

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ**  
**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**JOSUÉ BATISTA CAMARÃO FURTADO**

**DESENVOLVIMENTO DE UM SISTEMA DE CADASTRO DE INFORMAÇÕES, QUE CADASTRE EM UM BANCO DE DADOS AS INFORMAÇÕES RELATIVAS AOS ALUNOS E PROFESSORES PARA UMA INSTITUIÇÃO DE ENSINO (UNIVERSIDADE OU FACULDADE).**

**ALTAMIRA-PA**

**2019**

**JOSUÉ BATISTA CAMARÃO FURTADO**

**DESENVOLVIMENTO DE UM SISTEMA DE CADASTRO DE INFORMAÇÕES, QUE CADASTRE EM UM BANCO DE DADOS INFORMAÇÕES RELATIVAS AOS ALUNOS E PROFESSORES PARA UMA INSTITUIÇÃO DE ENSINO (UNIVERSIDADE OU FACULDADE).**

Segunda Avaliação de Programação Orientada a Objetos.

Desenvolvimento de um Sistema de Cadastro de informações, trabalho esse que deve ser submetido em um ficheiro que contenha relatório que detalhe e explique o passo a passo da execução e funcionalidades do seu sistema, exibindo as telas implementadas a execução os código referentes as funcionalidades seguidos da explicação.

O sistema também deverá ser apresentado no laboratório individualmente ao professor da disciplina.

Instituto Federal de Educação, Ciência e Tecnologia do Pará.

Instrutor: Paulo César de Almeida Júnior

**ALTAMIRA-PA**

**2019**

## PROJETO JAVA CRUD

Esse Projeto foi uma iniciativa do professor Paulo César como forma de 2º avaliação da disciplina de Programação Orientada a Objeto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Pará - Campos Altamira.

Tal projeto tem como finalidade o desenvolvimento de um sistema de cadastro de informações, que cadastre em um banco de dados informações relativas aos alunos e professores para uma instituição de ensino (universidade ou faculdade).

CRUD (acrônimo do inglês Create, Read, Update and Delete) são as quatro operações básicas (criação, consulta, atualização e destruição de dados).

## BANCO DE DADO

O BD(Banco de Dado) foi criado em MySQL, através da ferramenta de gerenciamento MySQL Workbench.

O MySQL pode ser encontrado na página abaixo:

<https://www.mysql.com/downloads/>

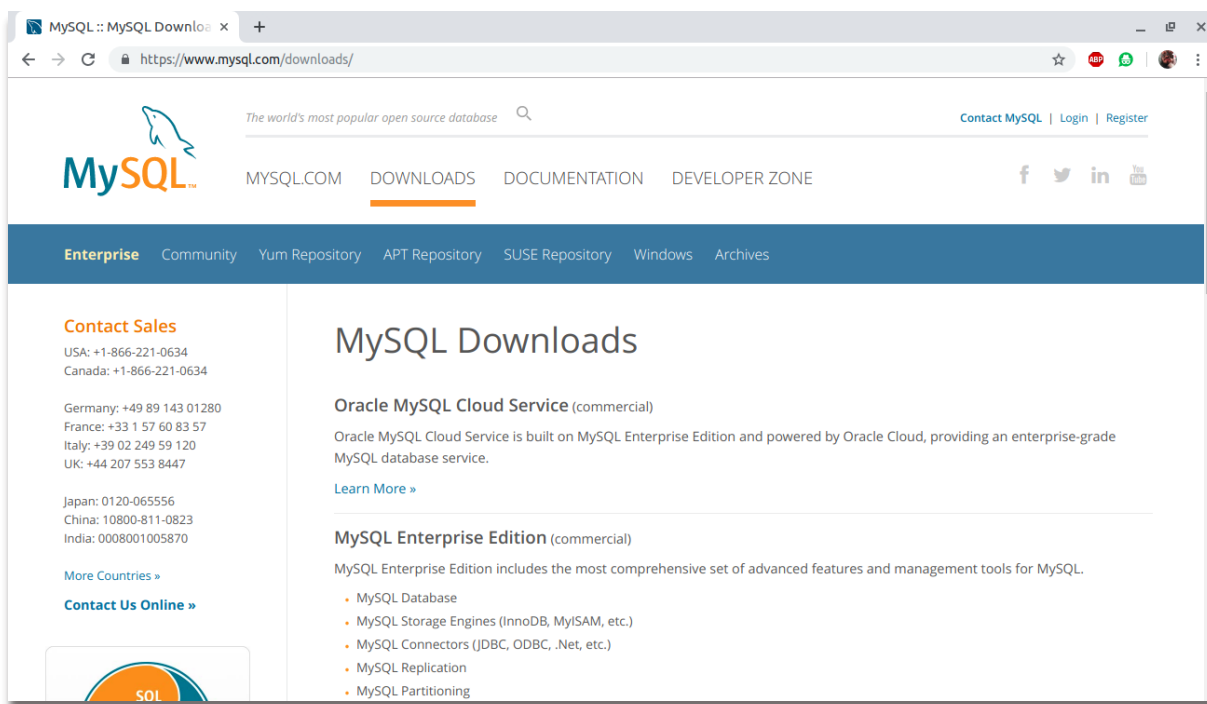


Imagem 1.0

O MySQL Workbench é uma ferramenta de design de banco de dados visual que integra o desenvolvimento, administração, design de banco de dados, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL.

