

# netBeans 4.1 primeiros passos



## Versão 1.0

**William Silva**

Este tutorial tem por finalidade fornecer informações básicas para instalação, configuração e execução dos seus primeiros programas em Java com o netBeans IDE 4.1 em seu sistema operacional.

## Índice

- Introdução
- netBeans IDE
- Estrutura de diretórios
- Configurando seu ambiente de trabalho
- Criando projetos
- Conectando banco de dados
- Criando Web Project

## Introdução

Este tutorial é fundamentado em artigos dos profs. Fernando Lozano, Fernando Anselmo, NetBeans IDE Field Guide, e várias informações de evangelistas da linguagem Java e da IDE netBeans, e se propõe a dar uma visão inicial da IDE netBeans e apresentar de agora em diante os primeiros passos para a sua instalação, configuração e usabilidade do ambiente de desenvolvimento, e da interface gráfica. O ideal é que este material seja utilizado por desenvolvedores, alunos e educadores, pois poderão observar o quanto esta IDE é flexível e colaborativa para o desenvolvimento de projetos entre equipes.

Foi-me requerido um considerável esforço pois estou conhecendo Java e a IDE netBeans a pouco tempo, portanto se houver algum erro de código ou explicação favor me contatarem via e-mail. Não pretendo fazer comparativo ou mostrar qual a melhor IDE para aprendizado o objetivo aqui é aliar o recurso da plataforma Java ao ambiente de desenvolvimento integrado.

Meus agradecimentos ao prof. Fernando Lozano pelo carinho, humildade e paciência. Ao Michael Nascimento meu conterrâneo que numa simples conversa de 05 minutos ao celular me fez ver a linguagem java com mais profundidade.

## netBeans Platform

O NetBeans Platform, é o runtime das suas aplicações fornecendo uma base sólida para o desenvolvimento de seus projetos e à criação de "modules". Veremos um pequeno resumo dos principais recursos da Platform.

- ❖ **Interface com o usuário** – Os menus, barras de ferramentas e outros componentes UI-interfaces de apresentação são fornecidos por esta plataforma.
- ❖ **Editor** - Rico em recursos para aplicações visuais Swing ou AWT, para aplicações WEB (Servlets e JSP, JSTL, EJBs), XML com suporte a DTDs e XML Schemas e etc.
- ❖ **Gerenciamento** – As views gerenciam a estrutura do seu projeto localmente ou remotamente, podendo também utilizar Projects, CVS, FTP ou base de dados remota.
- ❖ **Cross-Plataform** – Totalmente escrito em 100% Java, portanto multi-plataforma.
- ❖ **Wizards** - ferramentas de gerenciamento de código, criação de templates, etc.

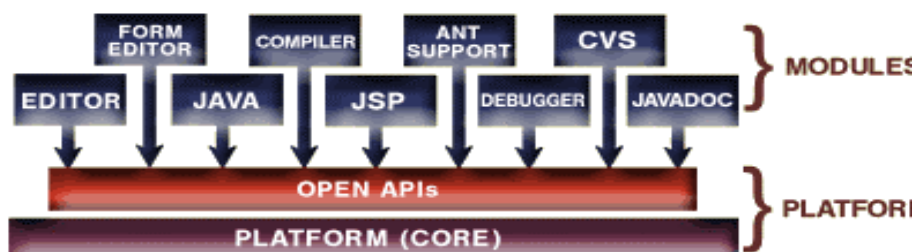


Figura 1

## netBeans IDE

netBeans IDE é um conjunto de bibliotecas, módulos e APIs. formando um ambiente integrado de desenvolvimento visual possibilitando ao desenvolvedor compilar, debugar, efetuar deploying de suas aplicações .

Podemos destacar os seguintes recursos:

- ❖ Debugador e compilador de programas;
- ❖ Auto-completar avançado, depurador de erros, depurador de Servlets;
- ❖ Syntax highlighting à XML,HTML,CSS,JSP,IDL;
- ❖ Suporta linguagens Java, C, C++;
- ❖ Suporte à XML e HTML, JSP, JSTL, Servlets, etc..;
- ❖ Recursos para desenvolvimento EJBs, Web Services;
- ❖ Macros de abreviação;
- ❖ Total suporte ao ANT e TOMCAT integrado na IDE;
- ❖ http Monitor para Monitoramento de suas aplicações WEB;
- ❖ Indentação automática de código disponível,
- ❖ Refatoração básica de código Java;
- ❖ Suporte a Database, Data view, Connection wizard;
- ❖ É possível integrar o módulos e extensões como o Jfluid e J2ME Mobility;
- ❖ Etc.

É um produto open source, 100% Java e possui vários módulos de expansão (modules), que podem ser baixados do próprio site <http://www.netbeans.org>.

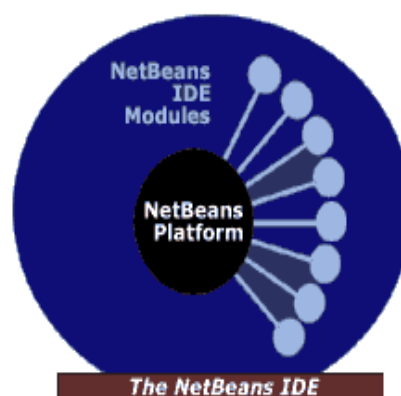


Figura 2

## Estrutura de diretórios

Nesta estrutura somente três pastas serão do nosso interesse.  
 (Figura 4)

- 1- A pasta bin e seu arquivo netBeans.exe, para que a IDE seja inicializada, pois a versão utilizada é sem instalador. (Figura 3)

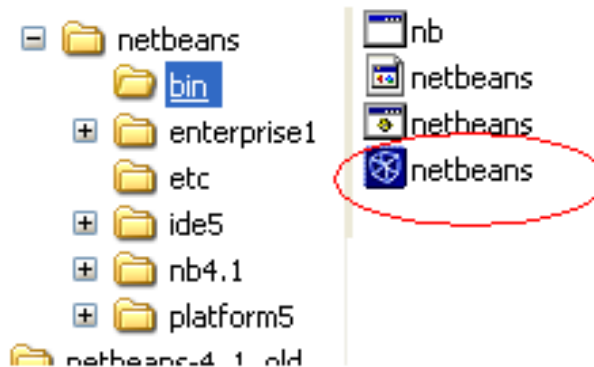


Figura 3

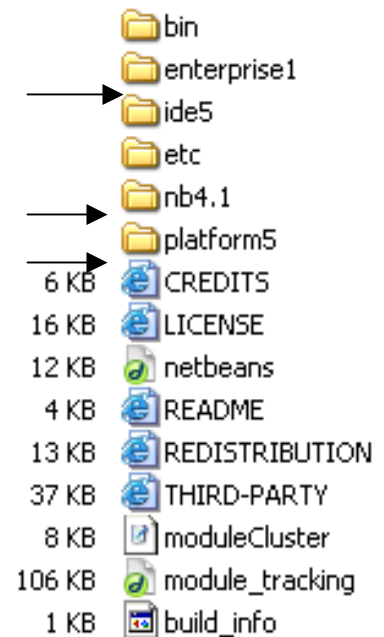


Figura 4

- 2- A pasta **<netbeans>\etc** e seu arquivo **netbeans.conf** para Alterar-mos o *look-and-feel* como veremos logo Abaixo. (Figura 4)

- 3- e a pasta **<netbeans>\nb4.1** e seu arquivo **shortcuts.pdf** que é um pequeno documento com dicas de atalho para aumentar a sua produtividade. (Figura 4)

## Bate bola

Antes de tudo gostaria de falar do meu ambiente de trabalho particular:

Tomcat 5.0.28	C:\Tomcat
SGBD	Firebird 1.5 + IBExpert
SGBD	Caché
UML	Jude UML
NetBeans 4.1	D:\netbeans4.1

02 Micros

Server	IP- 169.254.90.82 + proxy
Station	IP- 169.254.90.81 + proxy

Com este ambiente posso testar meus projetos feitos no netBeans, principalmente projetos para WEB, onde utilizo o Tomcat 5.5.7 do netBeans e o Tomcat 5.0.28 para teste de desempenho, e deploy de minhas aplicações .

Tenho uma pasta d:\jsdk onde instalo o Java SDK e suas atualizações posteriores, ou seja nunca altero está pasta pois assim não preciso alterar o meu path.

A minha versão do netBeans é sem instalador (netBeans.zip), pois assim isso posso ter a versão 4.0 e a 4.1 sem problemas.

## Configurando o seu ambiente de trabalho

Para podermos realizar este tutorial preciso instalar:

JSDK – Kit de Desenvolvimento de Software (SDK) **java 2 platformStandard Edition (J2SE)**, o **netBeans** que é necessário para o desenvolvimento de nossas aplicações e o **jdk-1\_5\_0-doc.zip**  
<http://java.sun.com/j2se/1.5.0/download.jsp>

### netBeans JSDK 1.50

O ideal é que vc. baixe a versão “Bundle”, pois ela já vem com o JSDK 1.50 + netBeans 4.1.  
 Baixe também os arquivos .doc. , pois eles serão de muita utilidade

**Tomcat** – É o container que usaremos para testar as nossa aplicações WEB. É importante que seja utilizada a versão 5.0.28 ou superior devido ao suporte ao Servlets 2.4 e JSP 2.0.  
<http://jakarta.apache.org/site/binindex.cgi>

### Tomcat

Veja o tutorial do mário d’Avila “Tomcat – Instalação e Configuração Básica”, no site : <http://www.mhavila.com.Br>  
 Revista javamagazine edição nº 06 e 18.  
 Sessão de tutoriais do <http://guj.com.br>

## Instalando o seu ambiente

- 1- Após a instalação do JSDK, seja ele “blunded ” ou instalação típica.
- 2- Configure o Path
  - a: No Windows XP
    1. Acesse o “meu computador” → propriedades → “avançado”

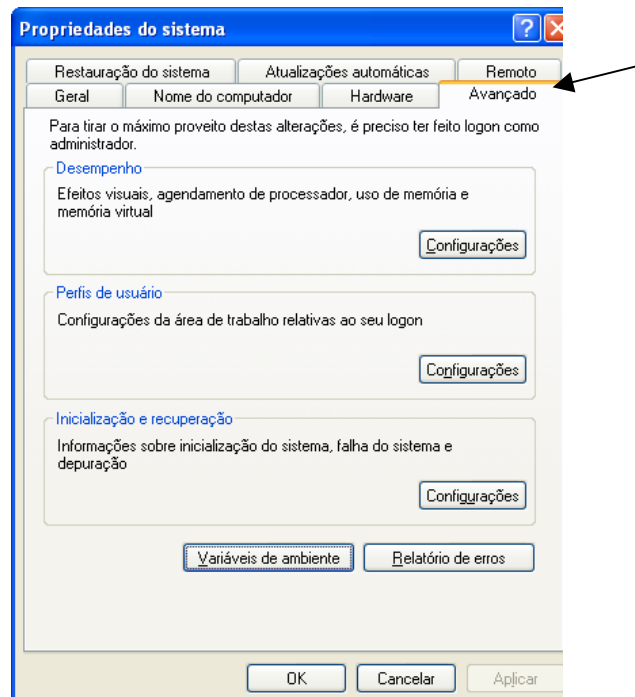


Figura 5

2. Pressione o botão "variáveis de ambiente" → "variáveis de sistema", crie uma variável com o nome JAVA\_HOME

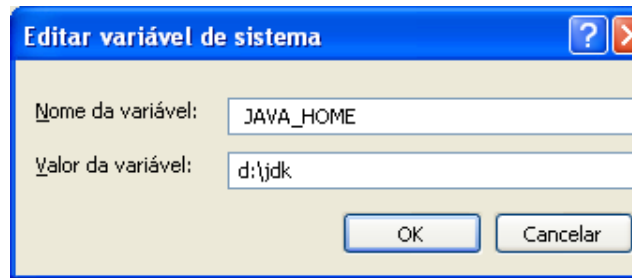


Figura 6

**Exemplo.: c:\jdk**

3. Edite a variável path e acrescente ao final do texto .: c:\jdk\bin;

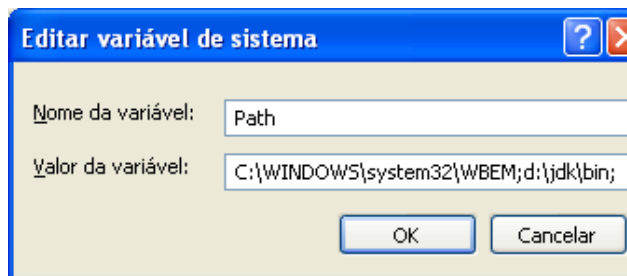
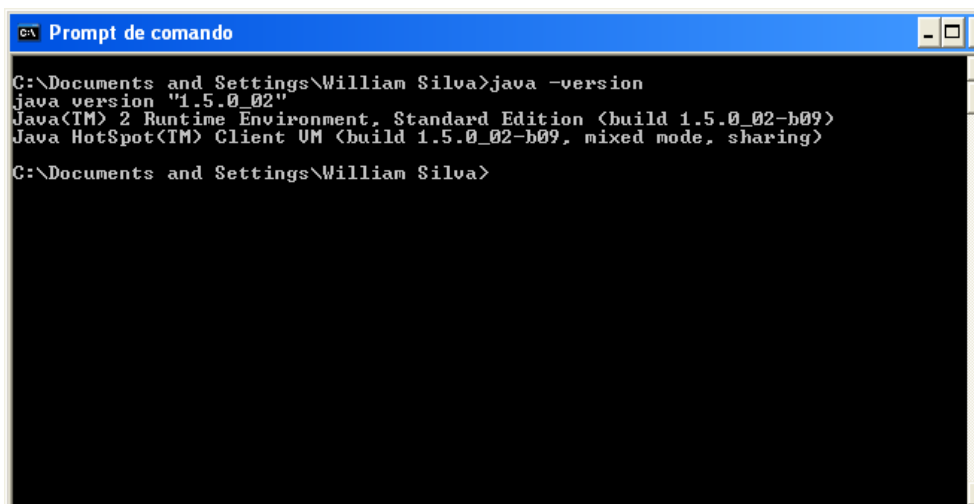


Figura 7

4. No prompt do sistema operacional digite : " Java -version "



```

C:\Documents and Settings\William Silva>java -version
java version "1.5.0_02"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_02-b09)
Java HotSpot(TM) Client VM (build 1.5.0_02-b09, mixed mode, sharing)
C:\Documents and Settings\William Silva>

```

Figura 8

3- Se quiser faça a instalação do Tomcat para seus testes, apesar de não ser um requisito imediato .

