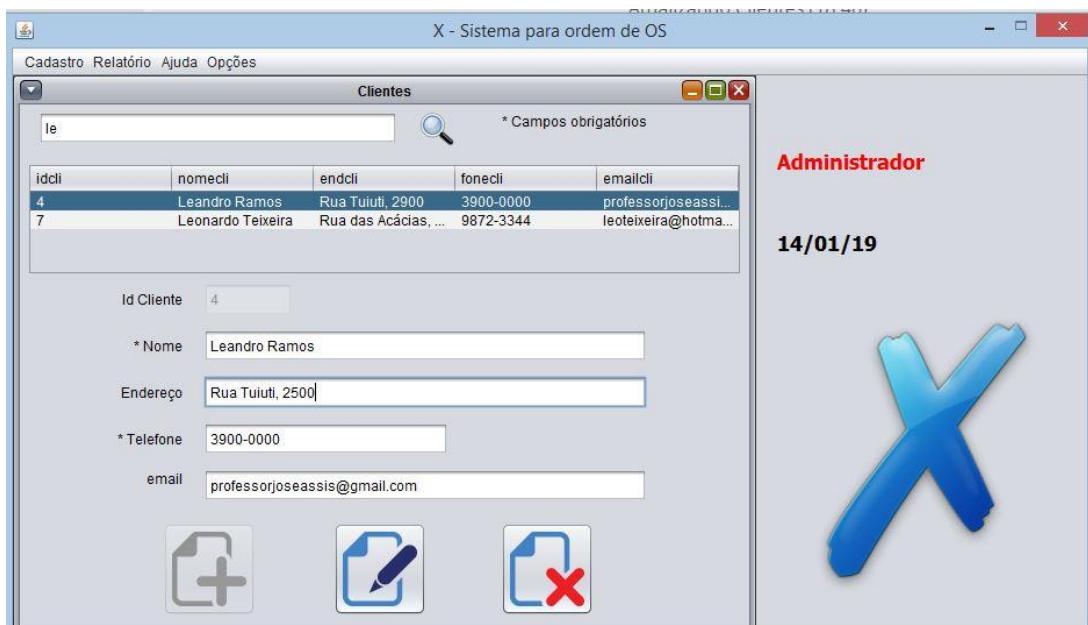
```
private void alterar() {  
    String sql = "update tbclientes set nomecli=?, endcli=?, fonecli=?, emailcli=? where  
    idcli=?";  
    try {  
        pst = conexao.prepareStatement(sql);  
        pst.setString(1, txtCliNome.getText());  
        pst.setString(2, txtCliEndereco.getText());  
        pst.setString(3, txtCliFone.getText());  
        pst.setString(4, txtCliEmail.getText());  
        pst.setString(5, txtClid.getText());  
        // Validação dos campos obrigatórios  
        if ((txtCliNome.getText().isEmpty()) || (txtCliFone.getText().isEmpty())) {  
            JOptionPane.showMessageDialog(null, "Preencha todos os campos  
obrigatórios!");  
        } else {  
            // A linha abaixo atualiza a tabela tbclientes com os dados do formulário  
            // A estrutura abaixo é usada para confirmar a alteração dos dados na tabela  
            int adicionado = pst.executeUpdate();  
            // A linha abaixo serve de apoio ao entendimento da lógica  
            // System.out.println(adicionado);  
            if (adicionado > 0) {  
                JOptionPane.showMessageDialog(null, "Dados do cliente alterados com  
sucesso!");  
                // As linhas abaixo limpam os campos  
                txtCliNome.setText(null);  
                txtCliEndereco.setText(null);  
                txtCliFone.setText(null);  
                txtCliEmail.setText(null);  
                btnAdicionar.setEnabled(true);  
            }  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, e);  
    }  
}
```

- Clicar com o botão direito do mouse sobre o botão "btnAlterar" e selecionar **Eventos → Action → actionPerformed**

- Inserir o código:

```
private void btnAlterarActionPerformed(java.awt.event.ActionEvent evt) {  
    // Chamando método para alterar dados do cliente  
    alterar();  
}
```

Alterar o endereço de Leandro Ramos de "Rua Tuiuti, 2900" para "Rua Tuiuti, 2500" e clicar no botão "Alterar":



Result Grid					
	idcli	nomecli	endcli	fonecli	emailcli
▶	1	Linus Torvalds	Rua Tux,2015	9999-9999	linus@linux.com
	2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
	3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
	4	Leandro Ramos	Rua Tuiuti, 2500	3900-0000	professorjoseassis@gmail.com
	5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
	6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
	7	Leonardo Teixeira	Rua das Acádias, 175	9872-3344	leoteixeira@hotmail.com

# REMOVENDO CLIENTES

- Inserir o código abaixo:

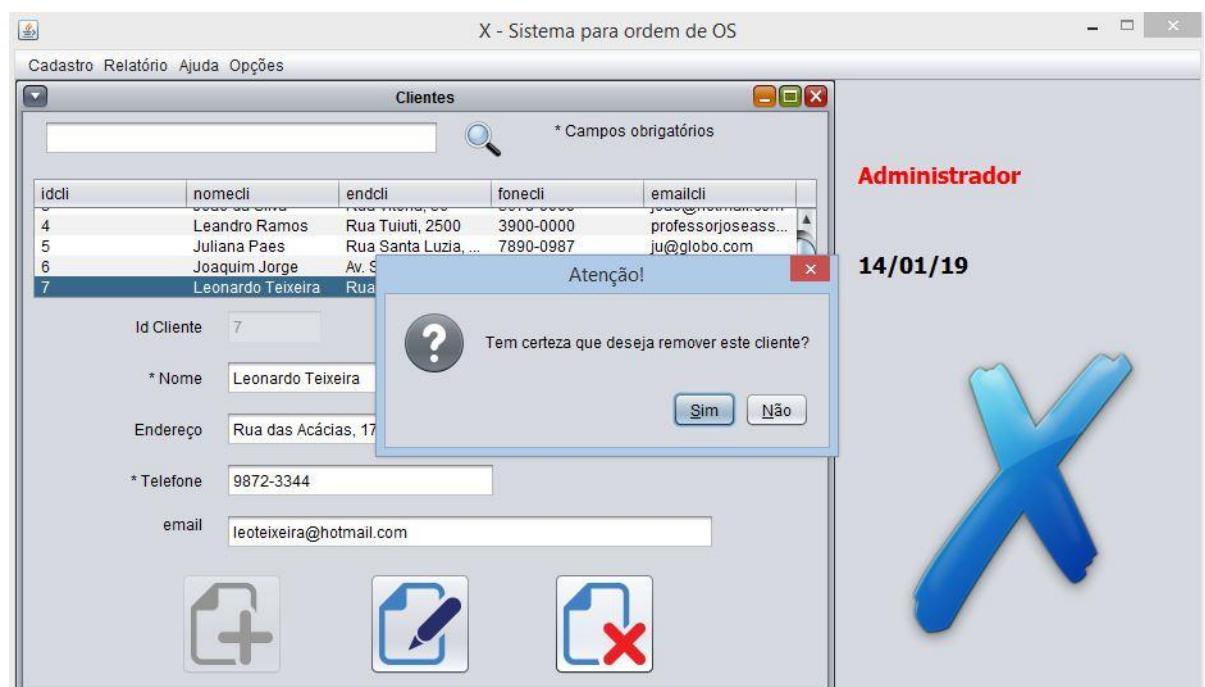
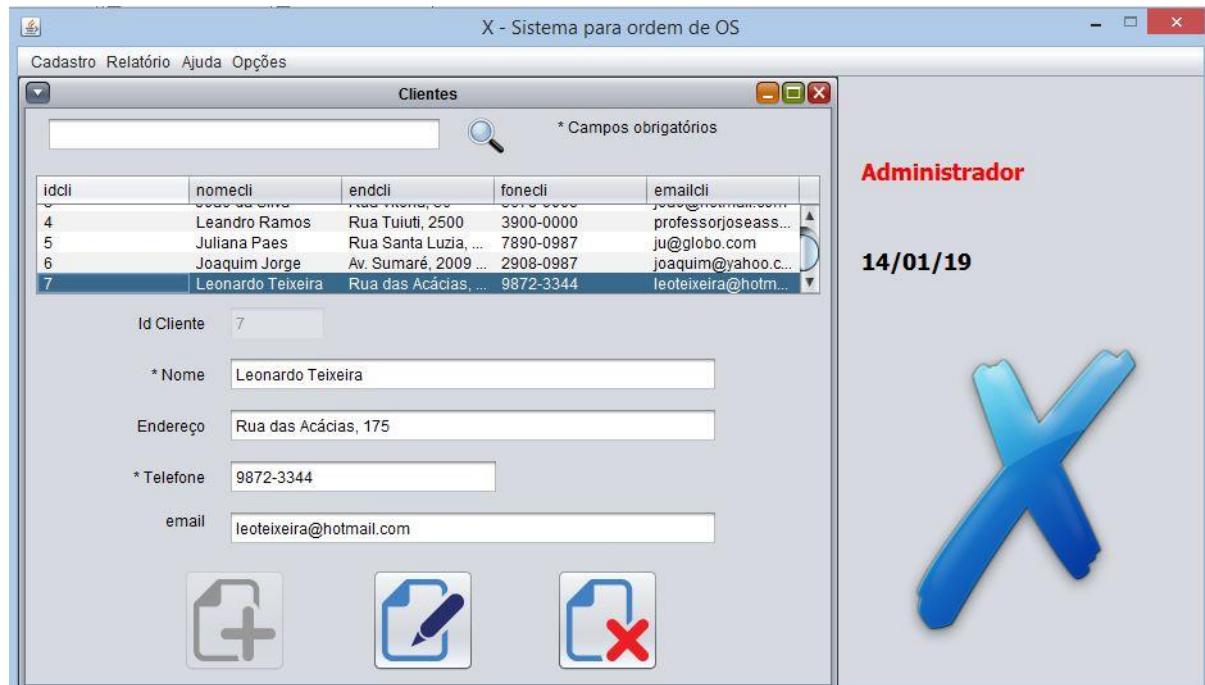
```
// Método responsável pela remoção de clientes
private void remover() {
    // A estrutura abaixo confirma a remoção do cliente
    int confirma = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja
remover este cliente?", "Atenção!", JOptionPane.YES_NO_OPTION);
    if (confirma == JOptionPane.YES_OPTION) {
        String sql = "delete from tbclientes where idcli=?";
        try {
            pst = conexao.prepareStatement(sql);
            pst.setString(1, txtClId.getText());
            int apagado = pst.executeUpdate();
            if (apagado > 0) {
                JOptionPane.showMessageDialog(null, "Cliente removido com
sucesso!");
                txtCliNome.setText(null);
                txtCliEndereco.setText(null);
                txtCliFone.setText(null);
                txtCliEmail.setText(null);
                btnAdicionar.setEnabled(true);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
```

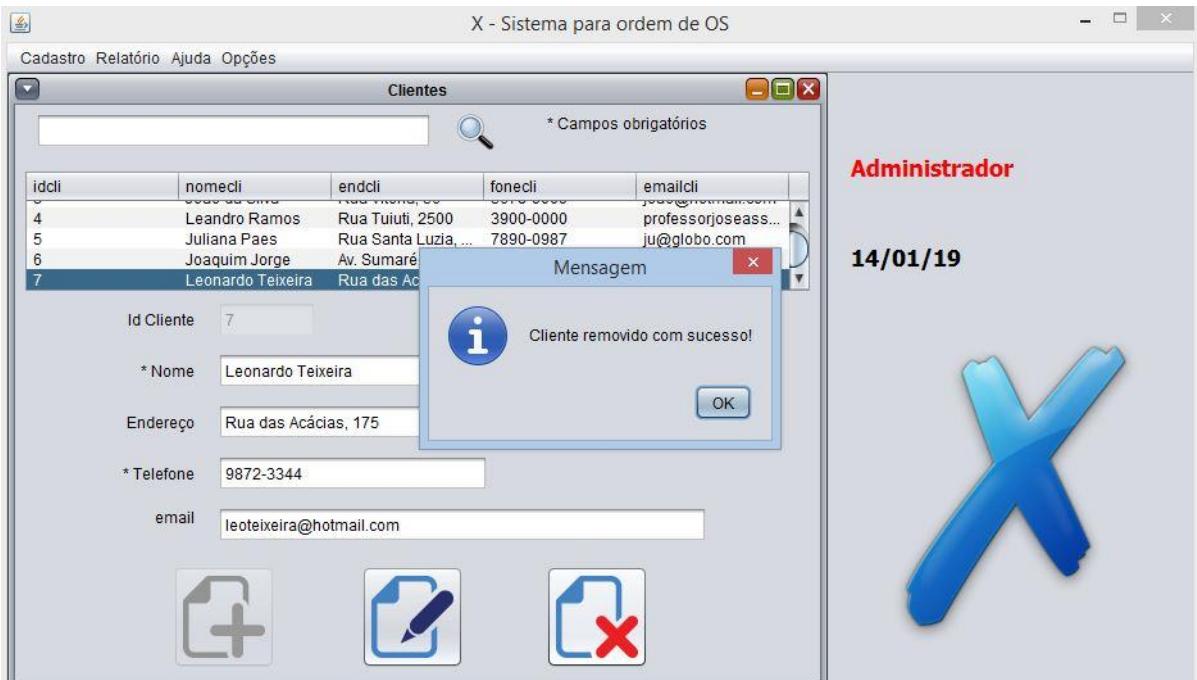
- Clicar com o botão direito do mouse sobre o botão "btnRemover" e selecionar **Eventos → Action → actionPerformed**

- Inserir o código:

```
private void btnRemoverActionPerformed(java.awt.event.ActionEvent evt) {
    // Chamando o método para remover um cliente
    remover();
}
```

Selecione o cliente Leonardo Teixeira e clique no botão "Remover":



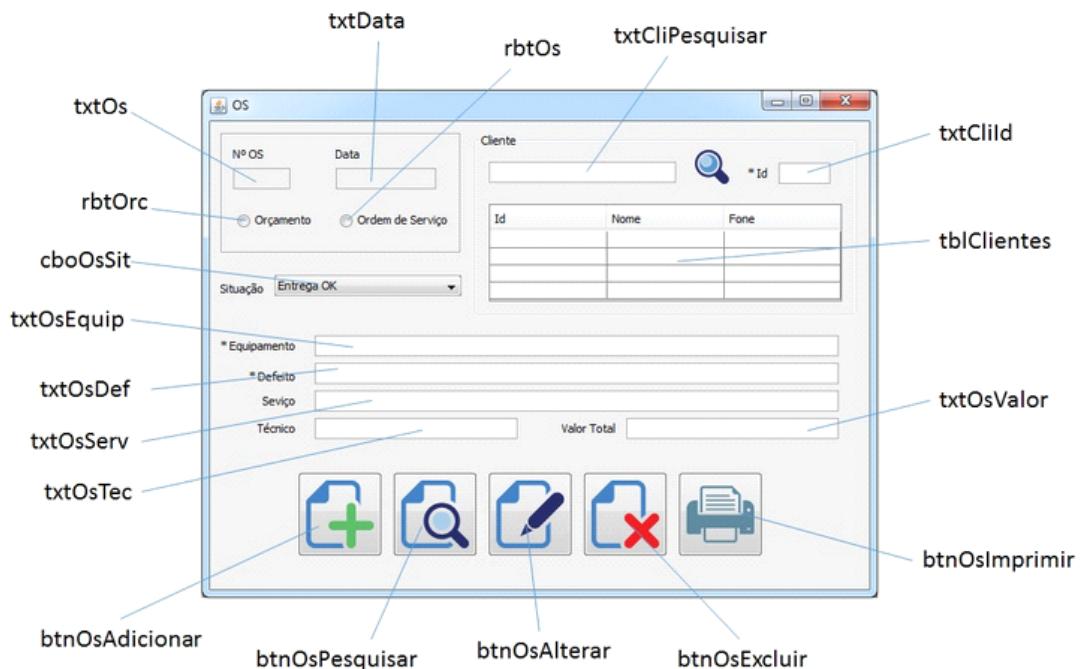


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	idcli	nomedcli	endcli	fonedi	emailcli
▶	1	Linus Torvalds	Rua Tux, 2015	9999-9999	linus@linux.com
	2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
	3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
	4	Leandro Ramos	Rua Tuiuti, 2500	3900-0000	professorjoseassis@gmail.com
	5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
	6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
*	NULL	NULL	NULL	NULL	NULL

# CONSTRUINDO A TELA OS

Tela OS – Nome das variáveis dos objetos:



## Vínculo do cliente com a tabela OS

```
use dbinfox;
describe tbclientes;
select * from tbclientes;
```

-- a linha abaixo cria um apelido aos campos da tabela

```
select idcli as Id, nomecli as Nome, fonecli as Fone from tbclientes where nomecli
like 'Jo%';
```

Inserir as seguintes lihas em TelaOS:

```
package br.com.infox.telas;

import java.sql.*;
import br.com.infox.dal.ModuloConexao;
import net.proteanit.sql.DbUtils;

public class TelaCliente extends javax.swing.JInternalFrame {

    Connection conexao = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

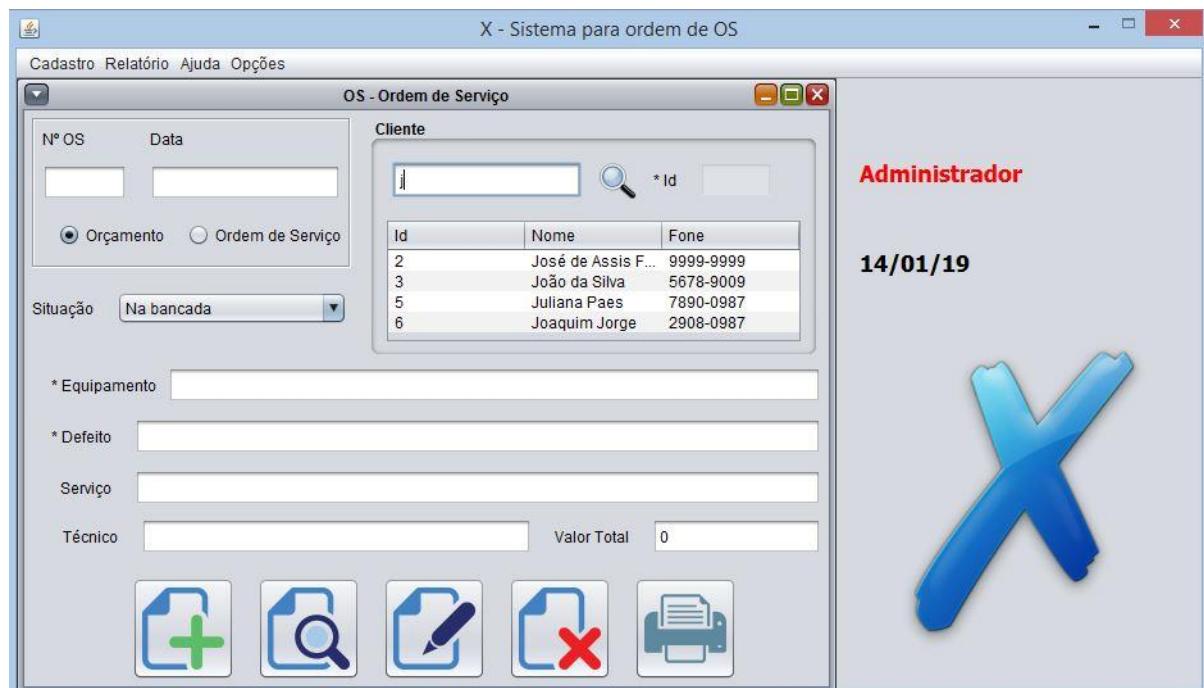
    public TelaCliente() {
        initComponents();
        conexao = ModuloConexao.conector();
    }
}
```

## **pesquisar\_cliente()**

```
// Método para pesquisar clientes pelo nome com filtro

private void pesquisar_cliente() {
    String sql = "select idcli as Id, nomecli as Nome, fonecli as Fone from tbclientes
where nomecli like ?";
    try {
        pst = conexao.prepareStatement(sql);
        // Passando o conteúdo da caixa de pesquisa para o ?
        // Atenção ao "%" - continuação da String sql
        pst.setString(1, txtCliPesquisar.getText() + "%");
        rs = pst.executeQuery();
        // A linha abaixo usa a biblioteca rs2xml.jar para preencher a tabela
        tblClientes.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

// Método para setar os campos do formulário com o conteúdo da tabela
private void setar_campos() {
    int setar = tblClientes.getSelectedRow();
    txtCliId.setText(tblClientes.getModel().getValueAt(setar, 0).toString());
}
```



- Clicar com o botão direito do mouse sobre o radiobutton "**Orcamento**" e selecionar **Eventos → Action → ActionPerformed**

- Inserir o código:

```
private void rbtOrcActionPerformed(java.awt.event.ActionEvent evt) {  
    // Atribuindo um texto a variável "tipo" se o radio button for selecionado  
    tipo = "Orçamento";  
}
```

- Clicar com o botão direito do mouse sobre o radiobutton "**Ordem de Serviço**" e selecionar **Eventos → Action → ActionPerformed**

- Inserir o código:

```
private void rbtOsActionPerformed(java.awt.event.ActionEvent evt) {  
    // Atribuindo um texto a variável "tipo" se o radio button for selecionado  
    tipo = "OS";  
}
```

- Clicar com o botão direito do mouse sobre o form e selecionar **Eventos → Action → InternalFrame → InternalFrame Opened**

- Inserir o código:

```
private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {  
    // Ao abrir o form, marcar o radio button "Orçamento"  
    rbtOrc.setSelected(true);  
    tipo = "Orçamento";  
}
```

## Emitindo uma OS

- No MySQL WorkBench:

```
use dbinfox;
describe tbos;
```

-- A linha abaixo altera a tabela adicionando um campo em uma determinada posição

```
alter table tbos add tipo varchar(15) not null after data_os;
alter table tbos add situacao varchar(20) not null after tipo;
```

## **emitir\_os()**

```
// Método para cadastrar uma OS
private void emitir_os() {
    String sql = "insert into tbos
(tipo,situacao,equipamento,defeito,servico,tecnico,valor,idcli) values (?,?,?,?,?,?,?,?)";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, tipo);
        pst.setString(2, cboOsSit.getSelectedItem().toString());
        pst.setString(3, txtOsEquip.getText());
        pst.setString(4, txtOsDef.getText());
        pst.setString(5, txtOsServ.getText());
        pst.setString(6, txtOsTec.getText());
        // .replace substitui a vírgula pelo ponto
        pst.setString(7, txtOsValor.getText().replace(",", "."));
        pst.setString(8, txtClild.getText());
        // Validação dos campos obrigatórios
        if (((txtClild.getText().isEmpty()) || (txtOsEquip.getText().isEmpty()) ||
(txtOsDef.getText().isEmpty())))
            JOptionPane.showMessageDialog(null, "Preencha todos os campos
obrigatórios!");
        } else {
            // A linha abaixo atualiza a tabela tbclientes com os dados do formulário
            // A estrutura abaixo é usada para confirmar os dados na tabela
            int adicionado = pst.executeUpdate();
            // A linha abaixo serve de apoio ao entendimento da lógica
            // System.out.println(adicionado);
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "OS adicionada com sucesso!");
                // As linhas abaixo limpam os campos
                txtClild.setText(null);
                txtOsEquip.setText(null);
                txtOsDef.setText(null);
                txtOsServ.setText(null);
                txtOsTec.setText(null);
                txtOsValor.setText(null);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

Preencher o formulário com as informações abaixo e clicar no botão "Emitir OS":

The screenshot shows the 'OS - Ordem de Serviço' (Order of Service) module of the 'X - Sistema para ordem de OS' application. The main window displays a form for entering service orders. On the left, there are fields for 'Nº OS' (Order Number) and 'Data' (Date), with radio buttons for 'Orçamento' (Estimate) and 'Ordem de Serviço' (Service Order). Below these is a dropdown menu for 'Situação' (Status) set to 'Aguardando APROVAÇÃO' (Waiting for APPROVAL). Further down are fields for 'Equipamento' (Equipment) containing 'Notebook xyz', 'Defeito' (Fault) containing 'Não liga' (Does not work), 'Serviço' (Service) containing 'Trocada da fonte' (Font replacement), 'Técnico' (Technician) containing 'Leandro', and 'Valor Total' (Total Value) set to '230'. At the bottom are five icons: a document with a plus sign, a magnifying glass, a document with a pen, a document with a red X, and a printer. To the right of the main window is a red text overlay 'Administrador' and a date '14/01/19'. A large blue 'X' is overlaid on the bottom right of the application window.

X - Sistema para ordem de OS

Cadastro Relatório Ajuda Opções

OS - Ordem de Serviço

Cliente

Nº OS Data

Orçamento  Ordem de Serviço

Situação Aguardando APROVAÇÃO

\* Equipamento Notebook xyz

\* Defeito Não liga

Serviço Troca da fonte

Técnico Leandro Valor Total 230

Mensagem

OS adicionada com sucesso!

OK

Administrador

14/01/19

**X**

# Pesquisando uma OS

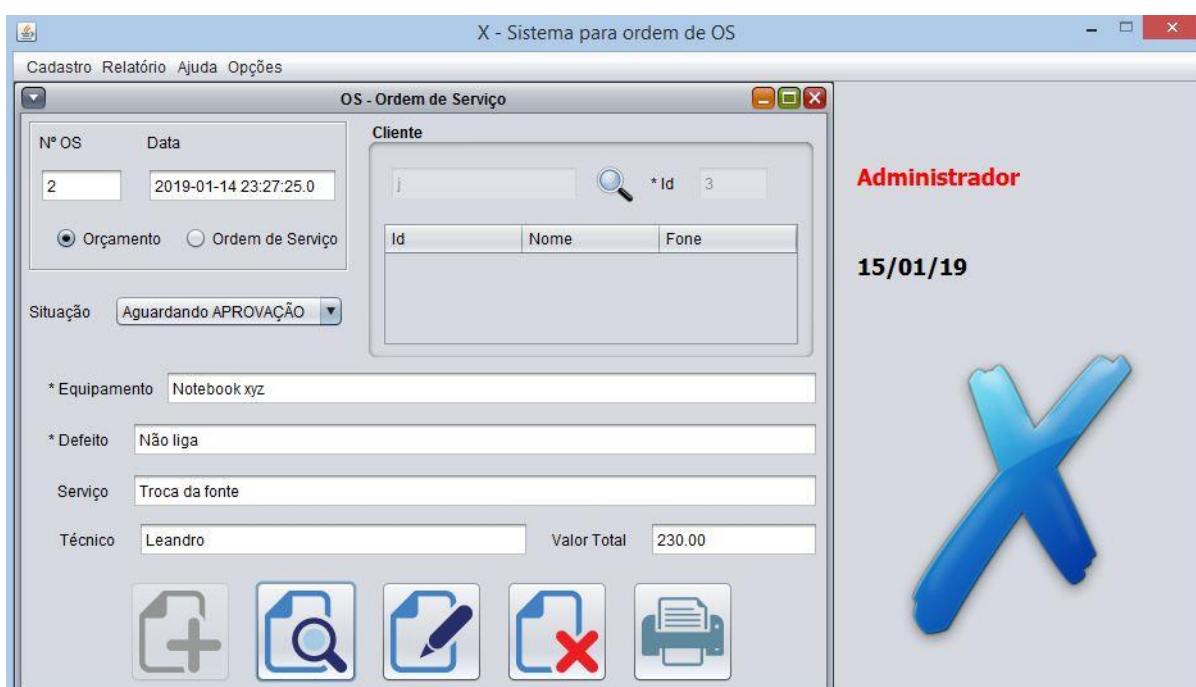
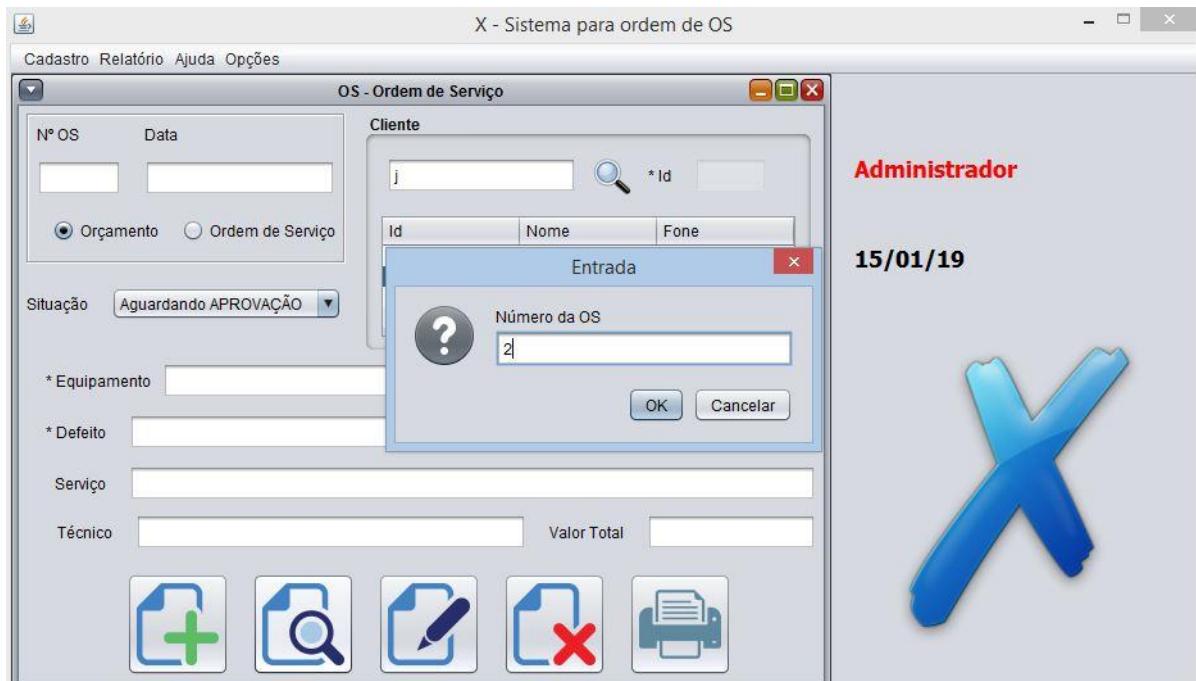
## **pesquisar\_os()**