## Configurando o seu IDE netBeans

O netBeans, ao ser instalado já pode ser utilizado para criar suas aplicações, mais gosto de fazer algumas configurações básicas tais como:

### 1 - mudar o look-and-feel:

Na pasta de instalação do netBeans "`<netBeans>\netbeans\etc \ netbeans.conf`", edito o arquivo "`netbeans.conf`" e acrescento a seguinte linha "`-laf javax.swing.plaf.metal.MetalLookAndFeel`" ficando assim a minha configuração.:

```
# ${HOME} will be replaced by JVM user.home system property
netbeans_default_userdir="${HOME}/.netbeans/4.1"
```

```
# options used by netbeans launcher by default, can be overridden by explicit
# command line switches
netbeans_default_options="-J-Xms32m -J-Xmx128m -J-XX:PermSize=32m -J-XX:MaxPermSize=96m -J-Xverify:none -laf javax.swing.plaf.metal.MetalLookAndFeel "
```

```
# default location of J2SE JDK, can be overridden by using --jdkhome <dir> switch
#netbeans_jdkhome="/path/to/jdk"
```

```
# clusters' paths separated by path.separator (semicolon on Windows, colon on Unices)
#netbeans_extraclusters="/absolute/path/to/cluster1:/absolute/path/to/cluster2"
```

### 2- Escolher um navegador diferente do Internet Explorer :

Com o netBeans, iniciado vou ao menu **Tools** → **setup wizard** → **Firefox** → **Finish**.

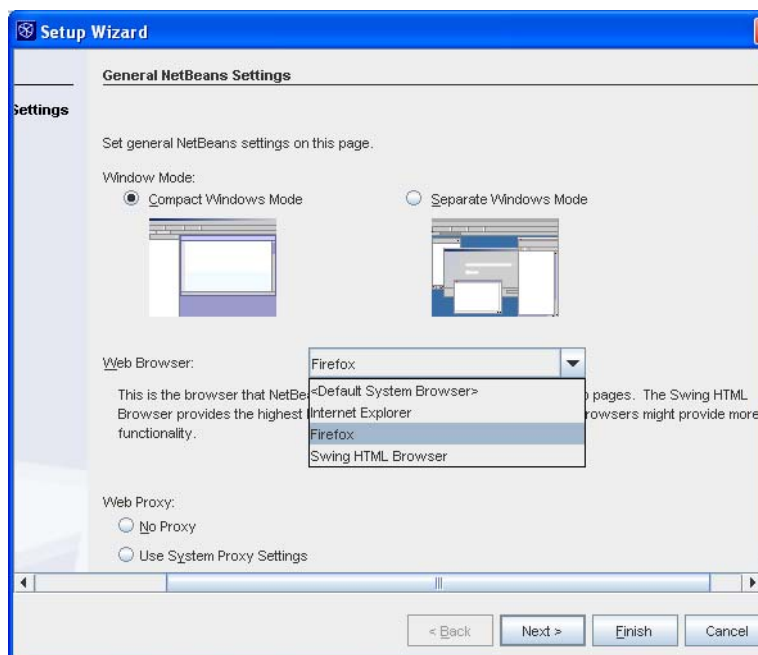


Figura 9

### 3- Configurar o javadoc para acessar recursos da API :

Com o netBeans, iniciado no menu Tools → Java Platform Manager .

Em seguida no menu Add Zip/Folder , selecione na sua pasta do Java\_home\docs\jdk-1\_5\_0-doc.zip

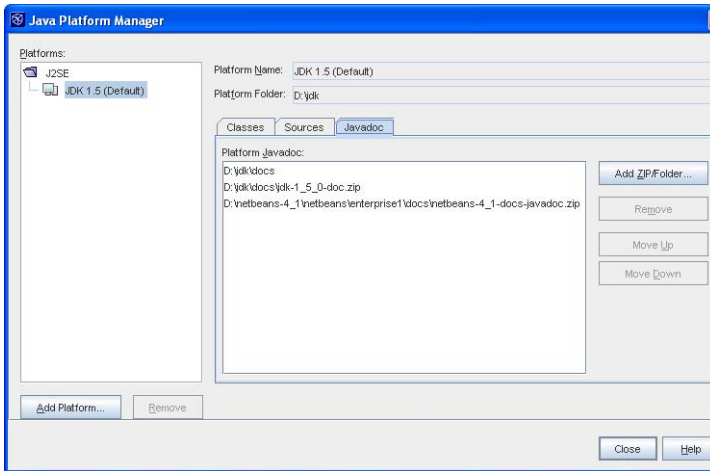


Figura 10

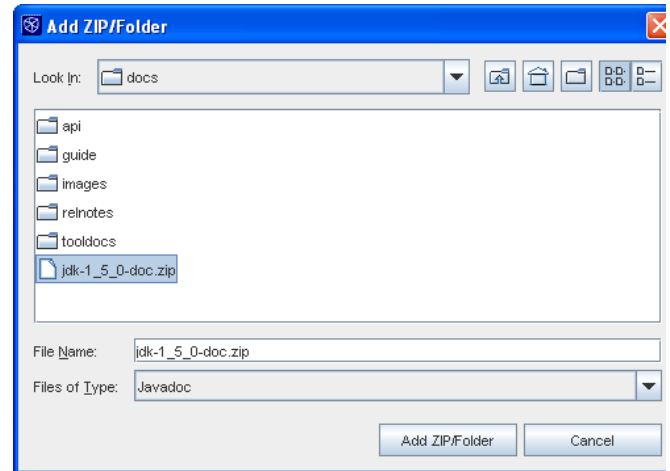


Figura 11

### 4- Alterar o tamanho da fonte de seu editor de código :

No menu Tools → options → editing → editor settings → javaeditor → font size .

**PS. Posso também selecionar todos os editores e mudar a fonte de uma só vez.**

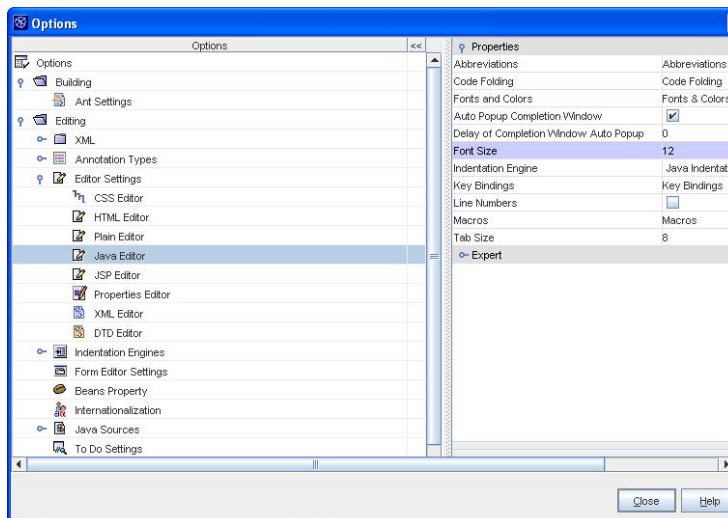


Figura 12

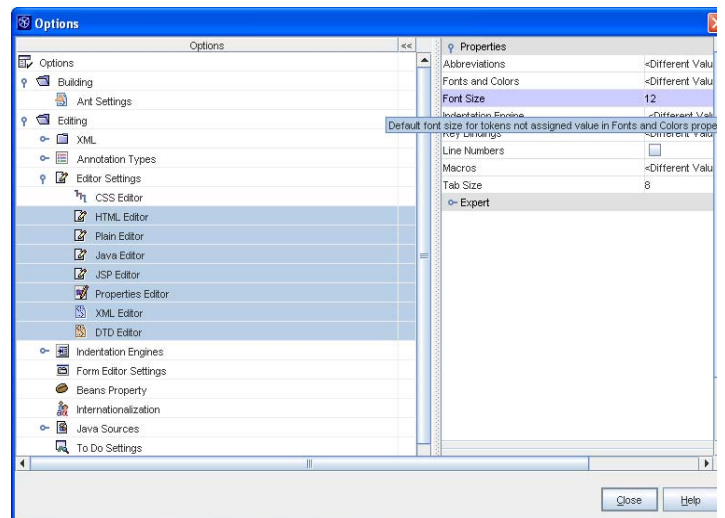


Figura 13

### 5--Alterar o JDK

Clique com o botão direito do mouse no seu projeto → Properties → Sources → Source Level .

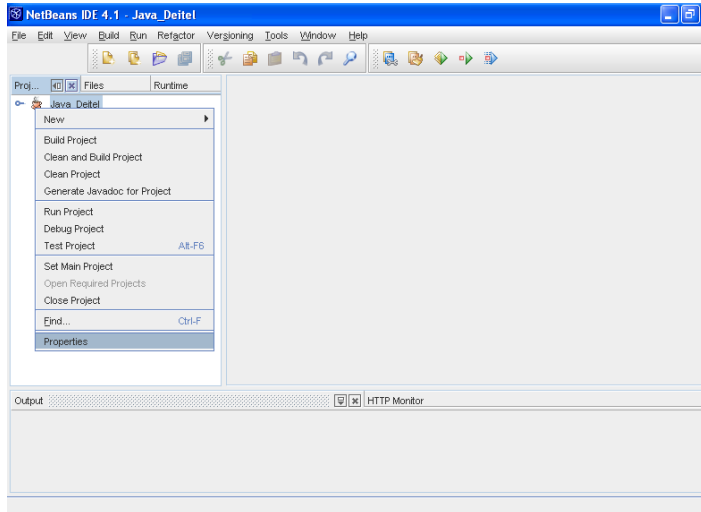


Figura 14

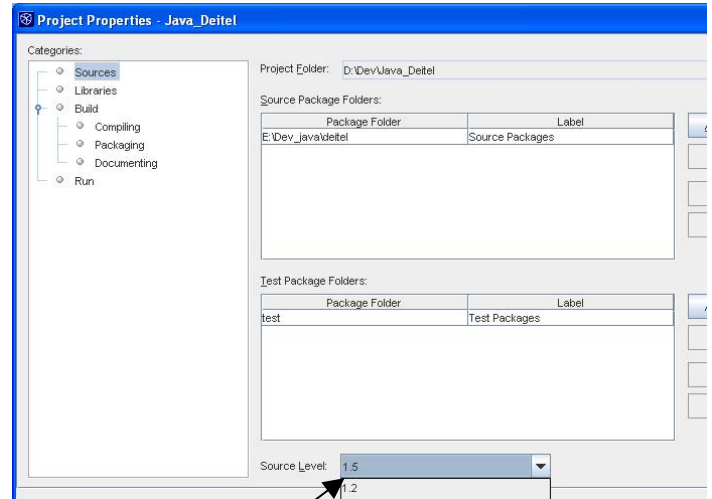


Figura 15

### 6- Instalar novas bibliotecas no seu class path de projeto:

No menu Tools → Library Manager → New Library , de um nome para suas bibliotecas.

**Normalmente instalo minhas bibliotecas na pasta do próprio netBeans = <netBeans>\JSF\ meus \*.jars . ou <netBeans>\hibernate\ meus \*.jars etc..., É somente uma sugestão.**

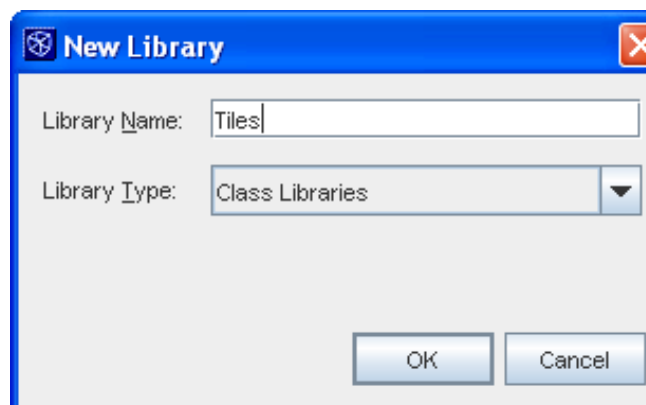


Figura 3



Após criar minha pasta de bibliotecas seleciono o botão → Add Jar/Folder . E seleciono a pasta onde eu salvei os meus \*. Jars

**Obs. Posso selecionar todos os \*. jars de uma só vez. (Figura 18)**

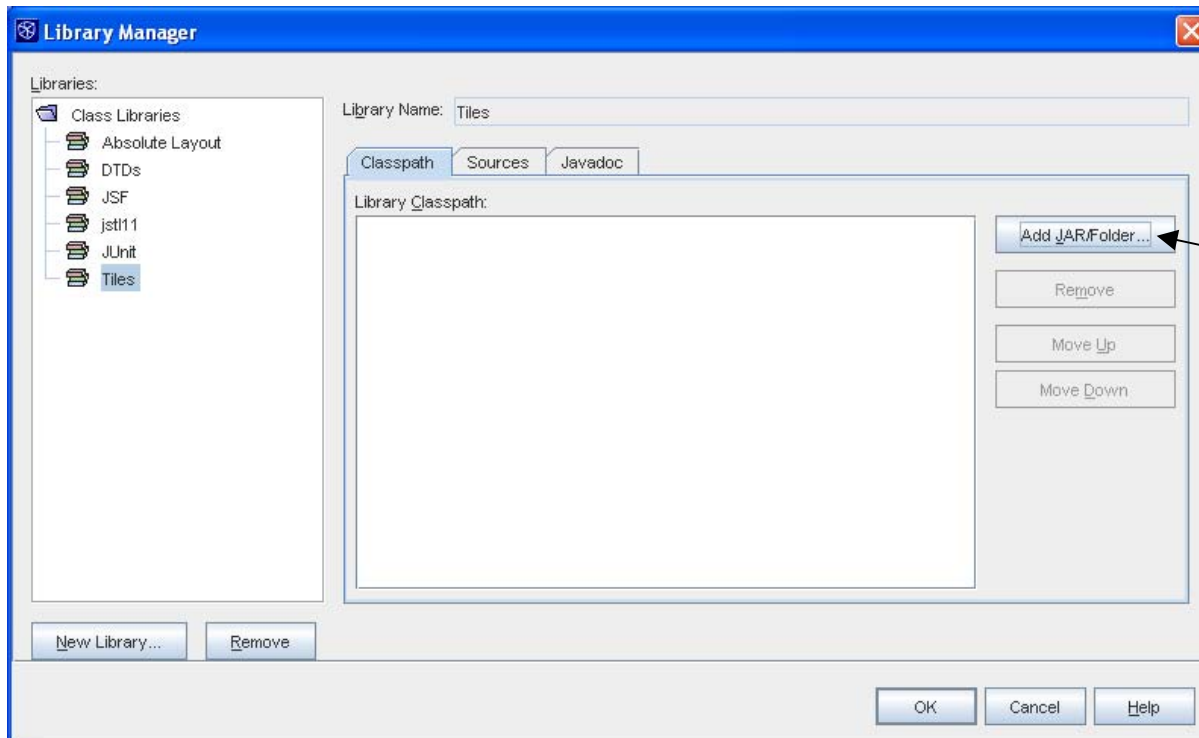


Figura 17

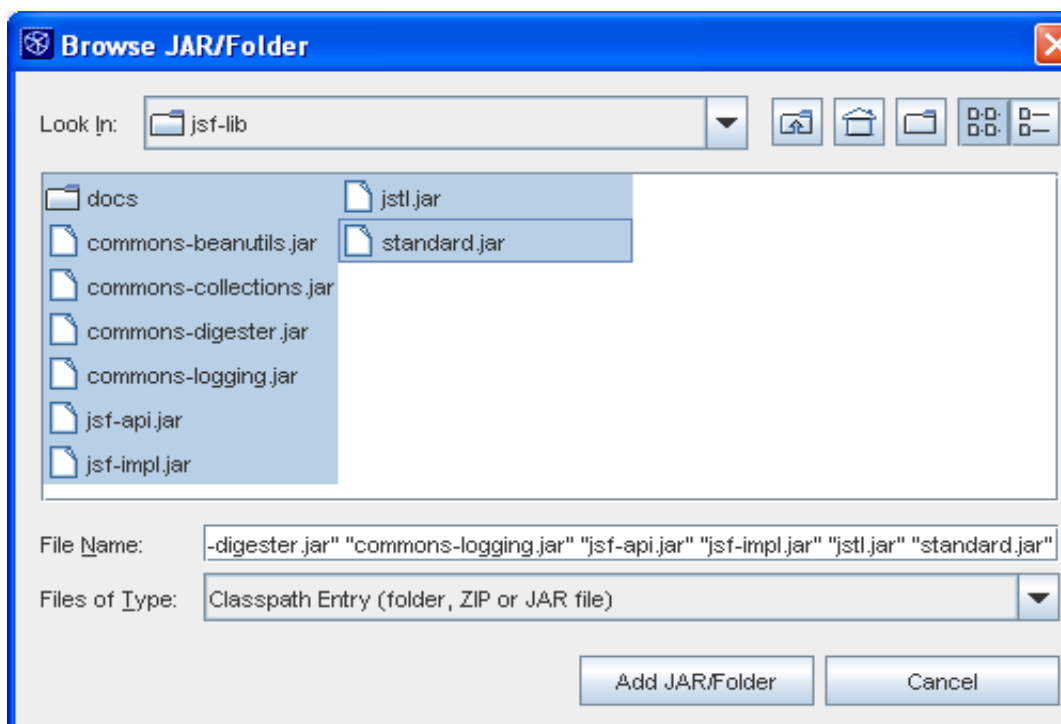


Figura 18

## 7- Instalando novos módulos

Existe duas maneiras de instalar novos módulos na sua IDE:

- Estando conectado na internet : →Tools → Update Center → Next , Será feita uma conexão com o servidor e em seguida aparecerá uma tela com vários módulos é só escolher e instalar.

Poderá também ser feita uma conexão via proxy, mais para isso deve ser feita uma pequena configuração.

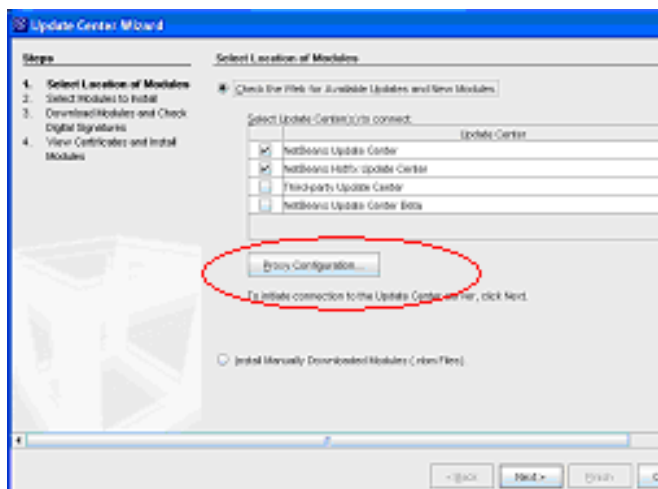


Figura 19

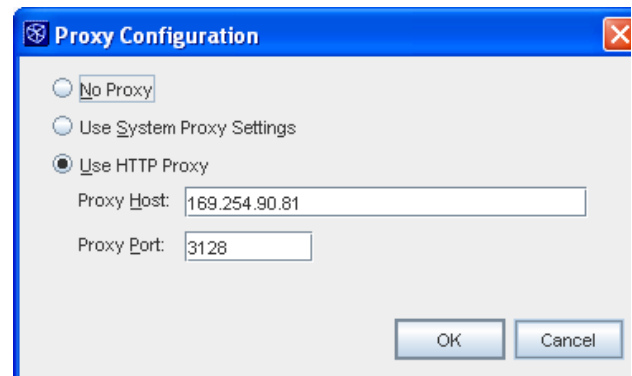


Figura 20

- Há casos em que os módulos são de outros fornecedores assim após efetuarmos download do mesmo em nosso computador devemos executar os seguintes passos:

→Tools → Update Center → selecionar o item << Install Manually downloaded Modules (.nbm Files)  
 →Next → Add , localizar a pasta onde você colocou o arquivo \*.nbm referente ao seu módulo escolhido e OK. (Figura 22)

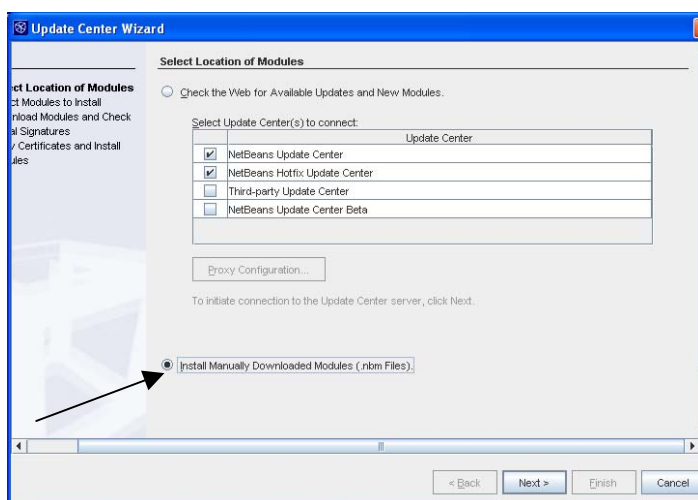


Figura 21

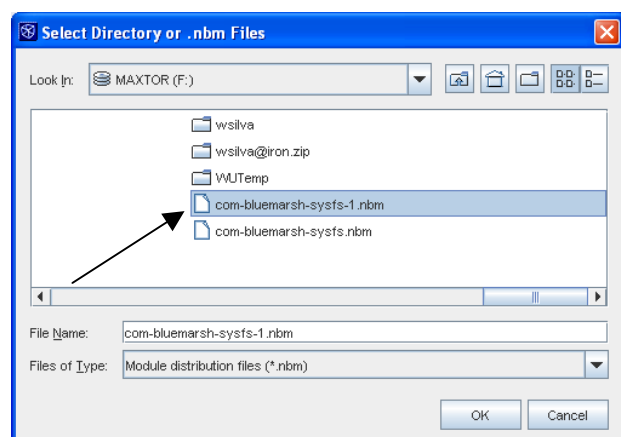


Figura 22

## Primeiro projeto

Ao ser iniciado o netBeans apresenta a seguinte tela .

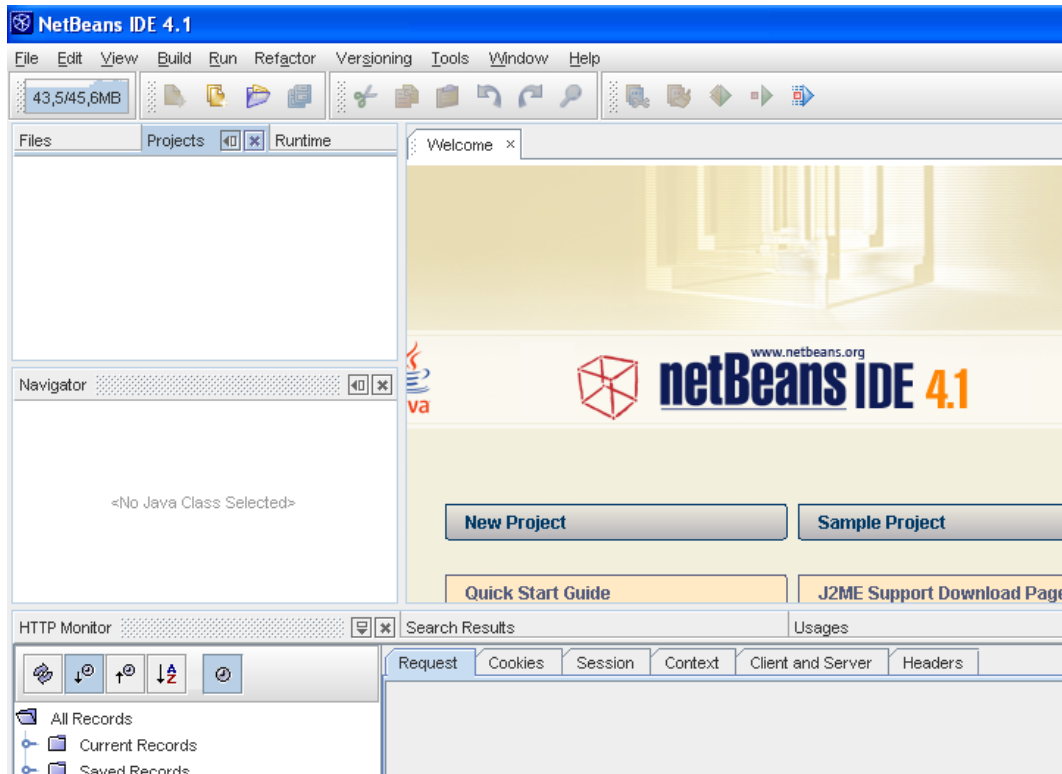


Figura 23

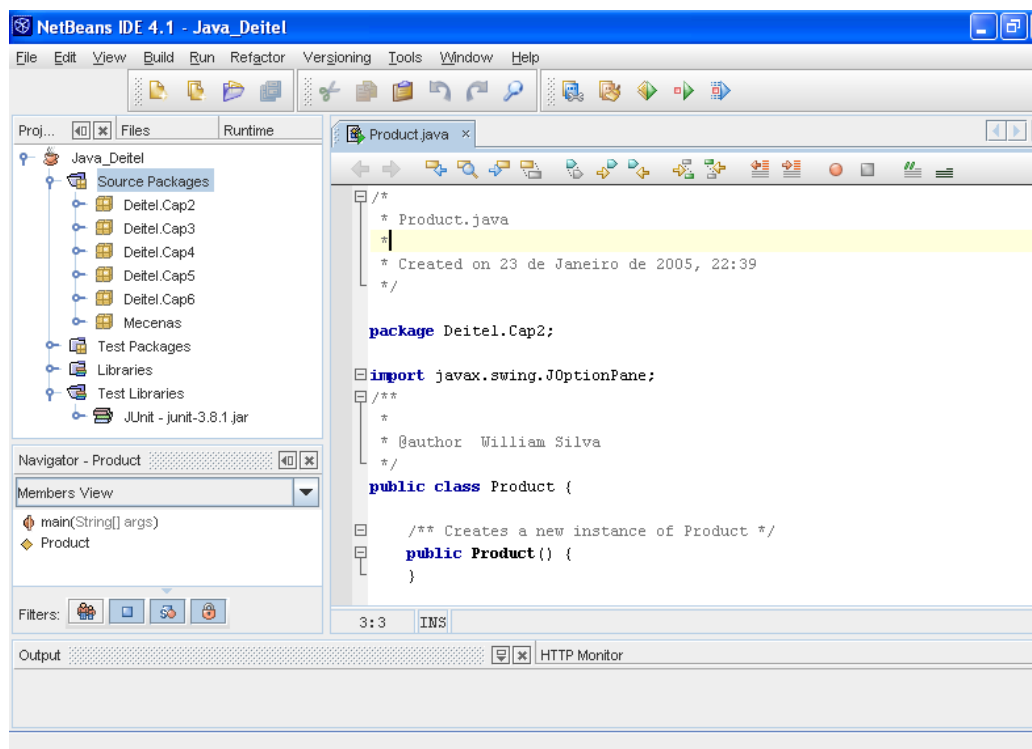


Figura 24

1- No menu file → new project.