```
// Método para pesquisar uma OS
private void pesquisar_os() {
    // A linha abaixo cria uma caixa de entrada do tipo JOptionPane
    String num_os = JOptionPane.showInputDialog("Número da OS");
    String sql = "select * from tbos where os = " + num_os;
    try {
        pst = conexao.prepareStatement(sql);
        rs = pst.executeQuery();
        if (rs.next()) {
            txtOs.setText(rs.getString(1));
            txtData.setText(rs.getString(2));
            // Setando os radiobuttons
            String rbtTipo = rs.getString(3);
            if (rbtTipo.equals("OS")) {
                rbtOs.setSelected(true);
                tipo = "OS";
            } else {
                rbtOrc.setSelected(true);
                tipo = "Orçamento";
            }
            cboOsSit.setSelectedItem(rs.getString(4));
            txtOsEquip.setText(rs.getString(5));
            txtOsDef.setText(rs.getString(6));
            txtOsServ.setText(rs.getString(7));
            txtOsTec.setText(rs.getString(8));
            txtOsValor.setText(rs.getString(9));
            txtClild.setText(rs.getString(10));
            // Evitando problemas
            btnOsAdicionar.setEnabled(false);
            txtCliPesquisar.setEnabled(false);
            tblClientes.setVisible(false);
        } else {
            JOptionPane.showMessageDialog(null, "OS não cadastrada!");
            // As linhas abaixo limpam os campos
            txtOsEquip.setText(null);
            txtOsDef.setText(null);
            txtOsServ.setText(null);
            txtOsTec.setText(null);
        }
    } catch (com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException e) {
        JOptionPane.showMessageDialog(null, "OS inválida!");
        // System.out.println(e);
    } catch (Exception e2) {
        JOptionPane.showMessageDialog(null, e2);
    }
}
```

```

private void btnOsPesquisarActionPerformed(java.awt.event.ActionEvent evt) {
    // Chamando o método pesquisar_os
    pesquisar_os();
}

```



## Alterando uma OS

### alterar\_os()

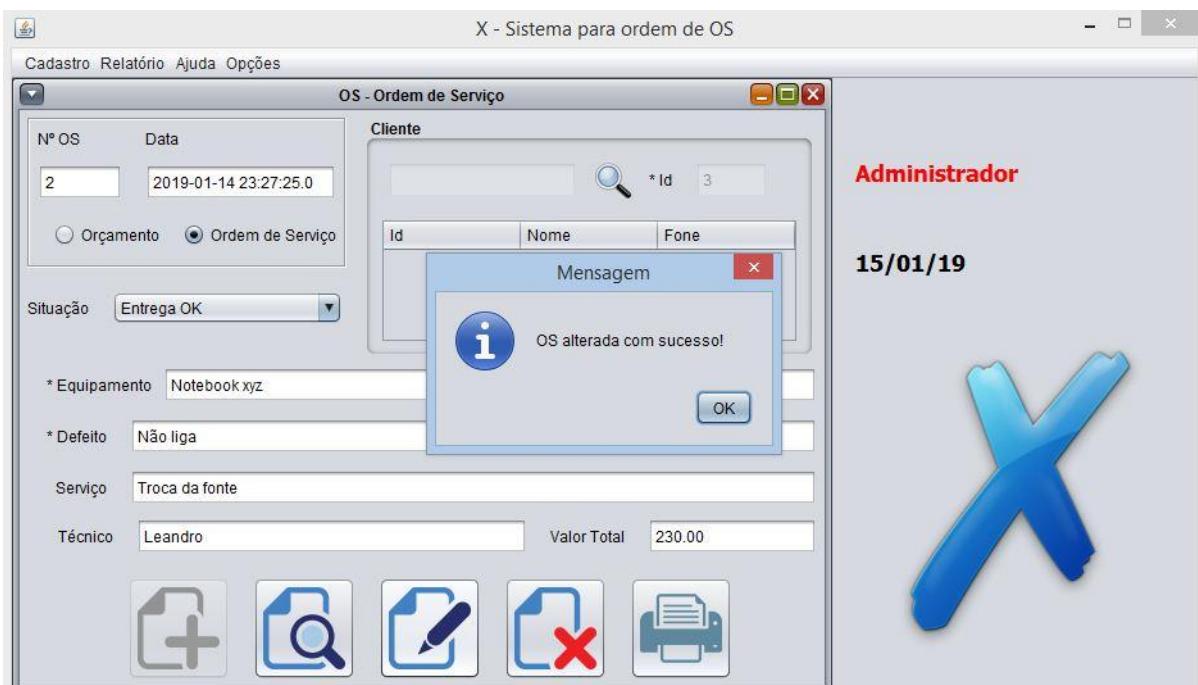
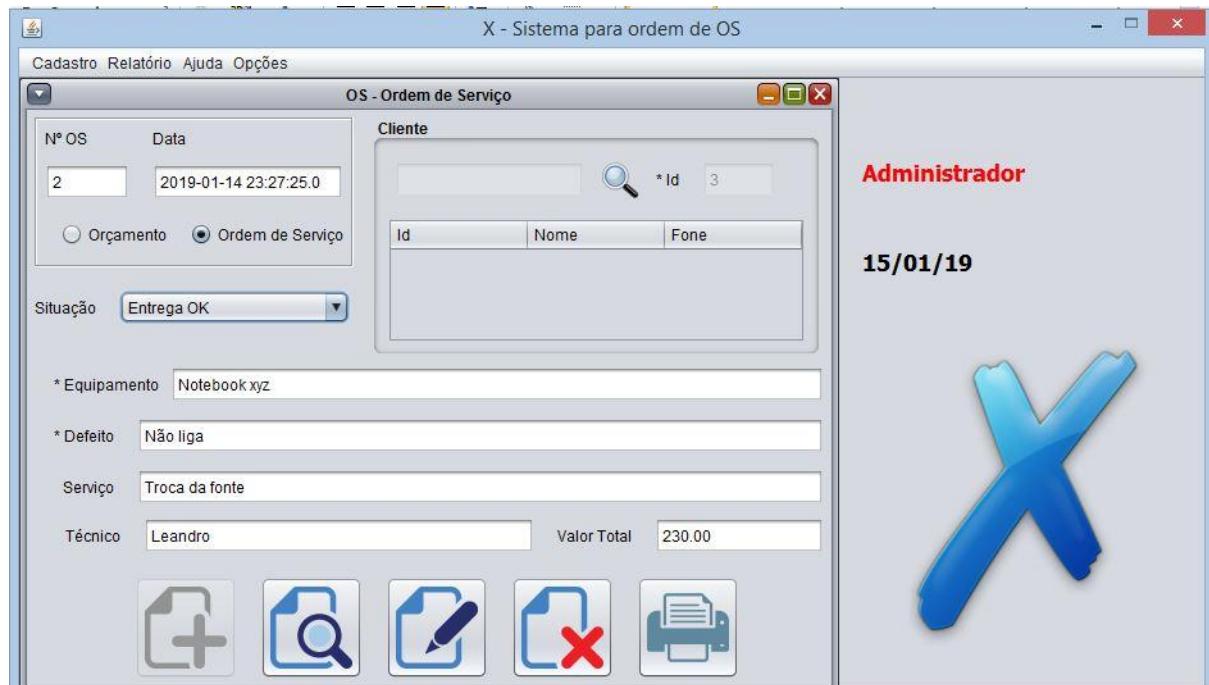
```
// Método para alterar uma OS
private void alterar_os() {
    String sql = "update tbos set tipo=?, situacao=?, equipamento=?, defeito=?, servico=?, tecnico=?, valor=? where os=?";
    try {
        pst = conexao.prepareStatement(sql);
        pst.setString(1, tipo);
        pst.setString(2, cboOsSit.getSelectedItem().toString());
        pst.setString(3, txtOsEquip.getText());
        pst.setString(4, txtOsDef.getText());
        pst.setString(5, txtOsServ.getText());
        pst.setString(6, txtOsTec.getText());
        // .replace substitui a vírgula pelo ponto
        pst.setString(7, txtOsValor.getText().replace(",", "."));
        pst.setString(8, txtOs.getText());
        // Validação dos campos obrigatórios
        if (((txtClild.getText().isEmpty()) || (txtOsEquip.getText().isEmpty()) || (txtOsDef.getText().isEmpty())))
            JOptionPane.showMessageDialog(null, "Preencha todos os campos obrigatórios!");
        else {
            // A linha abaixo atualiza a tabela tbclientes com os dados do formulário
            // A estrutura abaixo é usada para confirmar os dados na tabela
            int adicionado = pst.executeUpdate();
            // A linha abaixo serve de apoio ao entendimento da lógica
            // System.out.println(adicionado);
            if (adicionado > 0) {
                JOptionPane.showMessageDialog(null, "OS alterada com sucesso!");
                // As linhas abaixo limpam os campos
                txtOs.setText(null);
                txtData.setText(null);
                txtClild.setText(null);
                txtOsEquip.setText(null);
                txtOsDef.setText(null);
                txtOsServ.setText(null);
                txtOsTec.setText(null);
                txtOsValor.setText(null);
                // habilitar os objetos
                btnOsAdicionar.setEnabled(true);
                txtCliPesquisar.setEnabled(true);
                tblClientes.setVisible(true);
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

```

private void btnOsAlterarActionPerformed(java.awt.event.ActionEvent evt) {
    // Chamando o método alterar_os
    alterar_os();
}

```

Alterar OS nº 2 de "Orçamento" para "OS" e de "Aguardando aprovação" para "Entrega OK":



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	os	data_os	tipo	situacao	equipamento	defeito	servico	tecnico	valor	iddi
▶	1	2019-01-12 06:42:56			Notebook	Não liga	Troca da fonte	Zé	87.50	1
	2	2019-01-14 23:27:25	OS	Entrega OK	Notebook xyz	Não liga	Troca da fonte	Leandro	230.00	3
*	HULL	NULL	NULL	NULL	HULL	NULL	NULL	NULL	NULL	NULL

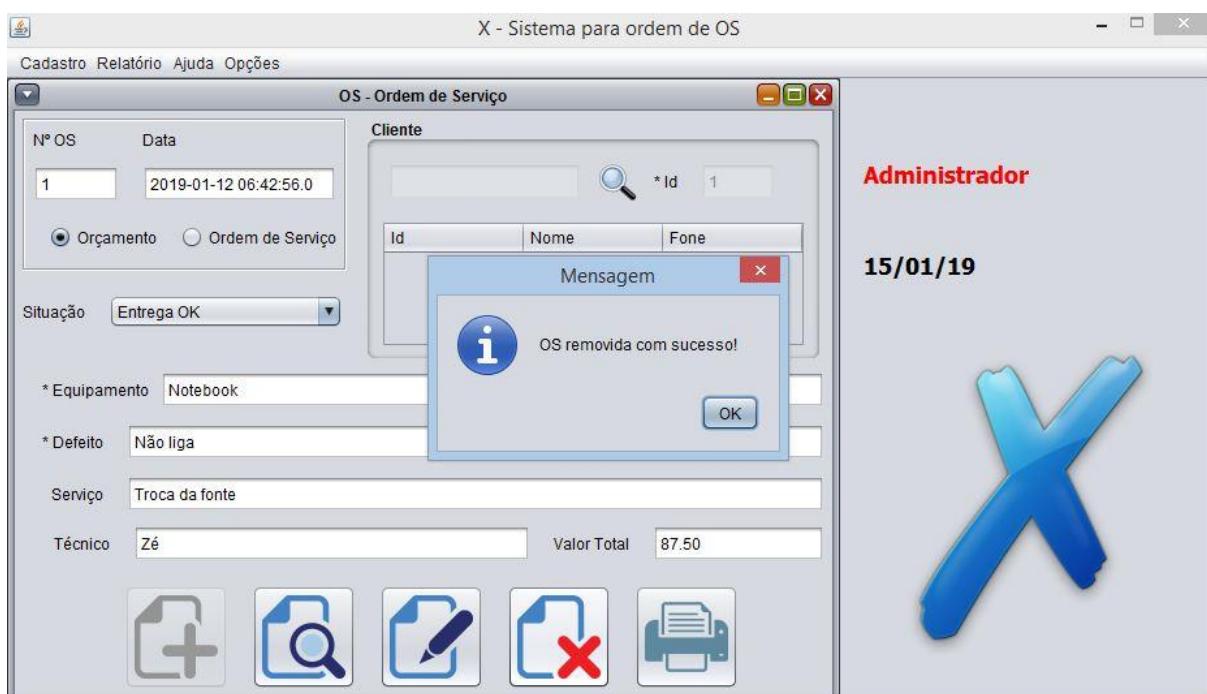
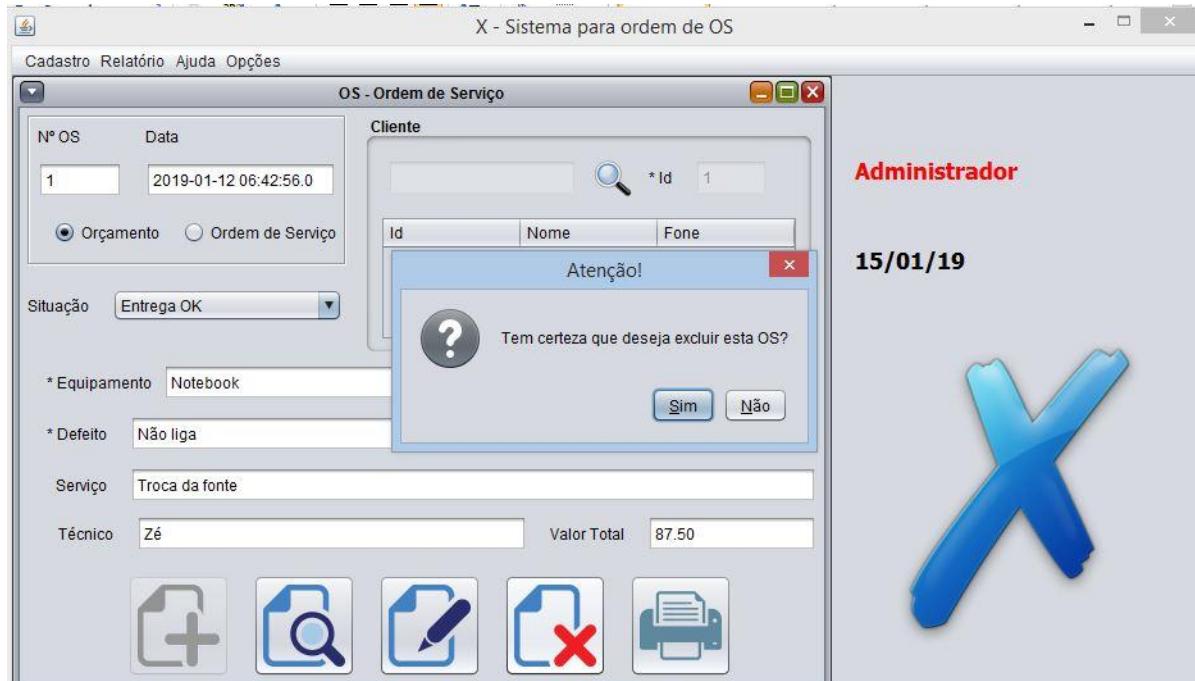
## Excluindo uma OS

### **excluir\_os()**

```
// Metodo para excluir uma OS
private void excluir_os() {
    int confirma = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja excluir
esta 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!");
                txtOsEquip.setText(null);
                txtOsDef.setText(null);
                txtOsServ.setText(null);
                txtOsTec.setText(null);
                txtOsValor.setText(null);
                // habilitar os objetos
                btnOsAdicionar.setEnabled(true);
                txtCliPesquisar.setEnabled(true);
                tblClientes.setVisible(true);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}

private void btnOsExcluirActionPerformed(java.awt.event.ActionEvent evt) {
    // chamando o método excluir_os
    excluir_os();
}
```

Consulte os dados da OS nº 1 e clique no botão "Remover":



# CRIANDO RELATÓRIOS NO JAVA

## Baixando, instalando e configurando o iReport

- Baixar o aplicativo iReport no link abaixo:

Download iReport:

<http://community.jaspersoft.com/project/ireport-designer/releases>

- Instalar o aplicativo.