Será apresentada a tela de new project onde podemos definir qual template e projeto usaremos. (Figura 25)

- General
- WEB
- Enterprise
- Samples
- 

o nosso foco inicial será no template general → Java application, pois com ele poderemos criar a nossa classe Java que automaticamente criará o método main class.

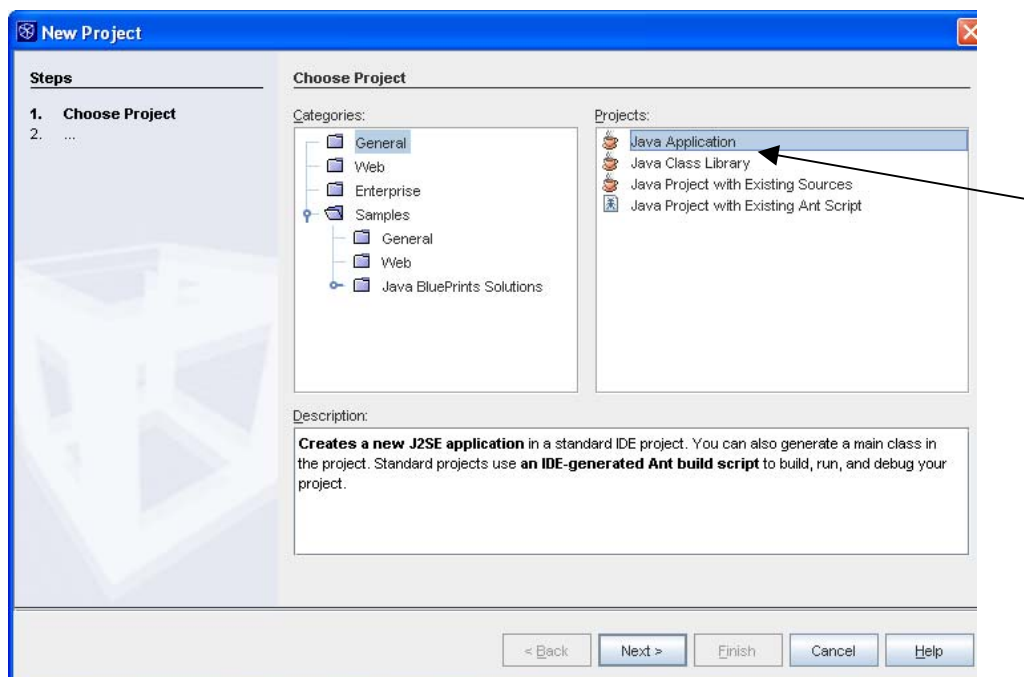


Figura 25

1- Escolhemos o tipo de aplicação;

1.2 - Nome e localização de nosso projeto;

1.3 – Finish;

Podemos observar que foi criado um projeto com o nome de HelloWorld. Com uma estrutura de package "com.mydomain.olaGuj", será feito um " **scanning project classpaths**" para habilitar o auto completar e algumas features.

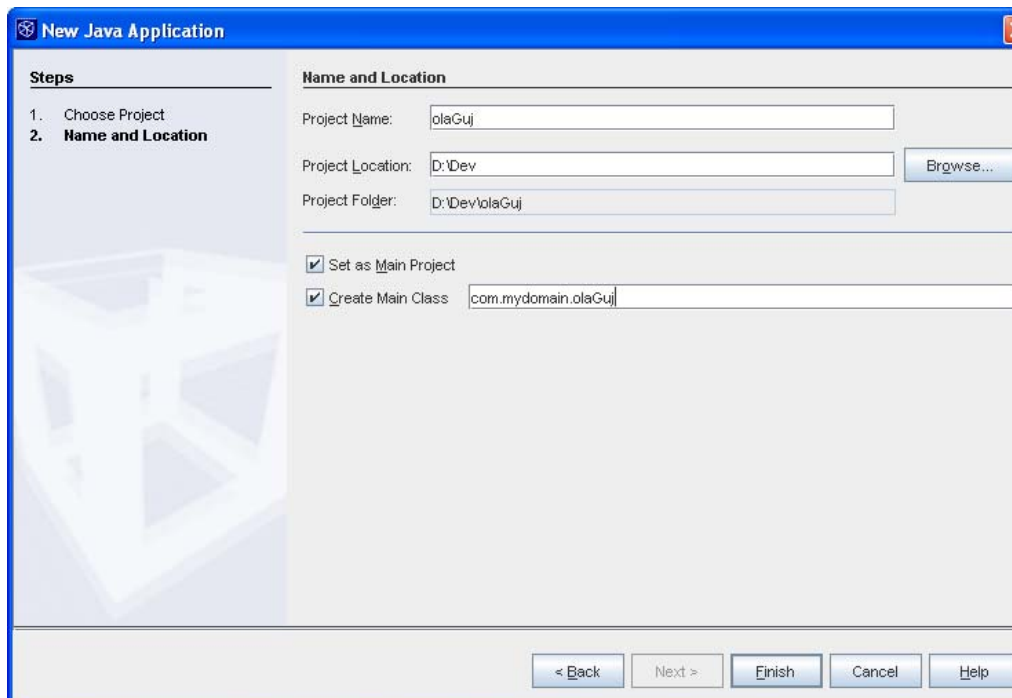


Figura 26

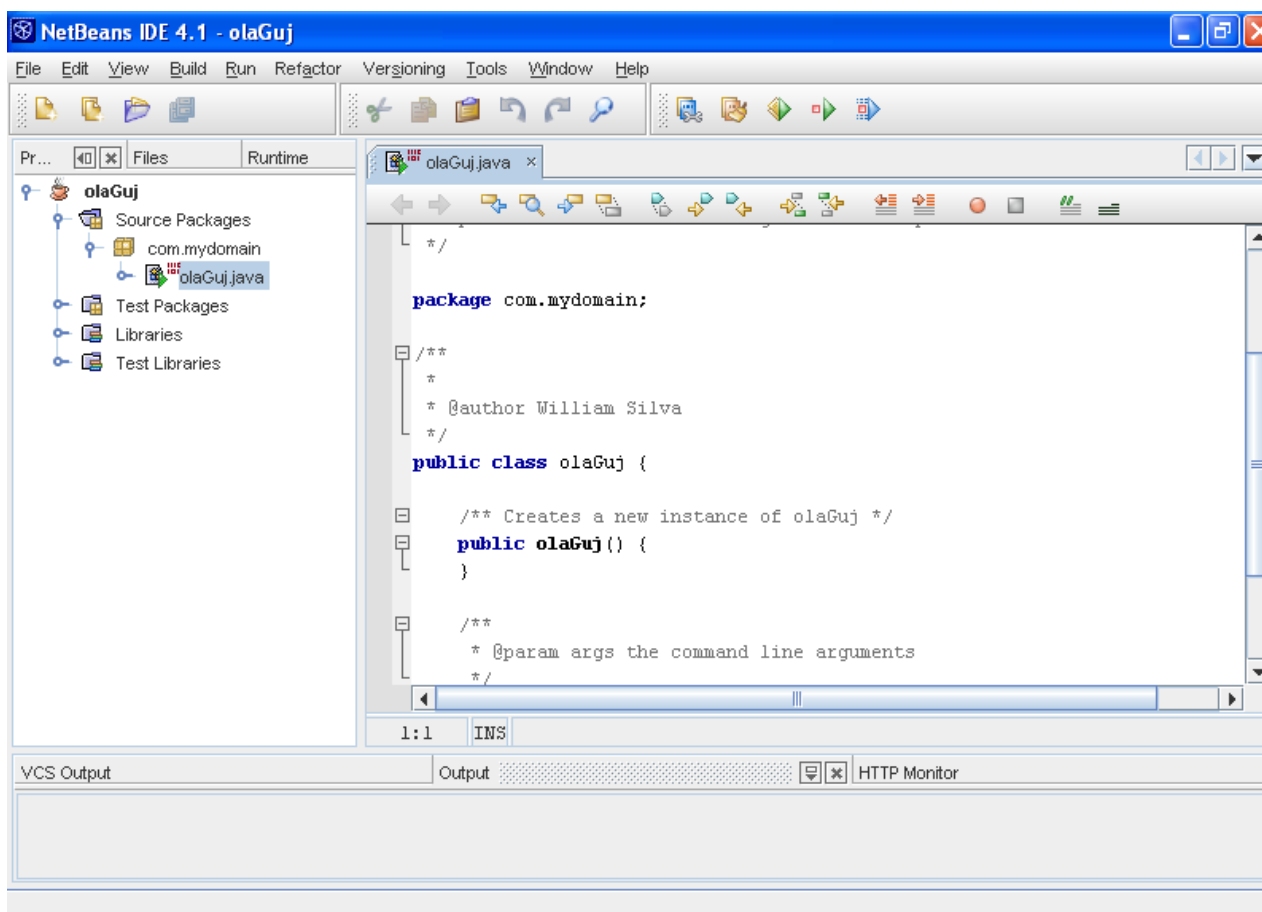


Figura 27

Se observarmos a área de projeto, com seus nodes expandidos, veremos toda a estrutura de um pacote Java. (Figura 28)

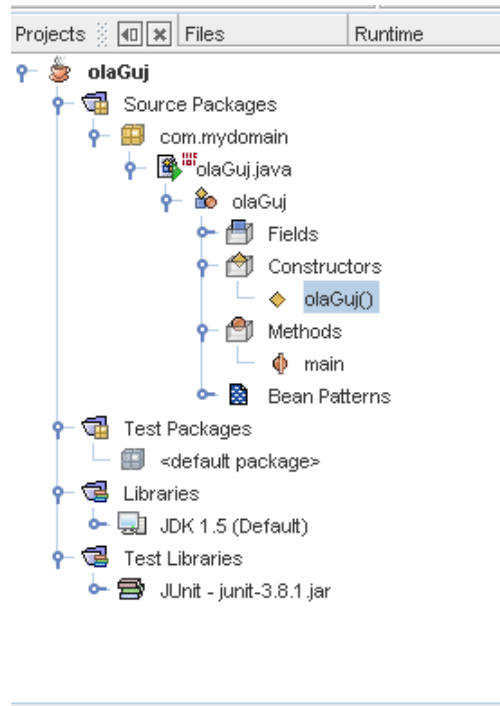


Figura 28

A área de codificação já deixa espaço para que o nosso código seja complementado:

“TODO code application logic here ” e digitamos “ **System.out.println (“Olá netBeans IDE”);** ”

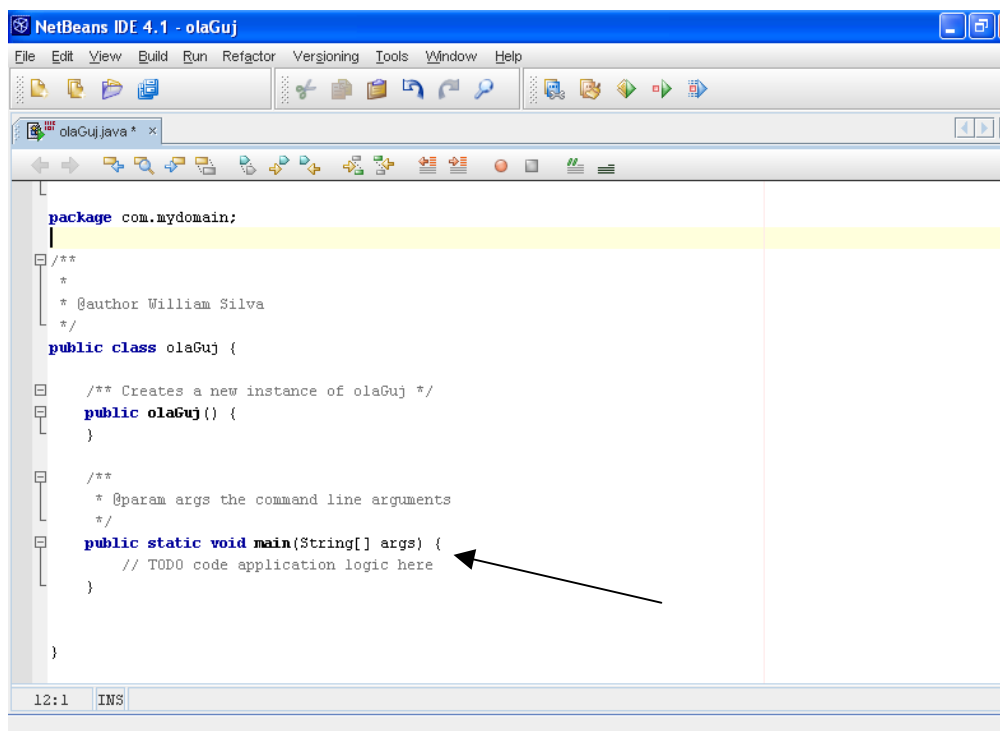


Figura 29

Conforme formos digitando o nosso código podemos fazer uso do auto completar utilizando as teclas :

*Ctrl + space* ou do **"Java Editor Abbreviations"** = `sout + space` .

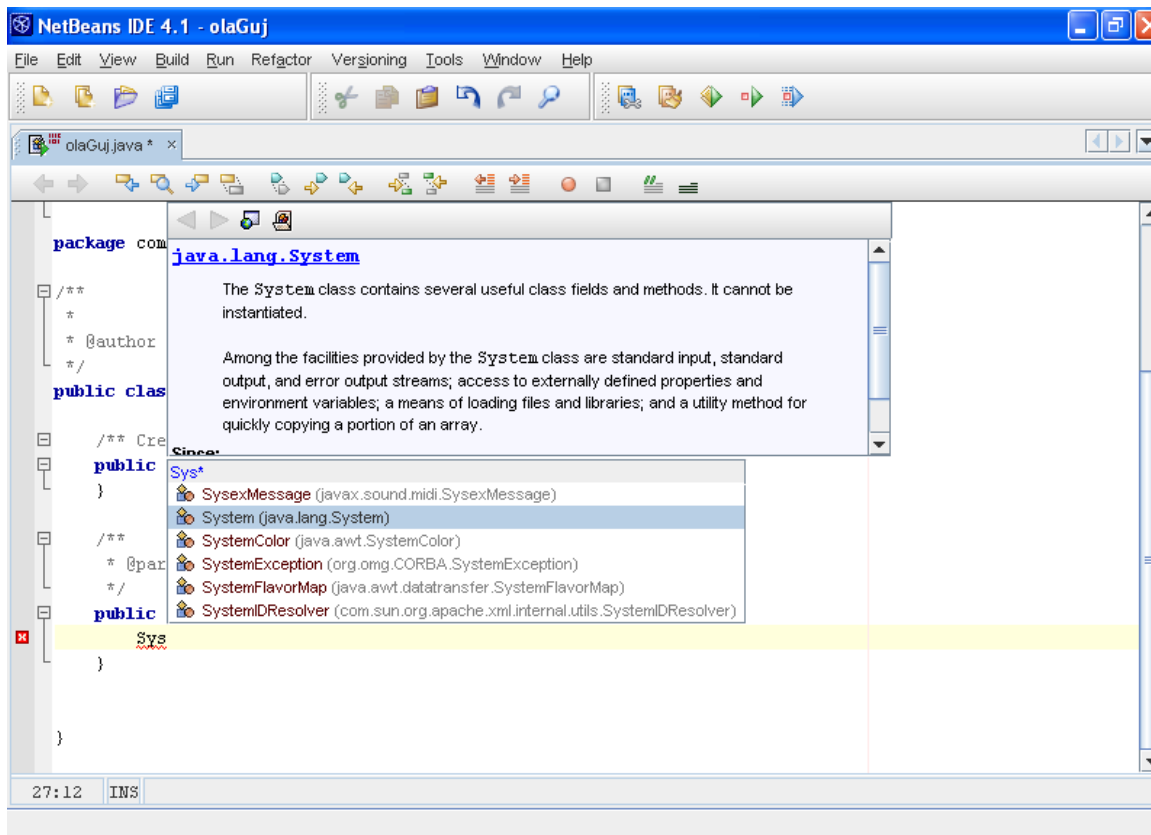


Figura 30

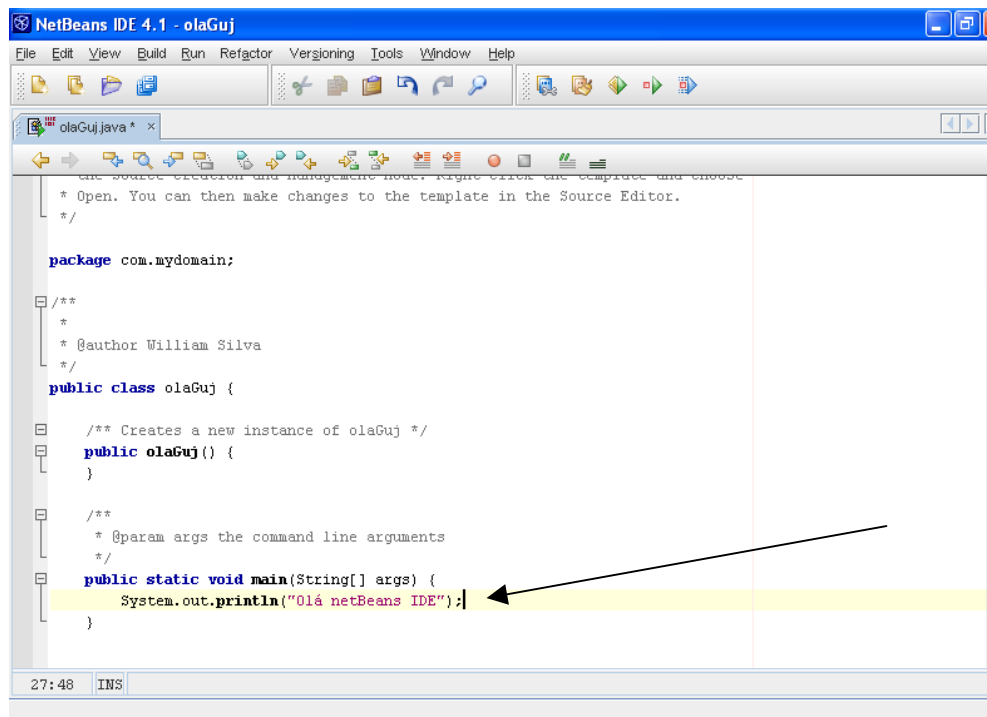
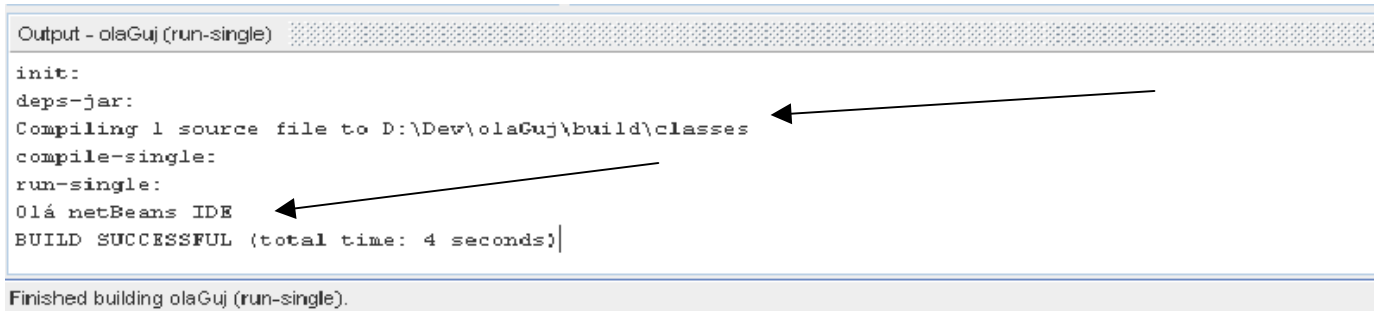


Figura 31

Salvamos nosso projeto com Ctrl+S;

Executando nosso projeto Shift+F6 ou F6;

Obs. Podemos observar na Output window o script do Ant e a mensagem " Olá netBeans IDE ".(Figura 32)



```
Output - olaGuj (run-single)
init:
deps-jar:
Compiling 1 source file to D:\Dev\olaGuj\build\classes
compile-single:
run-single:
Olá netBeans IDE
BUILD SUCCESSFUL (total time: 4 seconds)

Finished building olaGuj (run-single).
```

Figura 32

## Criando seu projeto

Clicando com o botão do mouse sobre seu package " **com.mydomain** " → New → Java Main Class e para nome de projeto " Welcome4 ". Apague todo o código pré-criado e digite o código abaixo, e analise o comportamento do netBeans IDE, percorra os menus, a árvore de projetos, clique com o botão do lado direito do seu mouse nos nós do seu projeto analise o menu drop-down apresentado, enfim explore a sua IDE . Para compilar pressione F6.

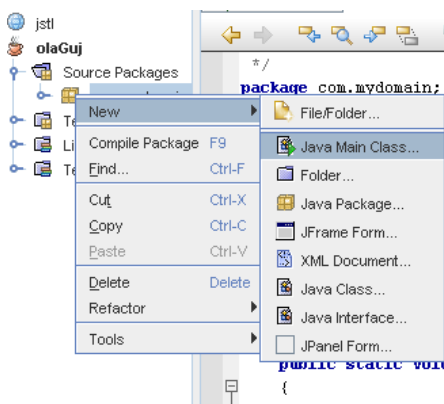


Figura 33

```

Welcome4.java
package com.mydomain;

import javax.swing.JOptionPane;
public class Welcome4 {

    public Welcome4() {
    }

    public static void main(String[] args)
    {
        JOptionPane.showMessageDialog(
            null, "Bem vindo\nao\nnetBeans\nIDE!!!");
        System.exit (0);
    }
}

```



### Editor de códigos

Com editor de código do netBeans você ira criar e gerar seus códigos de projetos. Ele trabalha com diferentes formas e estruturas de arquivos sendo ela Java, JSP, JSTL, XML, HTML, DTDs, Properties, etc. Além de possuir um perfeito editor visual para criação de form Swing e AWT permitindo utilizar as GUI Templates como Jpanel, JFrame entre outros.



Figura 34

### Dicas

Com vários arquivos de projeto aberto, utilize a barra de navegação para escolher a sua aba de projeto. Selecione o item "import" do código `welcome4.java` e pressione Shift+F1.

No menu View → Documentation Índices, selecione um item e navegue pela documentação de APIs. (Figura 36)

Use a documentação on-line do netBeans em Help → Help\_Contents (Figura 36)

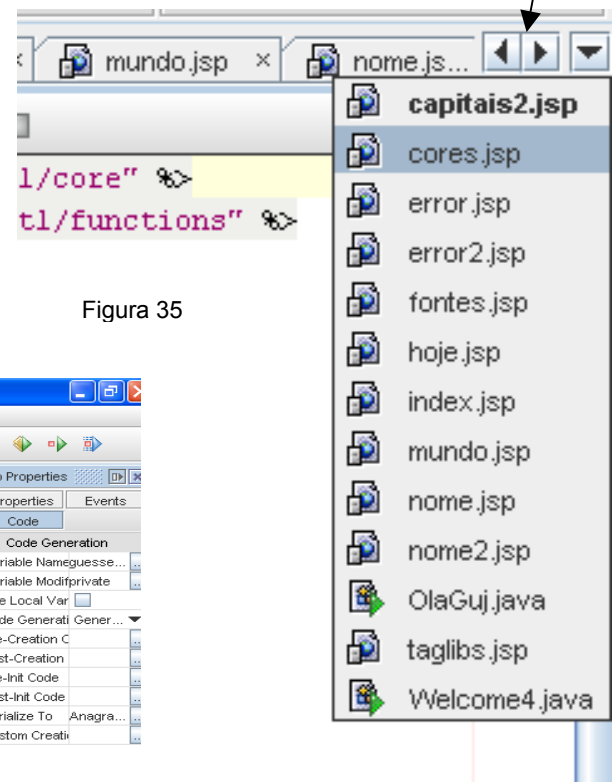


Figura 35

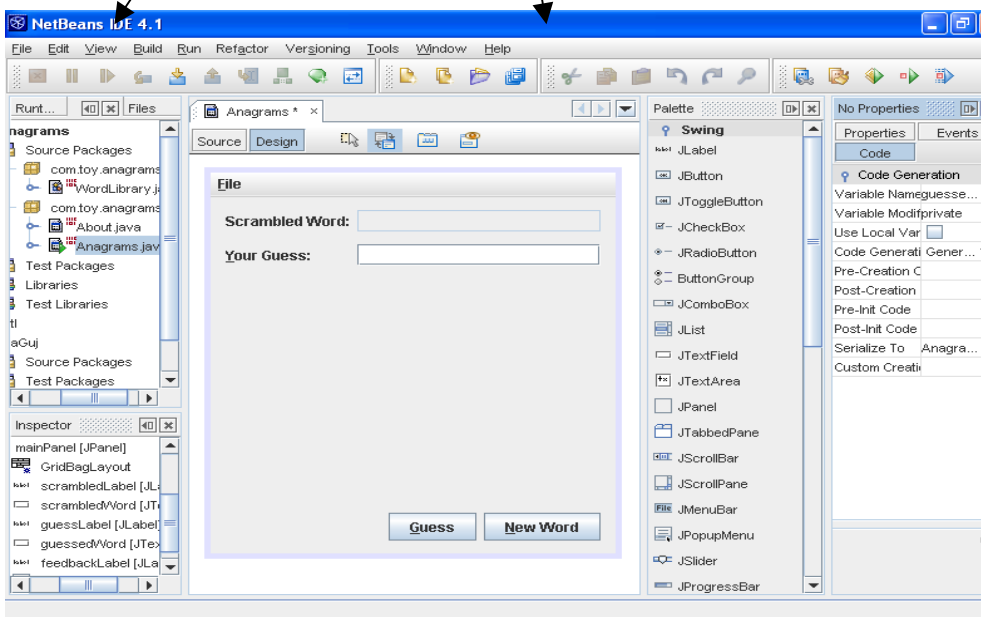


Figura 36

## Conectando bancos de dados:

A visão Runtime do netBeans IDE, é o nosso próximo foco pois nela configuramos: (Figura 38)

- Conexão com banco de dados;
- Iniciamos e paramos o Tomcat;
- Analisamos processos em execução;
- Catálogos XML
- Etc...