OBS.: Não funciona com o Java 8.

Na verdade ele aceita as funcionalidades do java 8, mas ele precisa da JRE 7 para que funcione. Faça o download do JRE 7 e configure o arquivo ireport.conf de:

#jdkhome="/path/to/jdk"

para: jdkhome="C:\Program Files (x86)\Java\jre7"

No MySQL WorkBench:

```
use dbinfox;
```

```
select * from tbclientes;
```

-- A instrução abaixo seleciona e ordena por nome todos os clientes cadastrados

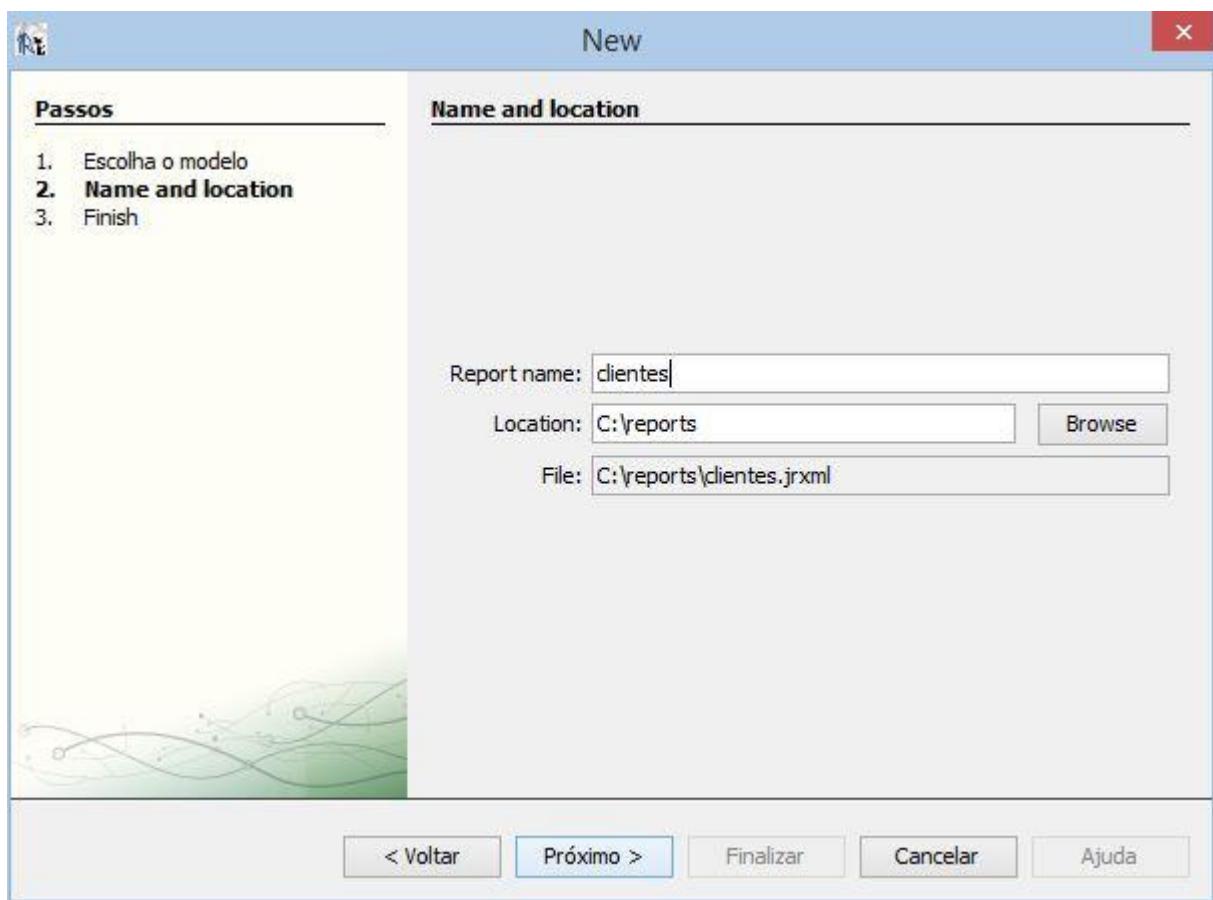
```
select * from tbclientes order by nomecli;
```

- Criar, no diretório raiz (C:), uma pasta chamada "**reports**" para armazenar os relatórios:

## Criando o arquivo de relatório com o iReport

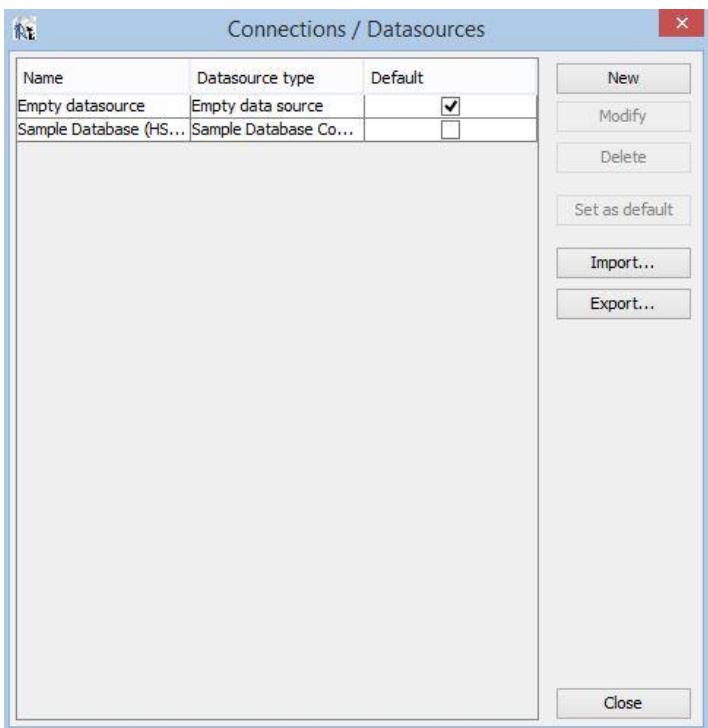
- Abrir o iReport e selecionar o menu:

Arquivo → New → blanck A4 → Open the template

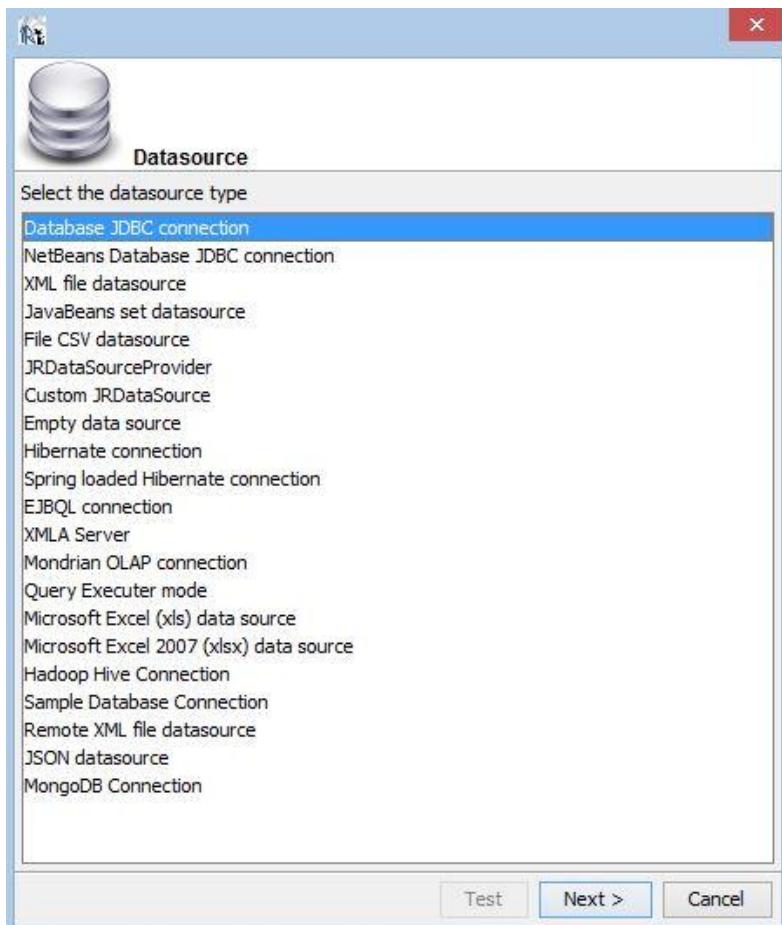


- Clique no botão "Próximo" e depois no botão "Finalizar"
- Será criado o arquivo **clientes.jrxml** na pasta C:\reports
- O iReport será aberto no modo "**Design**".
- O primeiro passo é ligar o iReport com o banco de dados que estamos usando.
- Clique no ícone "Report Datasources"

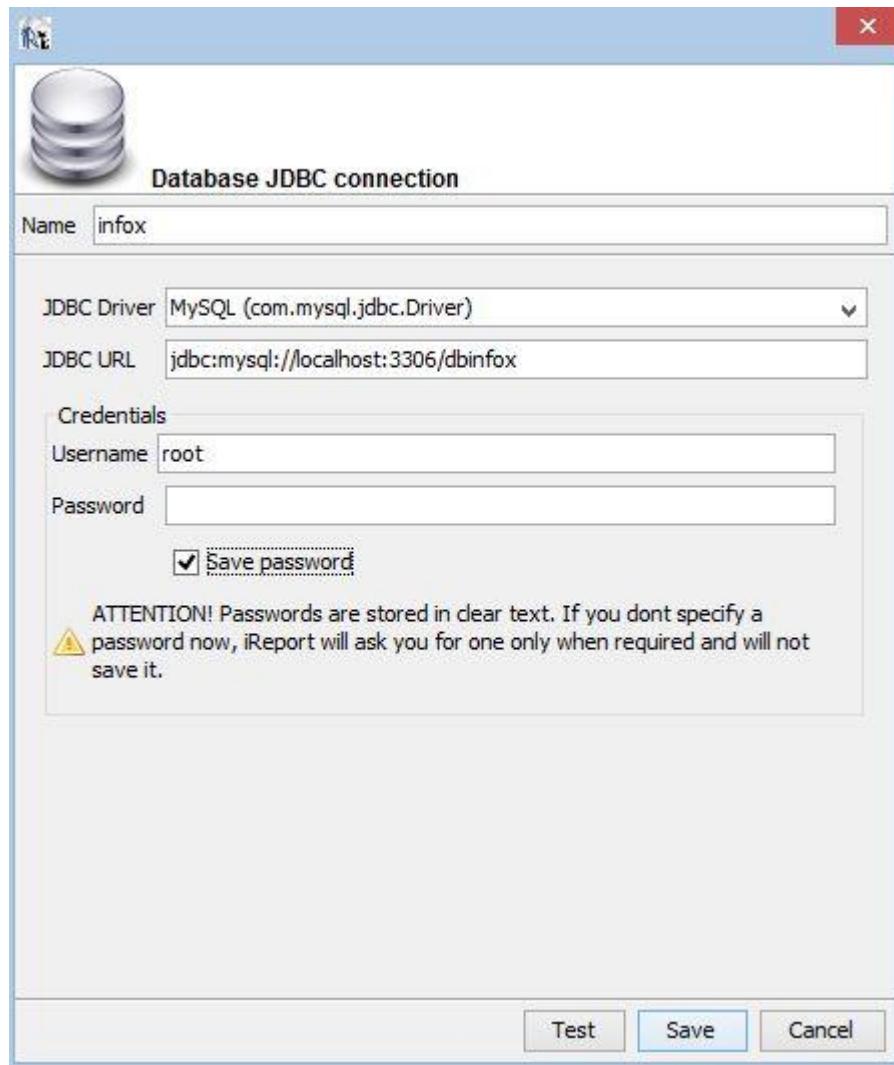




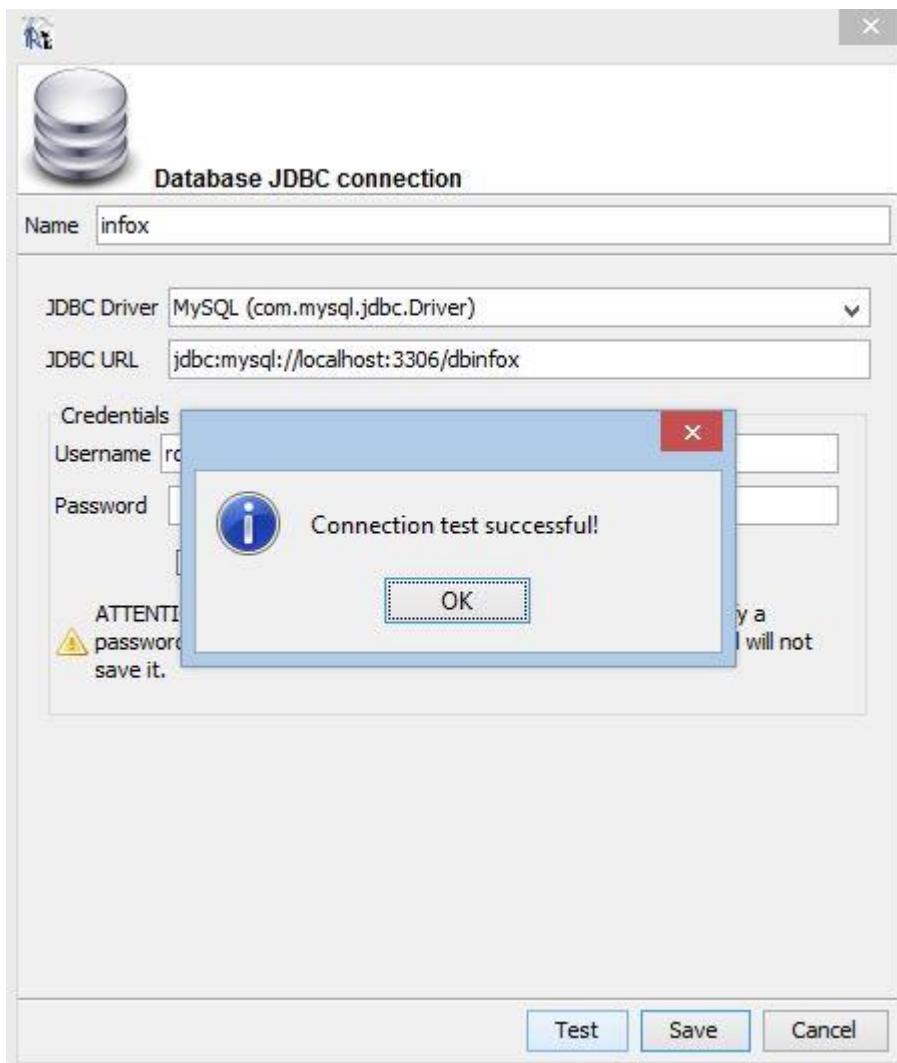
- Clique no botão "New"



- Selecione a primeira opção (Database JDBC Connection) e clique no botão "Next".
- Preencha com os dados abaixo:

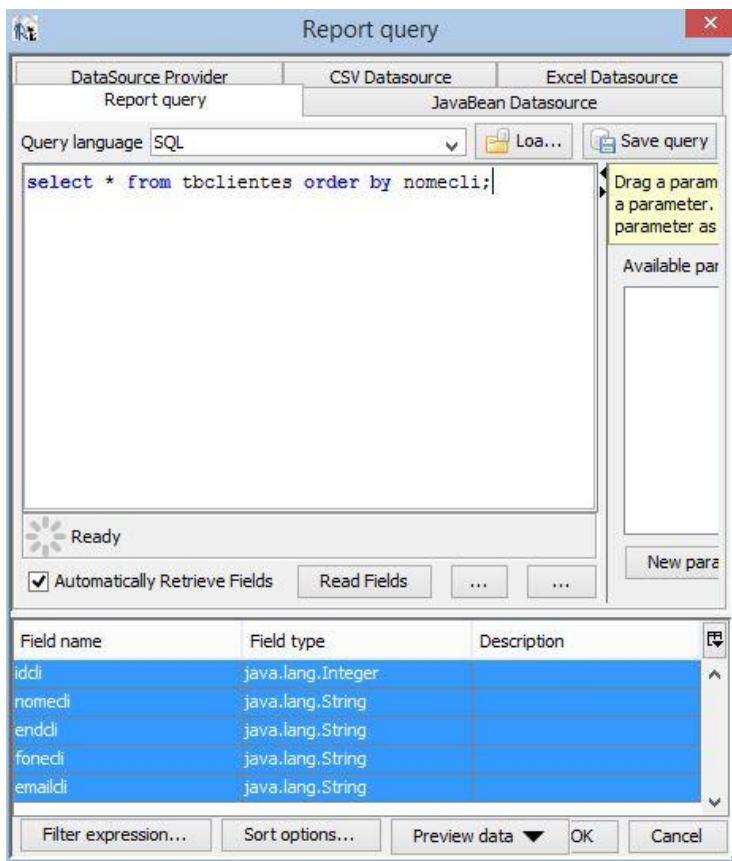


- Clique no botão "Test".

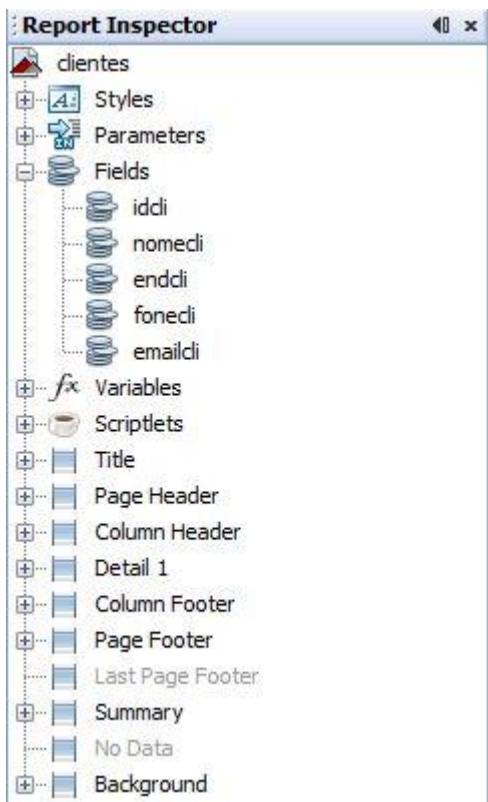


- Clique no botão "Save":

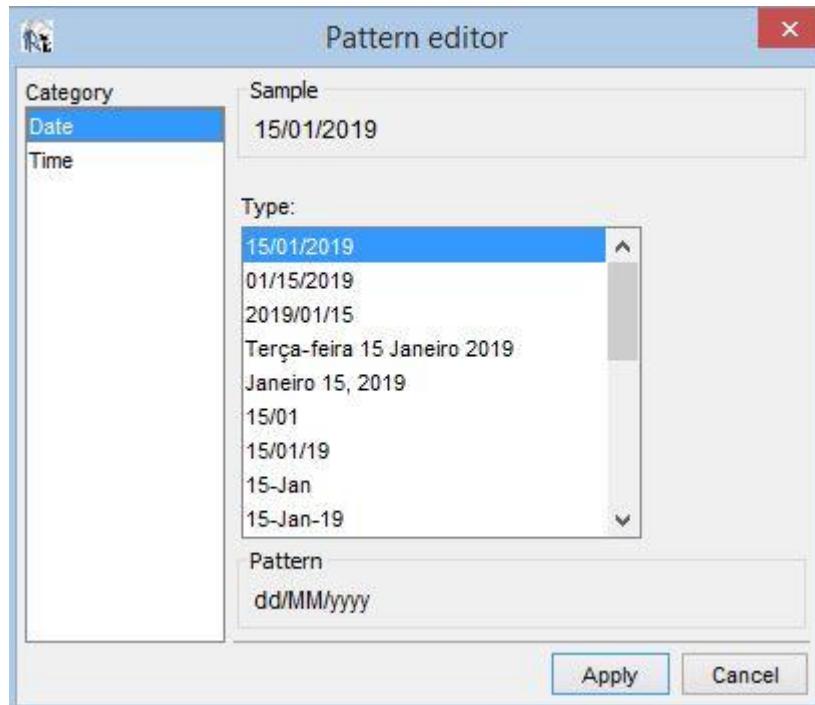
- Clique no botão que se encontra do lado direito de "Preview" e insira o comando SQL e clique no botão "OK":



- Em "Fields" aparecerão todos os campos da tabela:



- Em "Paleta" arraste o elemento "Static Text" para a área "Title" (lado esquerdo). E dentro do elemento digite "Clientes".
- Alterar o tamanho da fonte de 10 para 22.
- Em "Tools" arraste o elemento "Current Date" para a área "Title" (lado direito).



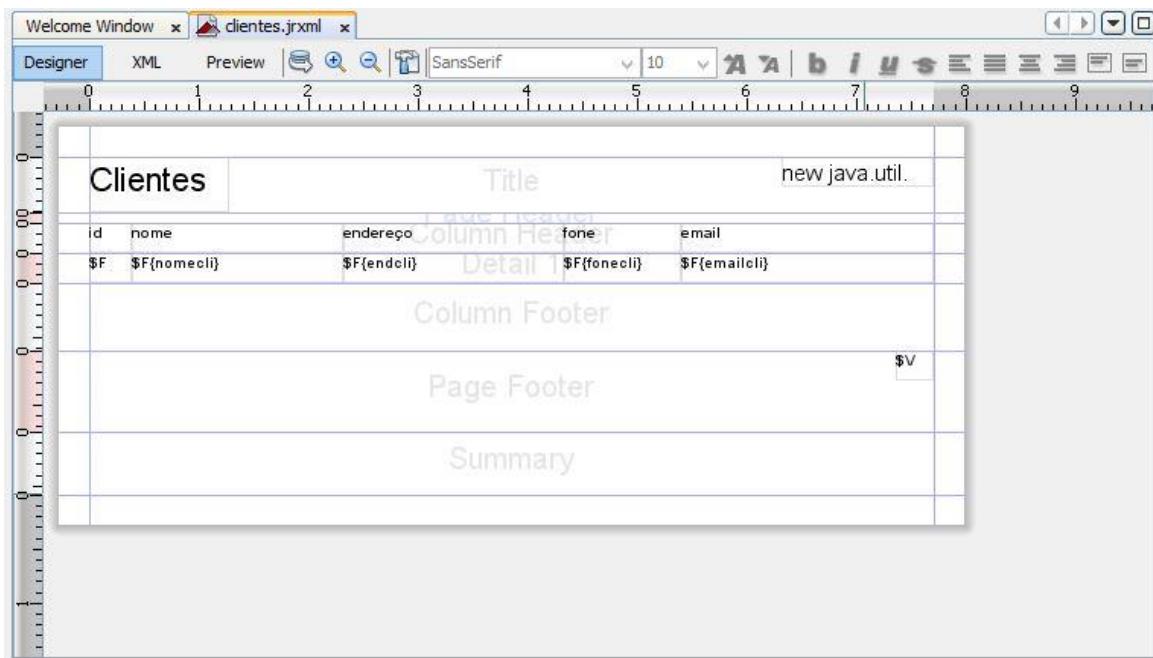
- Alterar o tamanho da fonte de 10 para 15.
- Arrastar os campos (Fields) da tabela para a área "Detail 1"

- Selecionar todos os campos em "Column Header" e colocá-los em negrito.

- Para visualizar como está ficando, clique em "Preview":

id	nome	endereço	fone	email
3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
4	Leandro Ramos	Rua Tuiuti, 2500	3900-0000	professorjoseassis@gmail.com
1	Linus Torvalds	Rua Tux,2015	9999-9999	linus@linux.com

- Arrastar um elemento "Page Number" para o lado direito da área "Column Footer".



- Clique no botão "Compile Report".



- Será criado o arquivo "clientes.jasper"
- Importar todas as bibliotecas necessárias do iReport para o NetBeans:

Bibliotecas → Adicionar JAR/Pasta

- Selecionar todas as bibliotecas da pasta:

C:\Program Files (x86)\Jaspersoft\iReport-5.6.0\ireport\modules\ext

- E clicar no botão "**Abrir**"

## Relatório de Clientes

- No arquivo "TelaPrincipal.java", no menu "Relatórios" incluir o submenu "Clientes" com nome de variável "menRelCli" e tecla de atalho "Alt-R"
- Arrastar este sub-menu para o topo do menu.



- Com o botão direito do mouse, criar o evento ao ser clicado:

Eventos → Action → actionPerformed

```
private void menRelCliActionPerformed(java.awt.event.ActionEvent evt) {  
    // Gerando um relatório de clientes  
    int confirma = JOptionPane.showConfirmDialog(null,"Confirma a impressão deste  
    relatório?","Atenção!",JOptionPane.YES_NO_OPTION);  
    if (confirma == JOptionPane.YES_OPTION) {  
        // Imprimindo o relatório com o framework JasperReport  
        try {  
            // Usando a classe JasperPrint para preparar a impressão de um relatório  
            // JasperPrint print =  
            JasperFillManager.fillReport("C:/reports/clientes.jasper",null,conexao);  
            // A linha abaixo exibe o relatório através da classe JasperViewer  
            // JasperViewer.viewReport(print,false);  
  
            } catch (Exception e) {  
                JOptionPane.showMessageDialog(null,e);  
            }  
    }  
}
```

Incluir no arquivo TelaPrincipal.java:

```
package br.com.infox.telas;

import br.com.infox.dal.ModuloConexao;
import java.text.DateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import java.sql.*;

public class TelaPrincipal extends javax.swing.JFrame {

    Connection conexao = null;
    /**
     * Creates new form TelaPrincipal
     */
    public TelaPrincipal() {
        initComponents();
        conexao = ModuloConexao.conector();
    }

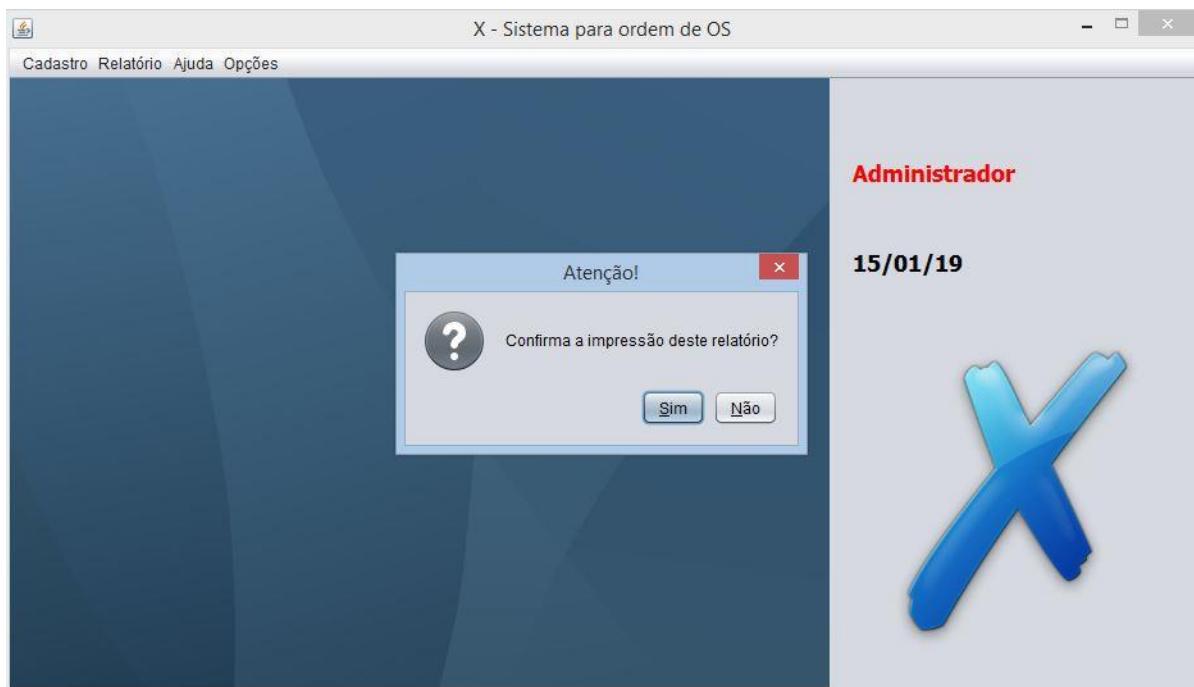
    private void menRelCliActionPerformed(java.awt.event.ActionEvent evt) {
        // Gerando um relatório de clientes
        int confirma = JOptionPane.showConfirmDialog(null,"Confirma a impressão deste relatório?","Atenção!",JOptionPane.YES_NO_OPTION);
        if (confirma == JOptionPane.YES_OPTION) {
            // Imprimindo o relatório com o framework JasperReport
            try {
                // Usando a classe JasperPrint para preparar a impressão de um relatório
                JasperPrint print = JasperFillManager.fillReport("C:/reports/clientes.jasper",null,conexao);
                // A linha abaixo exibe o relatório através da classe JasperViewer
                // JasperViewer.viewReport(print,false);

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

Incluir as linhas de importação:

```
package br.com.infox.telas;

import br.com.infox.dal.ModuloConexao;
import java.text.DateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import java.sql.*;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
```

A screenshot of a JasperReports viewer window. The title bar shows various icons and the date "15/01/2019". The main content area displays a table titled "Clientes" with the following data:

id	nome	endereço	fone	email
3	João da Silva	Rua Vitória, 56	5678-9009	joao@hotmail.com
6	Joaquim Jorge	Av. Sumaré, 2009 - casa 2	2908-0987	joaquim@yahoo.com
2	José de Assis Filho	Rua do Gasometro, 1234	9999-9999	professorjoseassis@gmail.com
5	Juliana Paes	Rua Santa Luzia, 456	7890-0987	ju@globo.com
4	Leandro Ramos	Rua Tuiuti, 2500	3900-0000	professorjoseassis@gmail.com
1	Linus Torvalds	Rua Tux, 2015	9999-9999	linus@linux.com

# Relatório de Serviços

No MySQL WorkBench:

```
use dbinfox;
```

- O bloco de instruções abaixo faz a seleção e união de dados de duas tabelas
- OSER é uma variável que contém os campos desejados da tabela OS
- CLI é uma variável que contém os campos desejados da tabela Clientes

```
select
OSER.os,data_os,tipo,situacao, equipamento, valor,
CLI.nomecli,fonecli
from tbos as OSER
inner join tbclientes as Cli
on (CLI.idcli = OSER.idcli);
```

Adicione a OS Abaixo:

The screenshot shows a Windows application window titled "X - Sistema para ordem de OS". The main area contains a form for creating a service order (OS). The form includes fields for "Nº OS" (with a placeholder box), "Data" (with a placeholder box), "Orçamento" (radio button selected), "Ordem de Serviço" (radio button), "Situacao" (dropdown menu showing "Aguardando peças"), "Equipamento" (text input with placeholder "Impressora modelo z"), "Defeito" (text input with placeholder "Não puxa o papel"), "Serviço" (text input with placeholder "Trocar o sensor de papel"), "Técnico" (text input with placeholder "Leandro"), and "Valor Total" (text input with placeholder "85,00"). Below the form are five icons: a green plus sign inside a document, a magnifying glass over a document, a pencil inside a document, a red X inside a document, and a printer. To the right of the form is a sidebar with the title "Administrador" and the date "15/01/19". A large blue "X" is overlaid on the right side of the window.

	os	data_os	tipo	situacao	equipamento	valor	nomecli	fonecli
▶	2	2019-01-14 23:27:25	OS	Entrega OK	Notebook xyz	230.00	João da Silva	5678-9009
	3	2019-01-15 12:48:59	Orçamento	Aguardando peças	Impressora modelo z	85.00	Joaquim Jorge	2908-0987

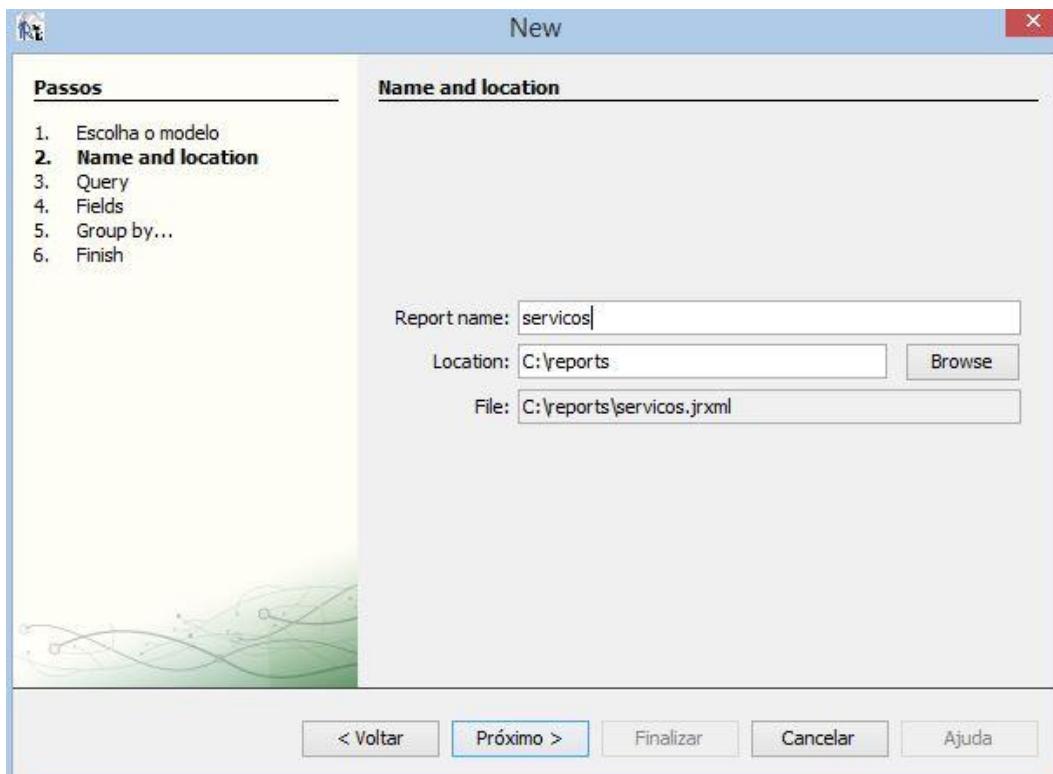
## Emissão de relatório de serviços / impressão de OS

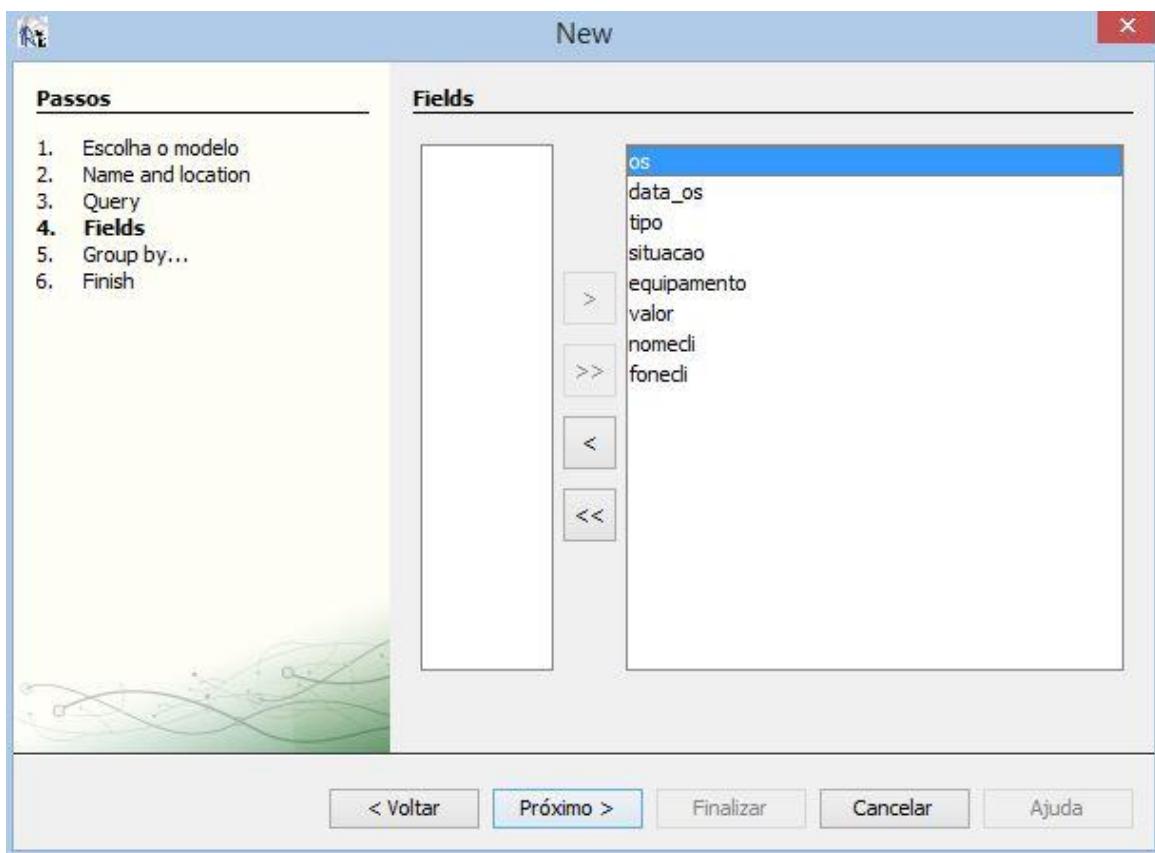
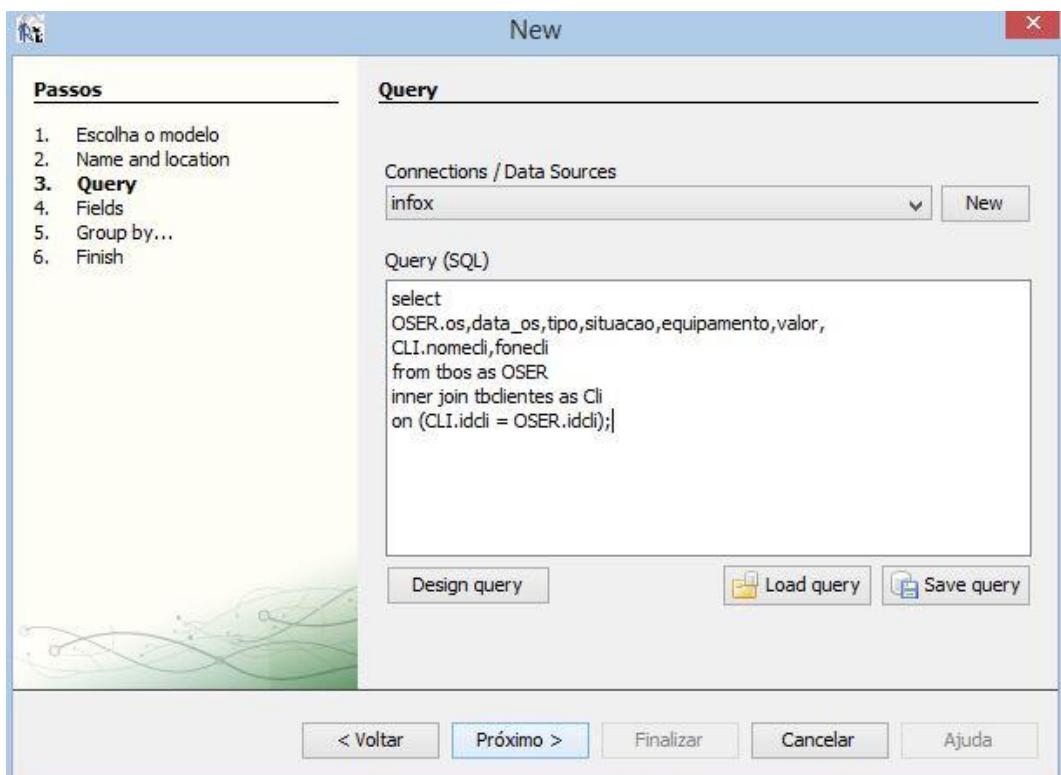
- O bloco de instruções abaixo faz a seleção e união de dados de duas tabelas
- OSER é uma variável que contém os campos desejados da tabela OS
- CLI é uma variável que contém os campos desejados da tabela Clientes

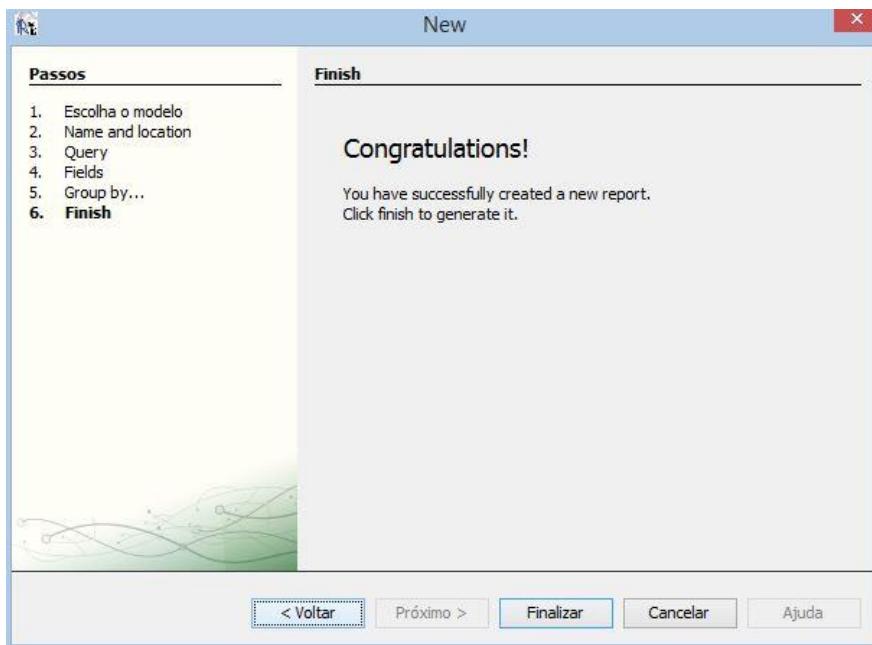
```
select
OSER.os,data_os,tipo,situacao, equipamento, valor,
CLI.nomecli,fonecli
from tbos as OSER
inner join tbclientes as CLI
on (CLI.idcli = OSER.idcli);
```

## Criando o relatório de serviços com o iReport

Crie o novo relatório usando o assistente de construção







Welcome Window x servicos.jrxml x

Designer XML Preview SansSerif 10 b i u s =

Serviços								Title	new java.util.
OS	Data	Tipo	Situação	Equipamento	Valor	Cliente	Fone		
\$F	\$F	\$F{tipo}	\$F{situacao}	\$F{equipamento}	\$F{valor}\$F{nomecli}	\$F{fonecli}			

Column Footer

Page Footer "Page "+\$V" "+\$V

Summary

### Serviços

15/01/2019

OS	Data	Tipo	Situação	Equipamento	Valor	Cliente	Fone
2	14/01/19	OS	Entrega OK	Notebook xyz	230.00	João da Silva	5678-9009
3	15/01/19	Orçamento	Aguardando peças	Impressora modelo z	85.00	Joaquim Jorge	2908-0987

Report Problems Window

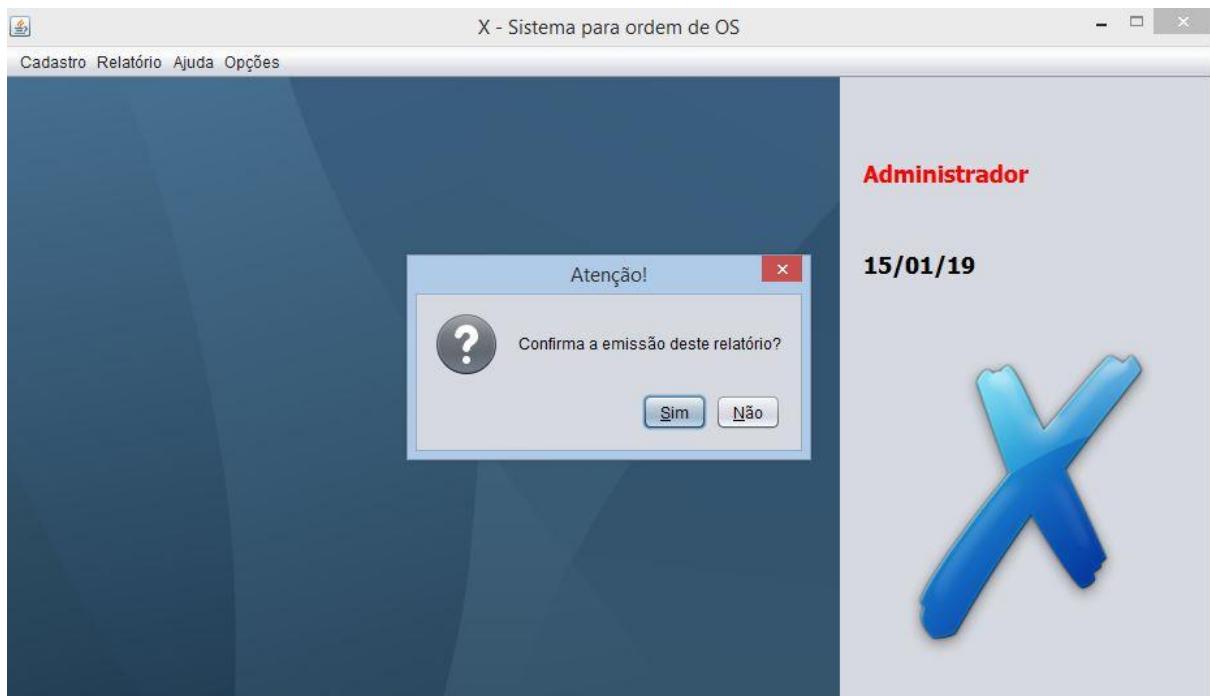
iReport console Finished [servicos.jrxml]

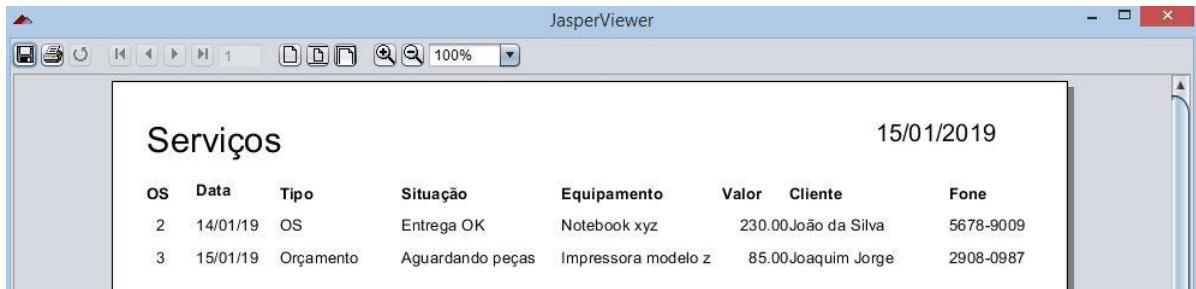
Compiling to file... C:\reports\servicos.jasper

Compilation running time: 672!

No arquivo TelaPrincipal.java