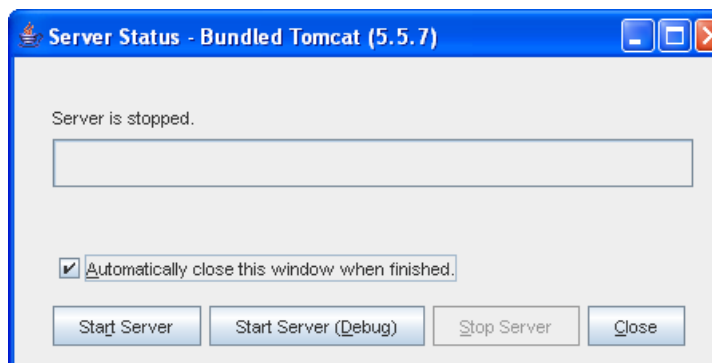


Figura 37

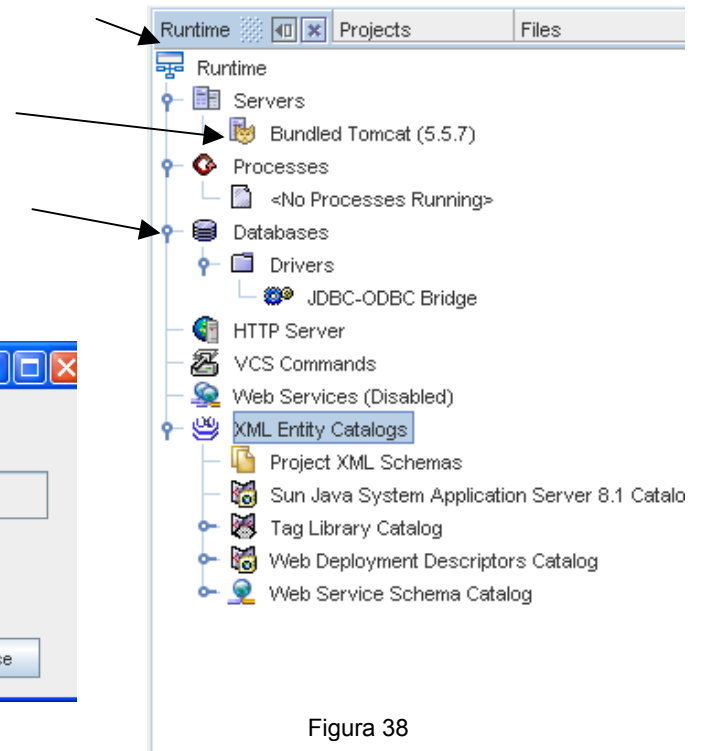


Figura 38

Antes de criarmos nossa fonte de dados, é preciso copiar as classes dos drivers JDBC (FirebirdSQL-1.5.5JDK\_1.3) para uma pasta de nossa escolha. (Figura 39)

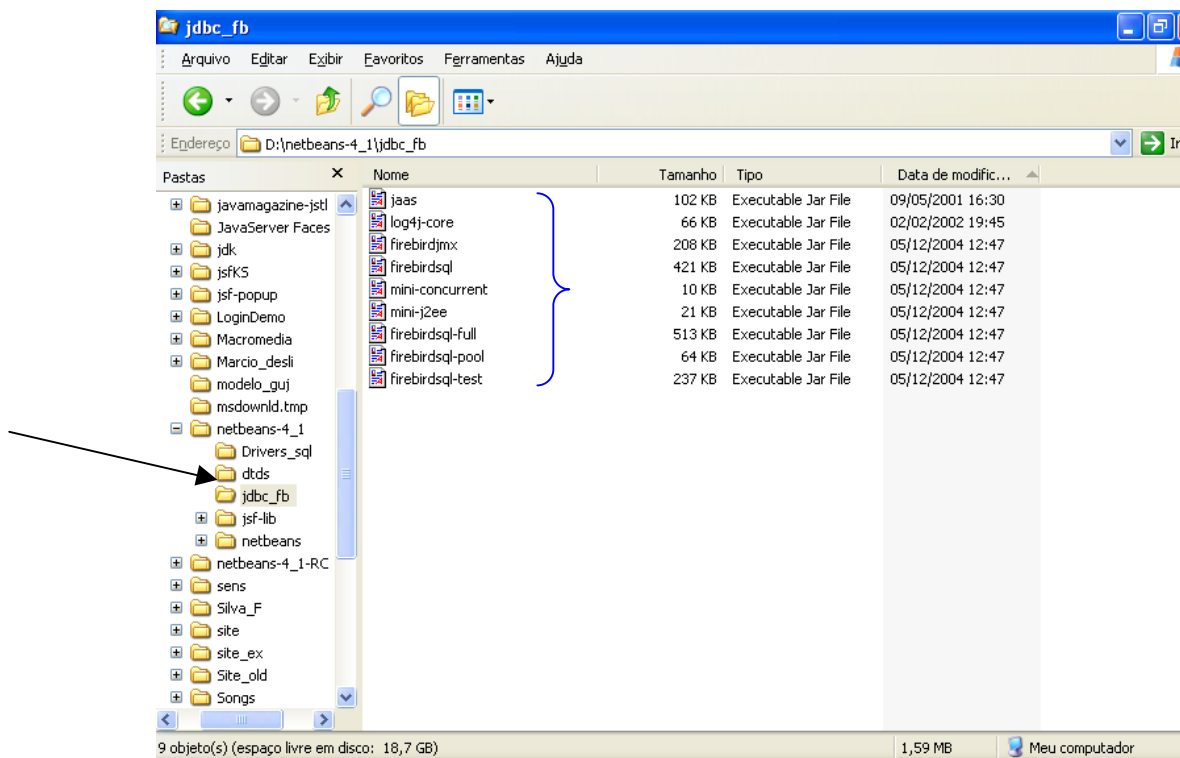


Figura 39

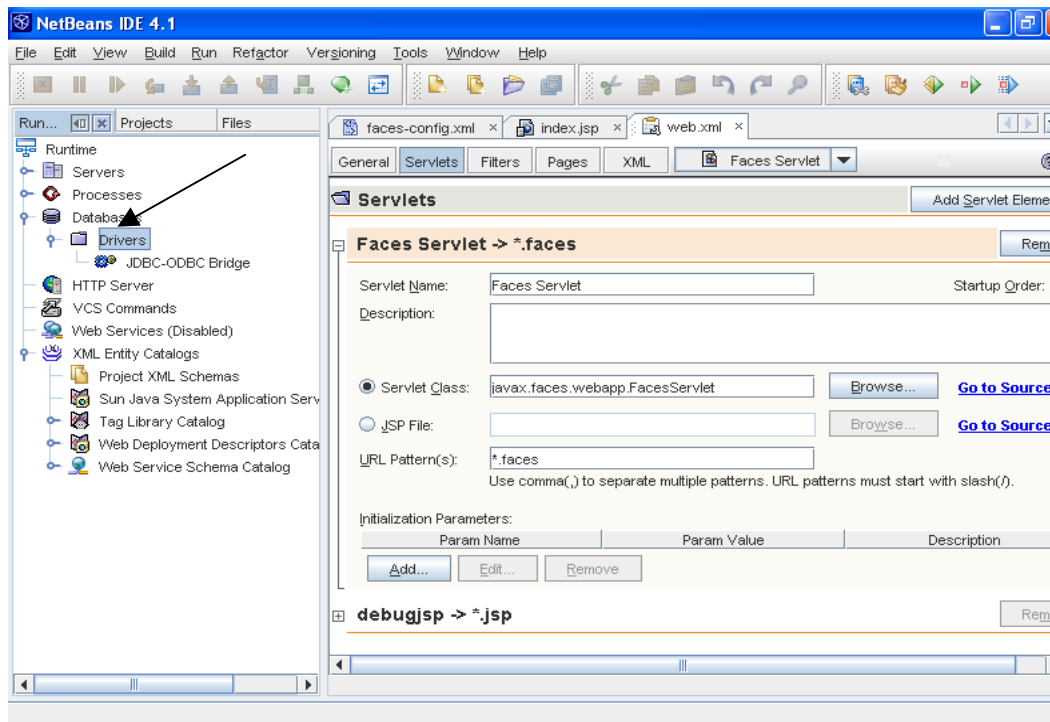


Figura 40

Com o botão direito do mouse Add → drivers (direcione para a sua pasta de drivers JDBC criada) e seleçone todas as classes do driver JDBC: (Figura 39)

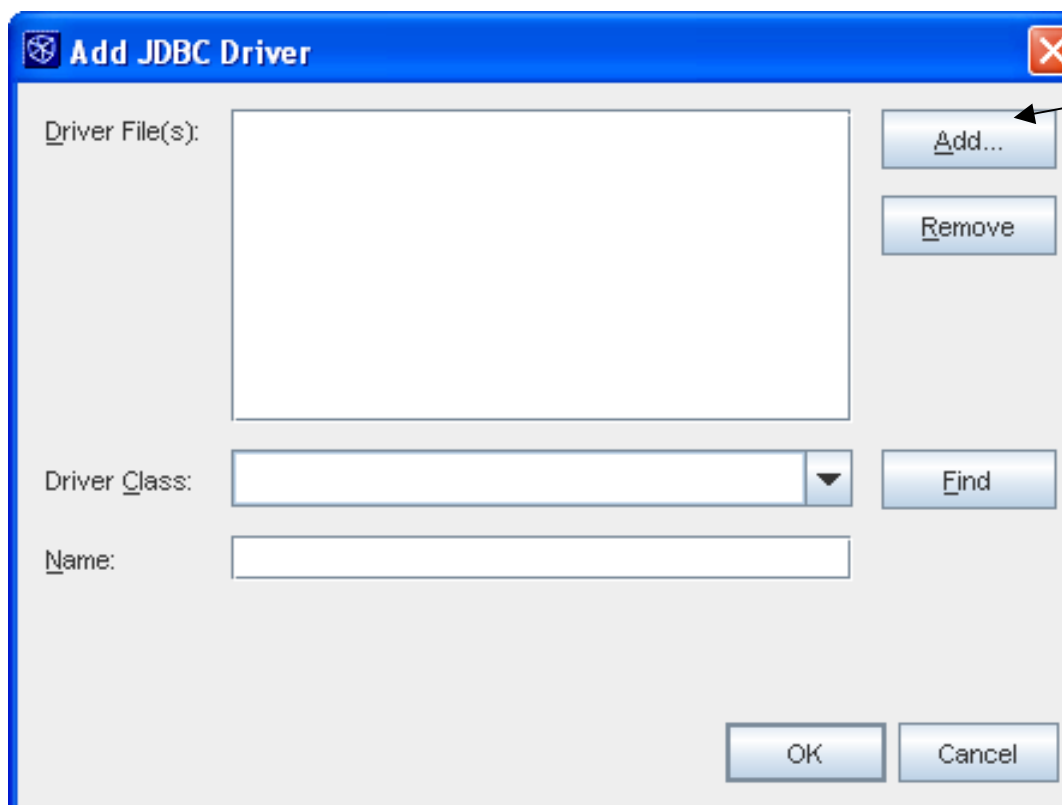


Figura 41

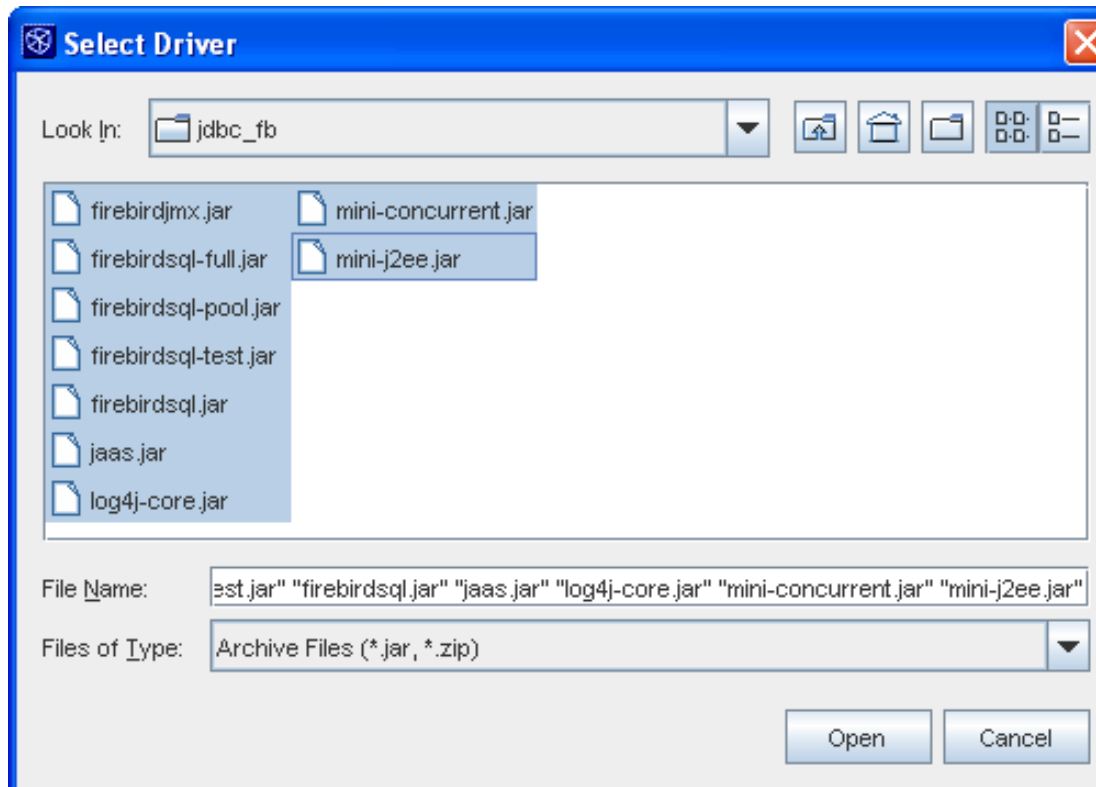


Figura 42

Agora com os drivers reconhecidos vamos selecionar com o botão direito do mouse, no driver desejado e criar a nossa conexão:

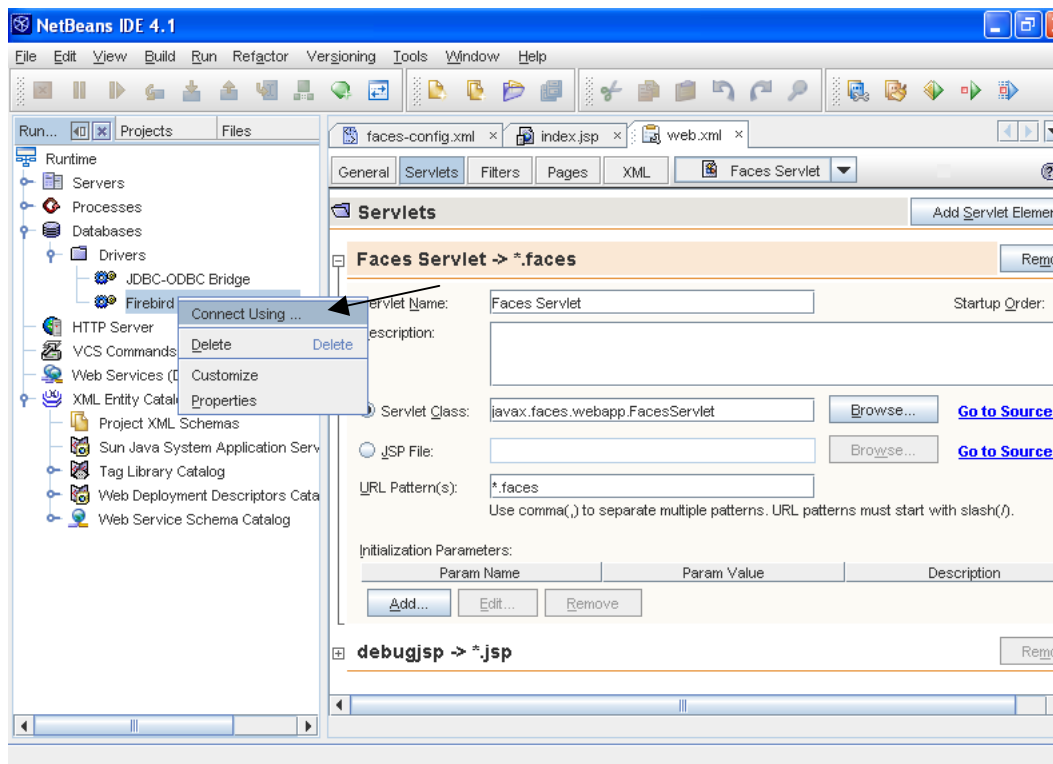


Figura 43

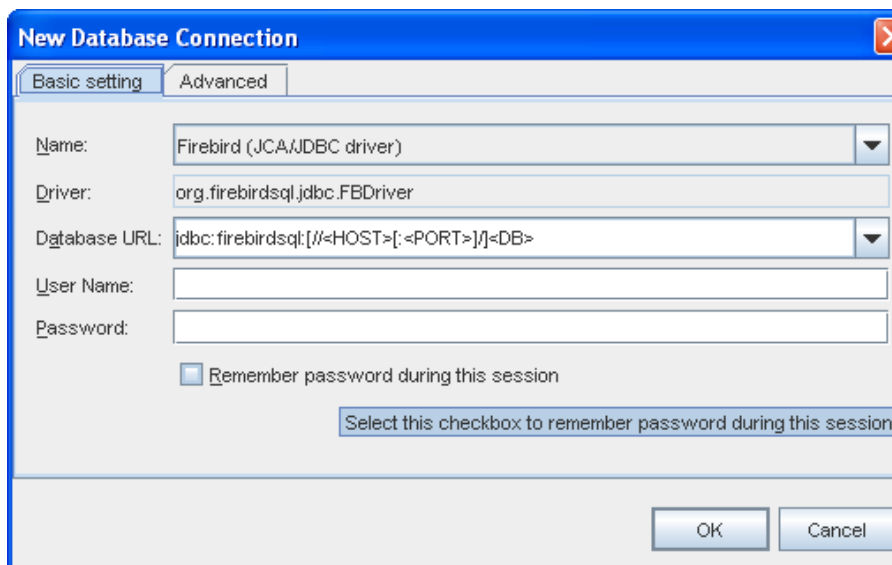


Figura 44

Em name → Selecione o nome do seu drive JDBC

Em database URL use o seguinte → jdbc:firebirdsql://localhost:3050/c:\employee.gdb

User name → SYSDBA

Password → masterkey

E OK...

**Obs..: O guardiam do firebird deve ser iniciado**



Firebird 1.5 Server  
 Manager

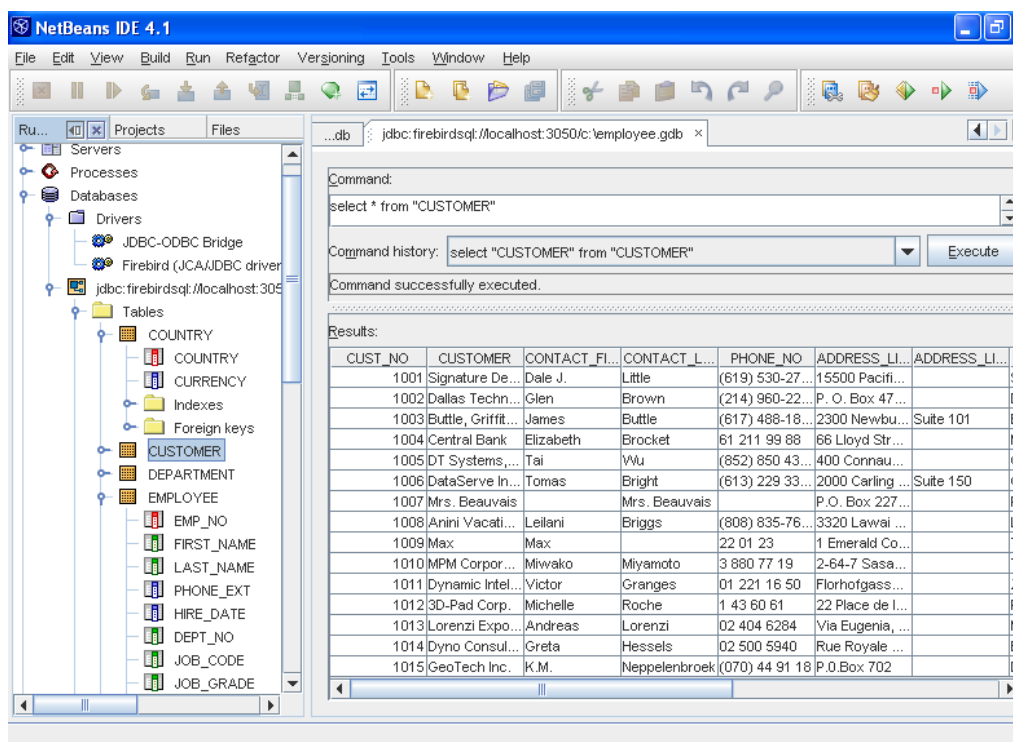


Figura 45

Podemos observar que é possível criar tabelas, executar comandos SQL, navegar pela base de dados etc.

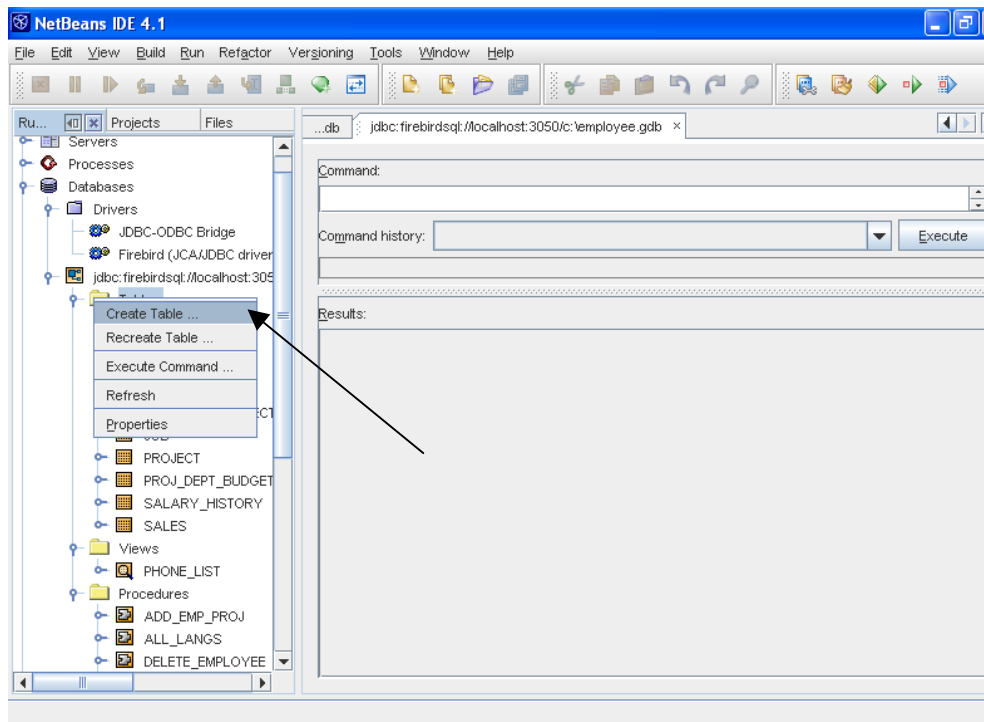


Figura 46

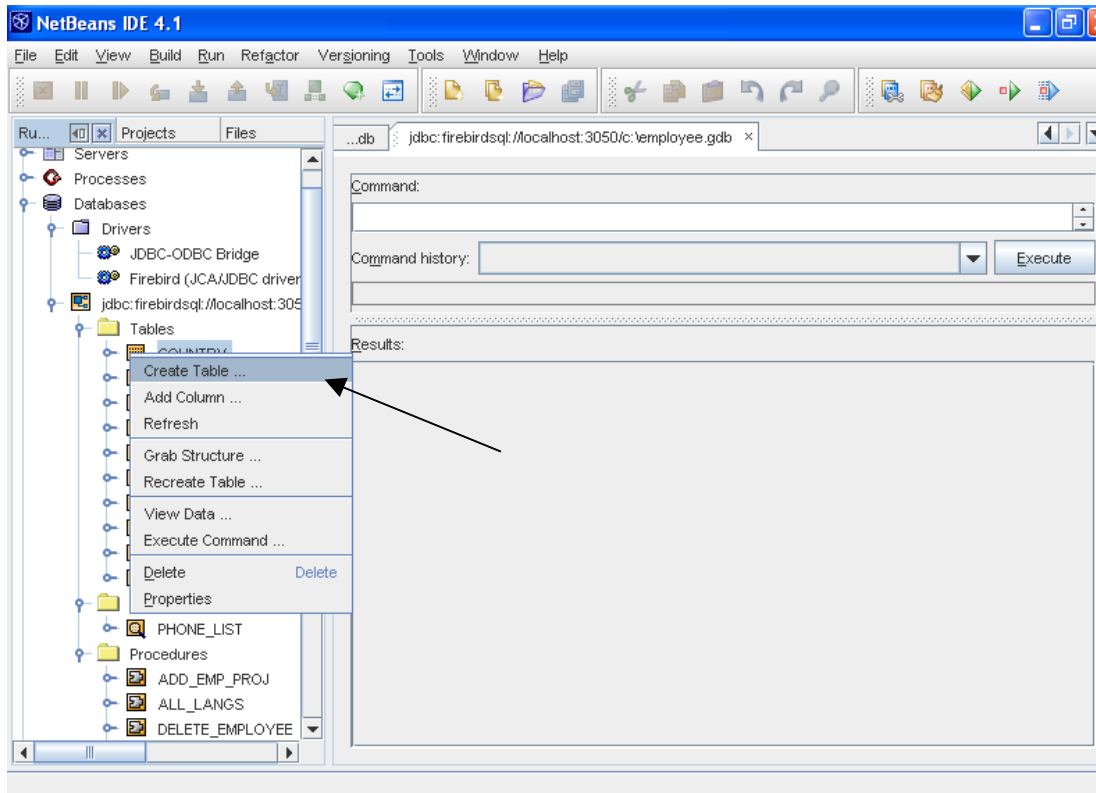


Figura 47

Um outro recurso é criar um pacote new library como já mencionado em "Configurando o seu IDE netBeans" no item 6 e, em seguida adicioná-los ao seu projeto, isso garante que quando for feito o "Build Project" eles serão automaticamente colocados no seu pacote "JAR" ou "WAR".

### Dicas

No site <http://jaybirdwiki.firebirdsql.org/download/Downloads>, você encontrará o drive Jaybird do tipo 4 para sua conexão JDBC. O Firebird não vem com um Front-End, por isso utilizo o IBExpert Personal Edition <http://ibexpert.com>

### Criando um Web project

Vamos agora criar um novo projeto, só que agora será criado um projeto para ser executado dentro de um container JSP e, como temos o Tomcat incluso no netBeans faremos uso dele e depois poderemos instalar o seu arquivo "WAR" em qualquer outro container JSP. Selecionamos o menu File → New Project → Web Application, como nome do projeto escolhemos "webBird". Repare que foi criado um projeto completo com estrutura de pastas e diretórios, porém o módulo que vai nos interessar no momento é o "Web Pages", e será nele que ficará toda a nossa estrutura de páginas JSP.