```
private void MenRelSerActionPerformed(java.awt.event.ActionEvent evt) {  
    // Gerando um relatório de serviços  
    int confirma = JOptionPane.showConfirmDialog(null,"Confirma a emissão deste  
    relatório?", "Atenção!",JOptionPane.YES_NO_OPTION);  
    if (confirma == JOptionPane.YES_OPTION) {  
        // Emitindo o relatório com o framework JasperReport  
        try {  
            // Usando a classe JasperPrint para preparar a impressão de um relatório  
            JasperPrint print  
            JasperFillManager.fillReport("C:/reports/servicos.jasper",null,conexao);  
            // A linha abaixo exibe o relatório através da classe JasperViewer  
            JasperViewer.viewReport(print,false);  
  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(null,e);  
        }  
    }  
}  
  
private void btnOslImprimirActionPerformed(java.awt.event.ActionEvent evt) {  
    // Chamando o método para imprimir uma OS  
    imprimir_os();  
}
```





## Criando o relatório de uma OS com o iReport

```
select * from tbos where os = $P{os}
```

No arquivo TelaOS.java:

```
// Método para imprimir uma OS
private void imprimir_os(){
    // Imprimindo uma OS
    int confirma = JOptionPane.showConfirmDialog(null,"Confirma a impressão desta
OS?","Atenção!",JOptionPane.YES_NO_OPTION);
    if (confirma == JOptionPane.YES_OPTION) {
        // Emitindo o relatório com o framework JasperReport
        try {
            // Usando a classe HashMap para criar um filtro
            HashMap filtro = new HashMap();
            filtro.put("os",Integer.parseInt(txtOs.getText()));
            // Usando a classe JasperPrint para preparar a impressão desta OS
            JasperPrint print
            JasperFillManager.fillReport("C:/reports/os.jasper",filtro,conexao);
            // A linha abaixo exibe o relatório através da classe JasperViewer
            JasperViewer.viewReport(print,false);

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

Welcome Window > os.jrxml

Designer XML Preview SansSerif 10 0 1 2 3 4 5 6 7 8

**Ordem de Serviço** \$F{os}

Data da OS \$F{data\_os}

Técnico: \$F{tecnico}

Valor: \$F{valor}

**Column Footer**

"Page "+\$V" " + \$V

**Summary**



Tipo	Situação	Equipamento	Defeito	Serviço
\$F{tipo}	\$F{situacao}	\$F{equipamento}	\$F{defeito}	\$F{servico}

Terça-feira 15 Janeiro 2019

**Ordem de Serviço** 2

Data da OS 14/01/19 23:27

Técnico: Leandro

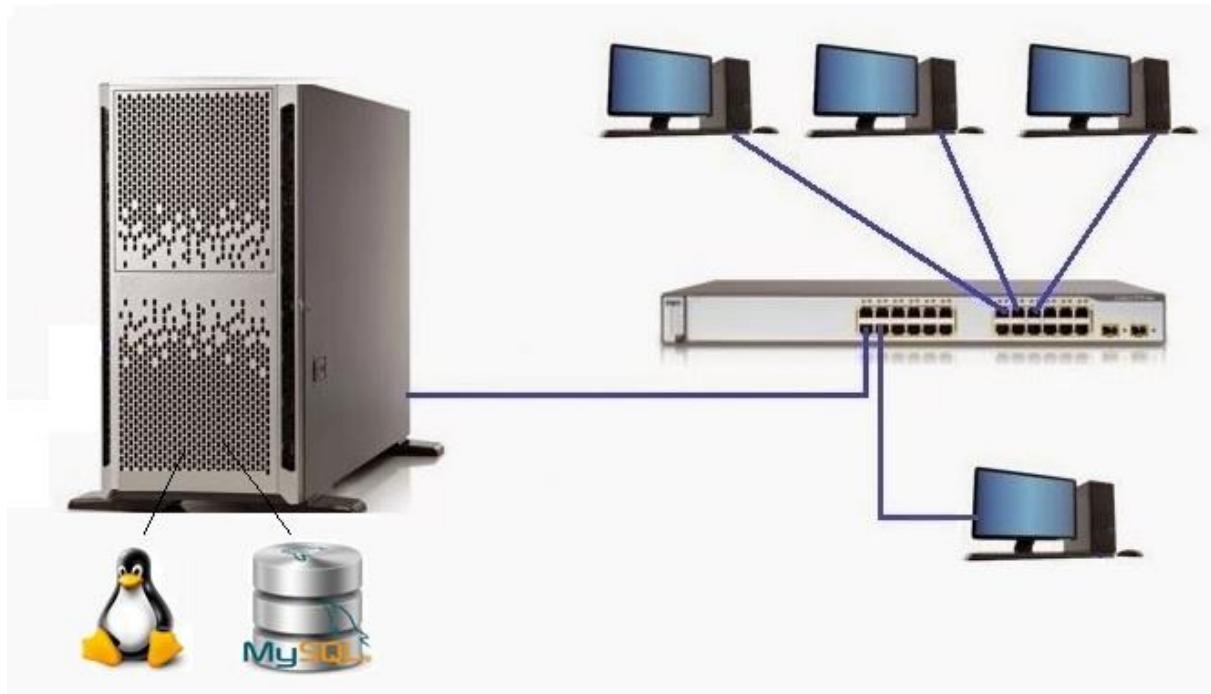
Valor: 230.00

Tipo	Situação	Equipamento	Defeito	Serviço
OS	Entrega OK	Notebook xyz	Não liga	Troca da fonte



# SERVIDOR DE BANCO DE DADOS

Esta aula traz ao aluno uma oportunidade de negócio, onde o projeto pode ser implementado em um cliente real e funcionar em rede. A ideia é migrar o banco de dados do Xampp para um servidor Linux com o MySQL Server.

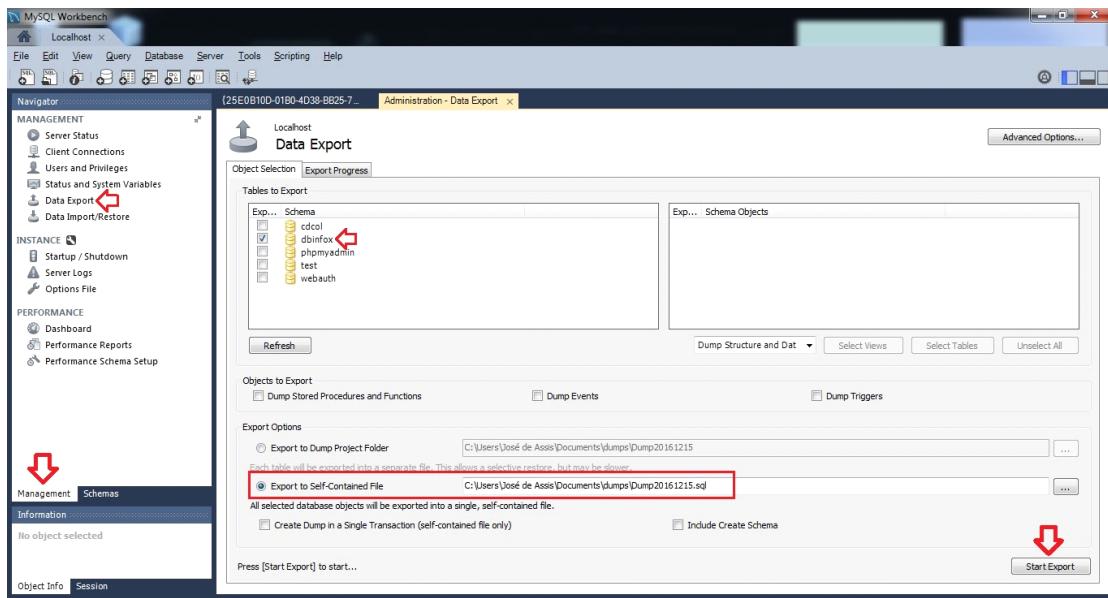


<http://www.aulaead.com/courses/curso-gratis-servidor-linux>

# Documentação e Implementação do projeto

## 1ª Etapa – Banco de Dados

Na aula 3 aprendemos a documentar e fazer back-up do banco de dados. O primeiro passo é a gente criar uma back-up do banco (estrutura e dados) usando o Workbench.



Copie o arquivo gerado para o servidor de banco de dados.  
No servidor de banco de dados, crie um banco de dados chamado dbinfox

```
root@debianze:/home/joseassis# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.53-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database dbinfox;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

Restaure o arquivo de back-up que você copiou para este servidor:

```
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
root@debianize:/home/joseassis# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.53-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

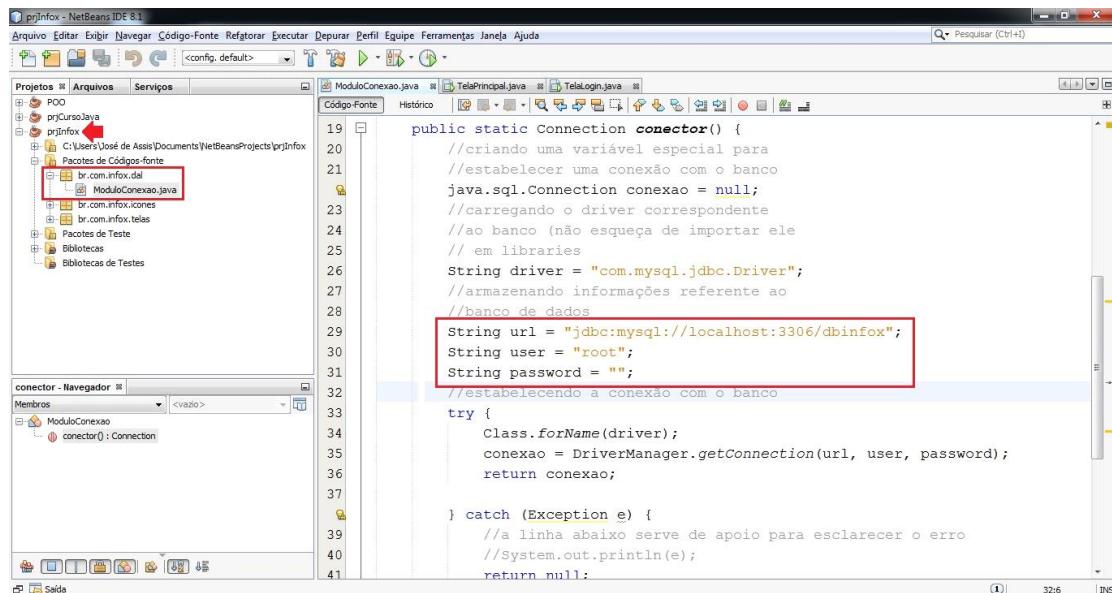
mysql> create database dbinfox;
Query OK, 1 row affected (0.00 sec)

mysql> Bye
root@debianize:/home/joseassis# mysql -u root -p dbinfox < Dump20161213.sql _
```

## 2ª Etapa – Módulo de Conexão

Abra o projeto e expanda o pacote **br.com.infox.dal**  
Abra o código fonte da classe **ModuloConexao.java**

- Na String **url** mude **localhost** para o **ip** do servidor de banco de dados
- Na String **user** mude o usuário **root** pelo usuário definido no banco de dados
- Na String **password** insira a senha definida no banco de dados



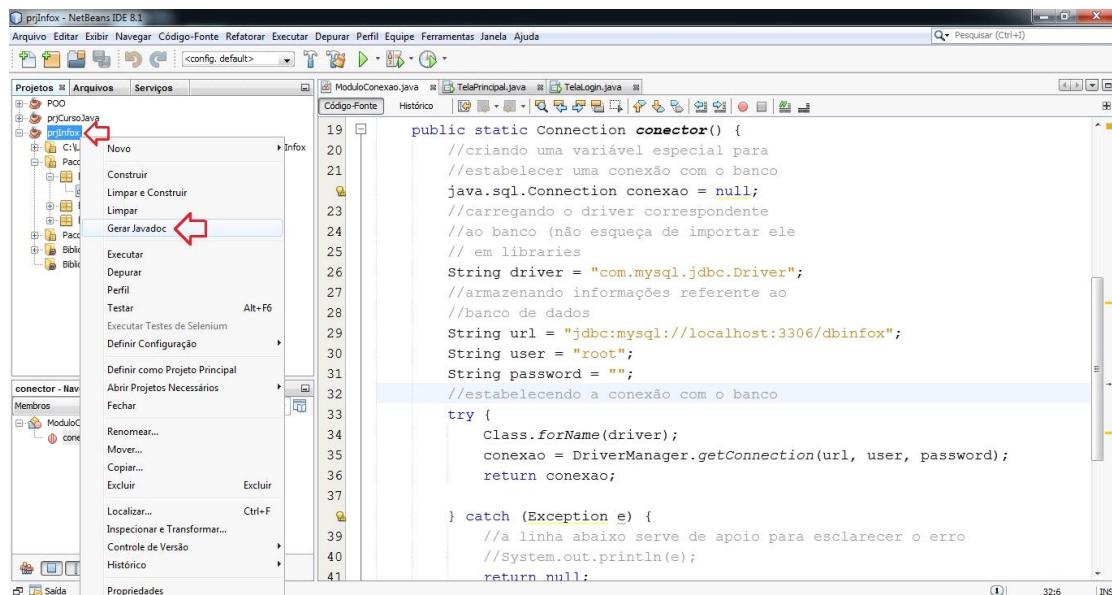
The screenshot shows the NetBeans IDE interface with the project 'pjInfox' open. In the left pane, the project structure is shown with a red box highlighting the package 'br.com.infox.dal'. Inside this package, the file 'ModuloConexao.java' is also highlighted with a red box. The right pane contains the code for 'ModuloConexao.java'. A red box highlights the connection URL definition:

```
String url = "jdbc:mysql://localhost:3306/dbinfox";
```

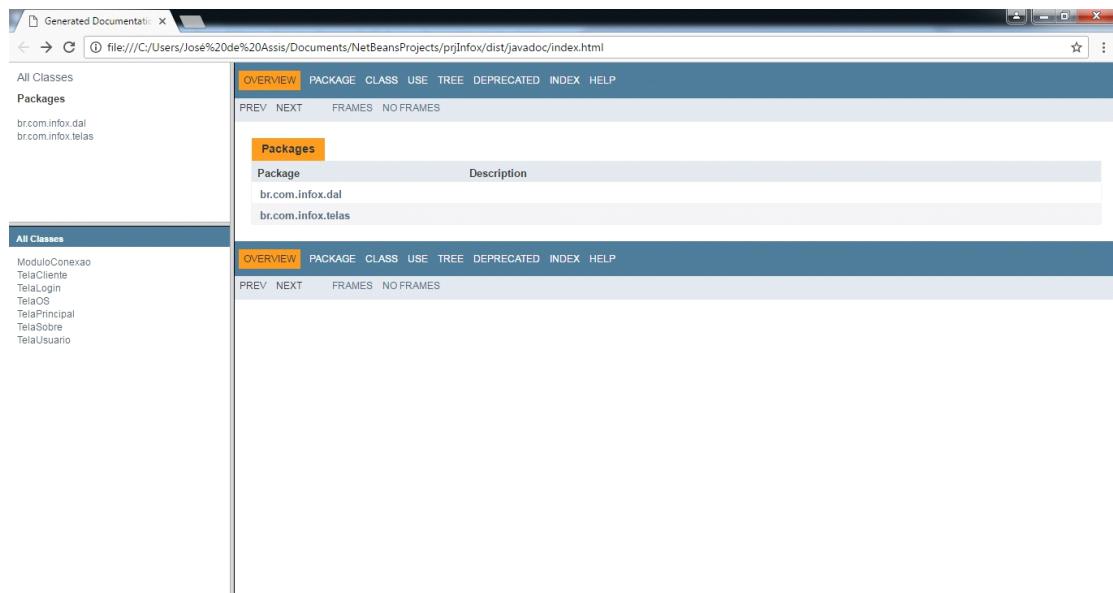
Salve o projeto.

## 3ª Etapa – Documentação do Projeto

Selecione o projeto com o botão direito do mouse e escolha a opção **Gerar Javadoc**

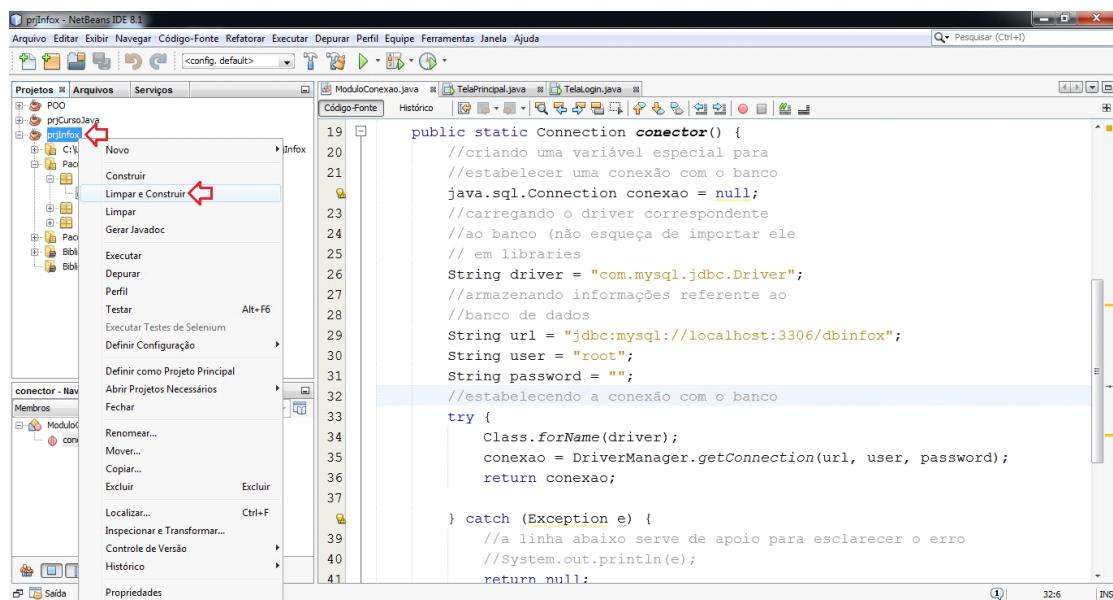


Ao final do processo a documentação será gerada e irá abrir no navegador:

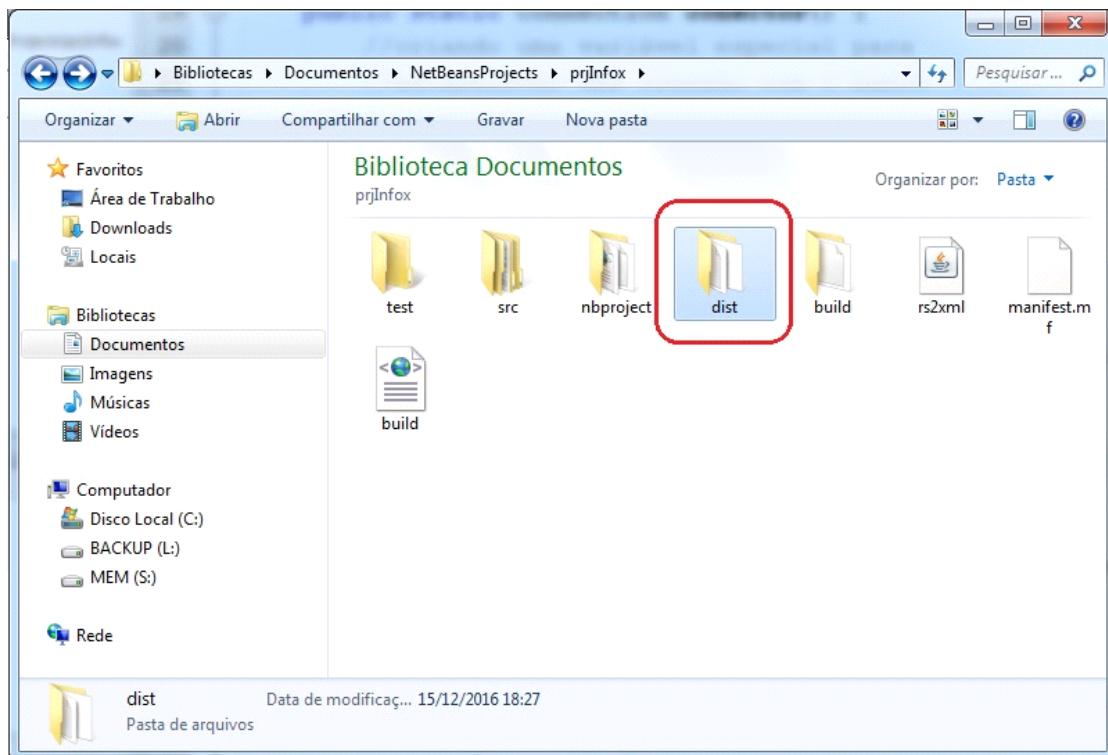


## 4ª Etapa – Publicação do projeto

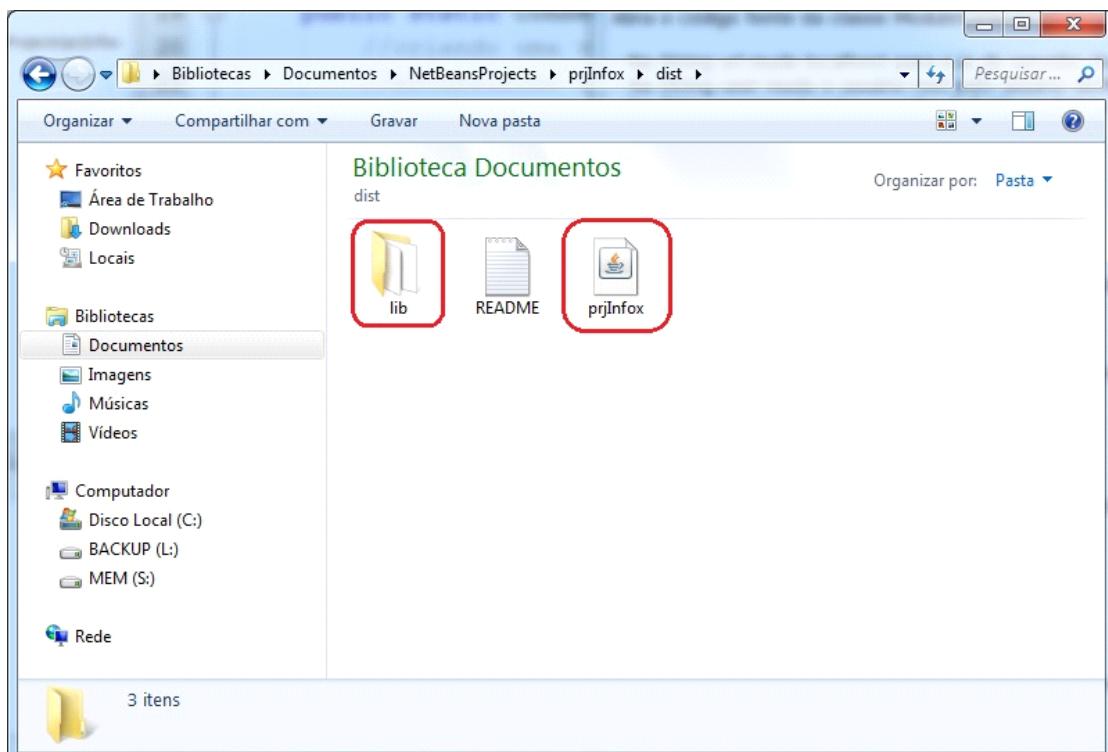
Selecione novamente o projeto com o botão direito do mouse e escolha a opção **Limpar e Construir**



No diretório padrão do projeto, localize a pasta “**dist**”



Dentro desta pasta iremos encontrar o arquivo **.jar** (executável do java) e uma pasta de bibliotecas:



## ATENÇÃO !!!

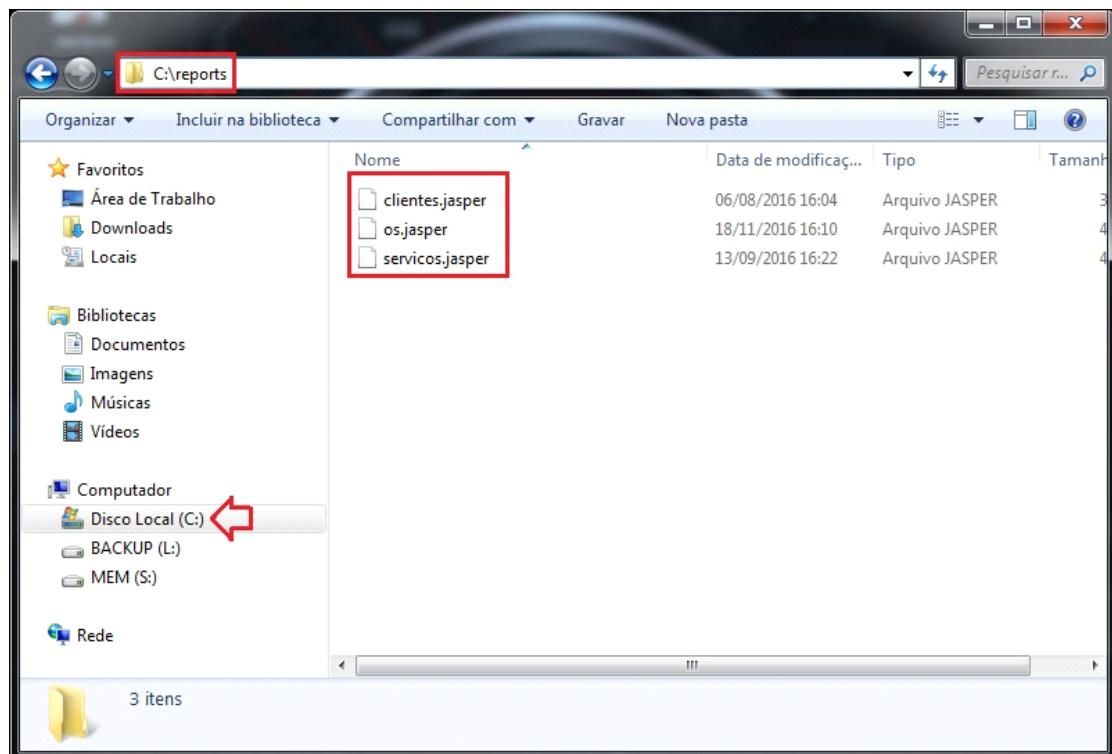
\*\*\* Este dois arquivos precisam estar juntos nos computadores que vão executar este projeto

\*\*\* Também é necessário que os computadores tenham o java(JRE) instalado.

### 5ª Etapa – Relatórios

Nos computadores que vão executar este projeto também é necessário criar uma pasta na unidade C: de nome “**reports**”

É necessário copiar os relatórios .jasper para esta pasta.



Obtenha os ícones e o código fonte deste projeto clicando no link abaixo:

<https://github.com/professorjosedeassis/infoX>