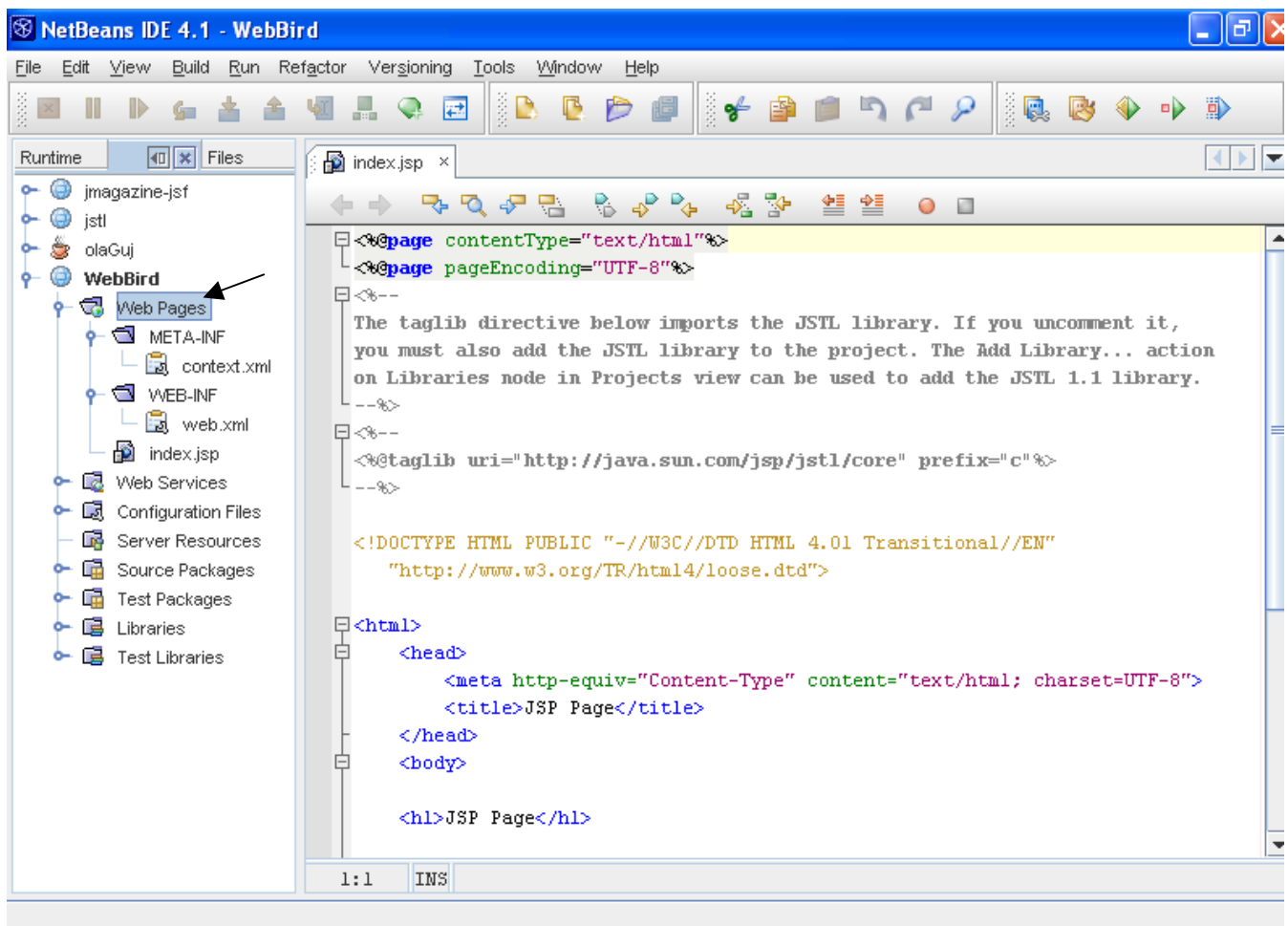


Figura 48

Com a pasta Web Pages selecionada, Ctrl+N → Categories → File Types → JSP → Next; e defina o arquivo com o nome " ServerConfiguration" , sem a terminação JSP e, digite os código abaixo. Repita o processo para os demais códigos.

ServerConfiguration.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%

String DataBaseAddr ="jdbc:firebirdsql:localhost/3050:C:/employee.gdb";
String UserName ="SYSDBA";
String Password ="masterkey";
```

Listagem.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@
include file="ServerConfiguration.jsp"
%>

<%@
page import="org.firebirdsql.jdbc.*,org.firebirdsql.logging.Logger,java.sql.*"
%>
<%
Class.forName("org.firebirdsql.jdbc.FBDriver");
Connection aConnection = DriverManager.getConnection( DataBaseAddr,
UserName, Password);

Statement stmt = aConnection.createStatement();

String SQL="SELECT * FROM EMPLOYEE ";
SQL=SQL+ "ORDER BY LAST_NAME";
ResultSet aQuery = stmt.executeQuery(SQL);
%>

<html>
<body bgcolor="#E0E0E0">
<table>
<tr><td>EMP_NO</td><td>First Name</td><td>Last Name</td></tr>
<%
while (aQuery.next())
{
%>

continua.....
```



```

<tr>
<td><%= aQuery.getString("EMP_NO") %></td>
<td><%= aQuery.getString("FIRST_NAME") %></td>
<td><%= aQuery.getString("LAST_NAME") %></td>
</tr>
<%
}
%>

</table>
<%
aQuery.close();
stmt.close();
aConnection.close();
%>
</body>
</html>

```

Pesquisa.jsp

```

<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<%

Connection con = null;
Statement stm = null;
ResultSet res = null;
%>

<html>
<body bgcolor=#E0E0E0 >
<TABLE>
<b>PESQUISA SELECT JAYBIRD :</b><br>
</TABLE>
<%
Class.forName("org.firebirdsql.jdbc.FBDriver").newInstance();
con = DriverManager.getConnection("jdbc:firebirdsql:localhost/3050:C:/employee.gdb",
"SYSDBA", "masterkey");
stm = con.createStatement();
res = stm.executeQuery("select * from employee");

continua.....

```

```
while (res.next())
{
    out.println(res.getString("FIRST_NAME")+"<br>");
}

res.close();
%>

</body>
</html>
```

Index.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<html>
<head>
<title>
    index
</title>
</head>
<body bgcolor="#ffffff">
<h1>
Exemplos Jaybird
</h1>
<br />
<b>Dica :</b> Estude JSTL...!
<br />
<br />
<a href="listagem.jsp">Lista a tabela Employee.gdb</a>
<br />
<a href="pesquisa.jsp">Exemplo de Select</a>
</form>
</body>
</html>
```

Selecione na pasta Web Pages a página index.jsp pressione a tecla F6 ou Shift+F6, observe que o Tomcat será iniciado e todo um processo de depuração estará sendo executado, podendo ser observado pela janela Output. Em seguida o seu navegador selecionado como padrão será iniciado com a página index.jsp que em uma rede intranet poderá ser acessada pelo seguinte endereço "<http://localhost:8084/WebBird/index.jsp>"

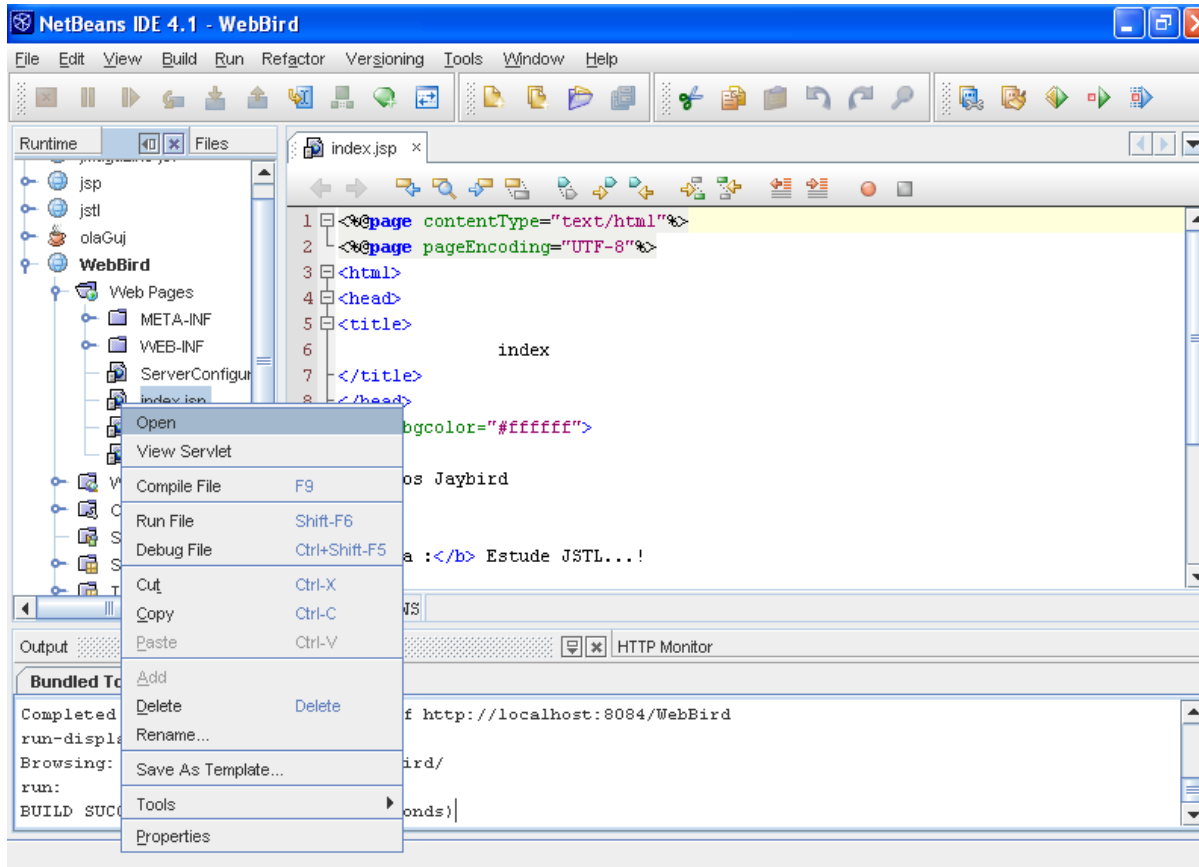


Figura 49

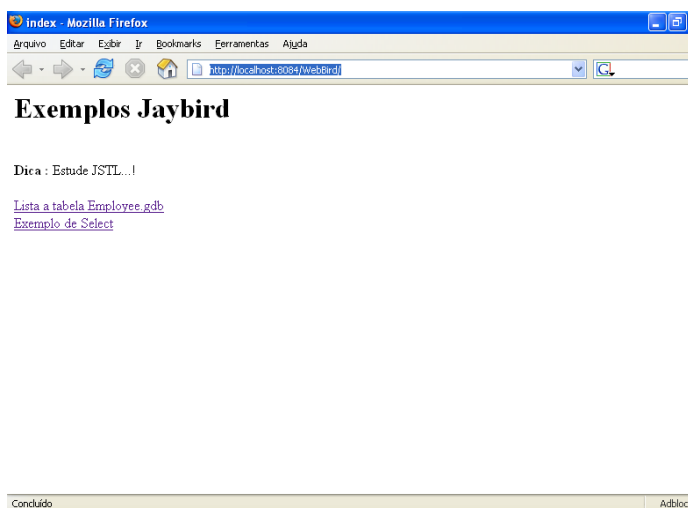


Figura 50

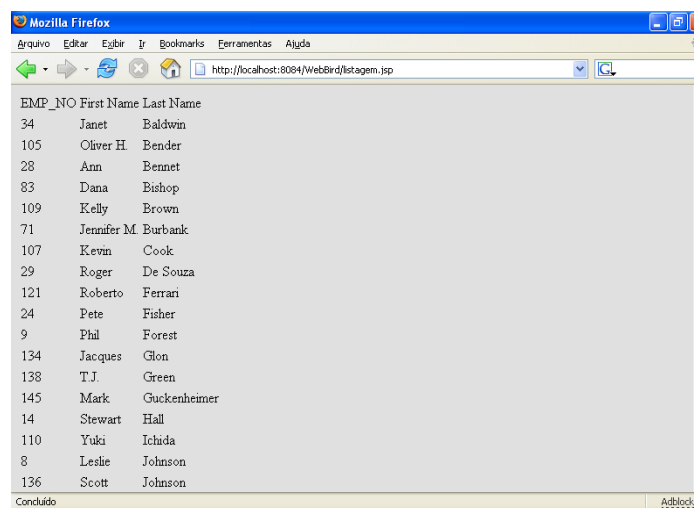


Figura 51

## Conclusão

Este tutorial mostra a IDE netBeans e como ela é completa e flexível, onde o desenvolvedor iniciante ou avançado pode criar qualquer tipo de projeto. Existem mais recursos oferecidos por essa plataforma mais será motivo para tutoriais futuros e será possível explorar o desenvolvimento de aplicações usando refatoração de código, Junit para teste de código, build com Ant e etc.

Espero ter sido claro em minhas exposições, pois tentei aqui ser o mais claro possível para que esse contacto com a IDE o auxiliem no aprendizado da linguagem Java.

## Leituras recomendadas

Anselmo, Fernando. Tudo que você queria saber sobre JDBC, Visual Books,2001.

Anselmo, Fernando. Tudo que você queria saber sobre O JSP..quando utiliza o Servidor Tomcat com o Banco MySql, Visual Books,2002.

Mecenas, Ivan. Java2 Fundamentos,Swing e JDBC, AltaBooks ,2002.

Deitel,H.M. Java como programar 4º edição,Bookmam,2002.

Revista Java Magazine.

Revista Mundo Java.

Revista SQLMagazine.

**William Silva** ([wsilva@iron.com.br](mailto:wsilva@iron.com.br)), complementa o curso de Análise de Sistemas na Unaerp-Campus Guarujá, é consultor independente há mais de 12 anos desenvolvendo sistemas e projetos de redes. Atualmente pesquisa o armazenamento de imagens médicas e, prontuário eletrônico em sistemas distribuídos de telemedicina e educação a distância.