O MySQL Workbench pode ser baixado na página abaixo:

<https://dev.mysql.com/downloads/workbench/>

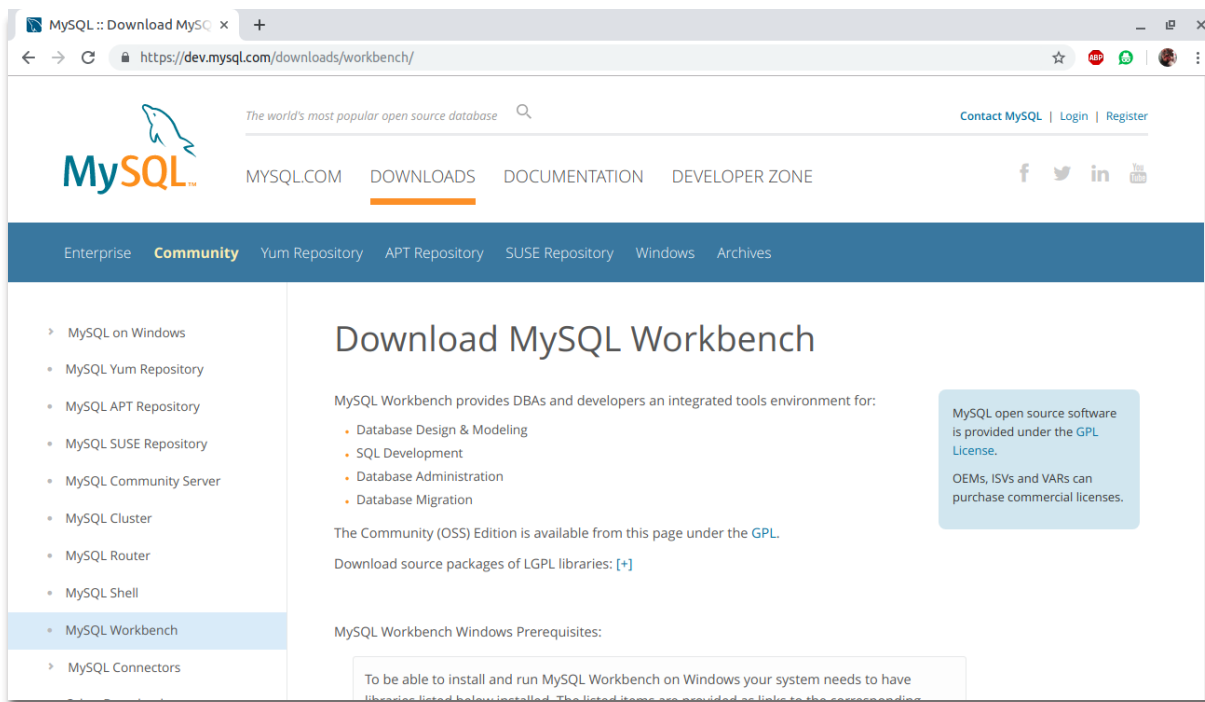


Imagem 1.1

MySQL Workbench instalado:

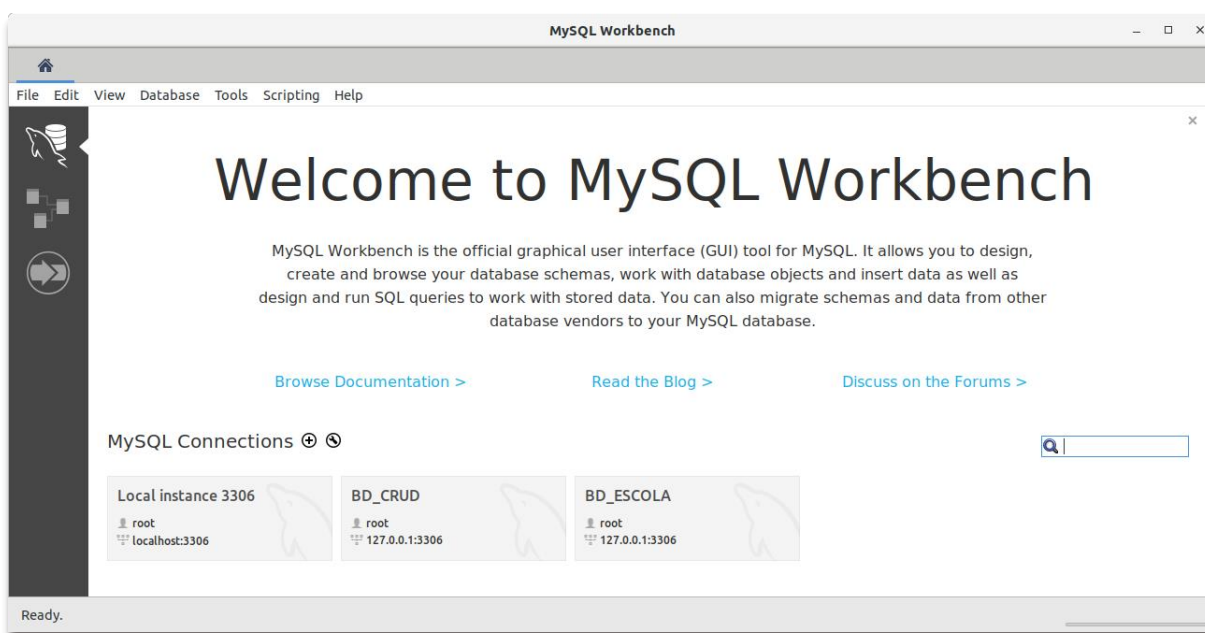


Imagem 1.2

Foram criadas apenas duas tabelas “PROFESSOR” e “ALUNO” com seus respectivos atributos:

PPROFESSOR: Nome, RG, CPF, Endereço, Bairro, Telefone, Email, Sexo, Disciplina e Atuação.

ALUNO: Nome, RG, CPF, Endereço, Bairro, Telefone, Sexo, Curso e Período.

## MODELO LÓGICO

O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

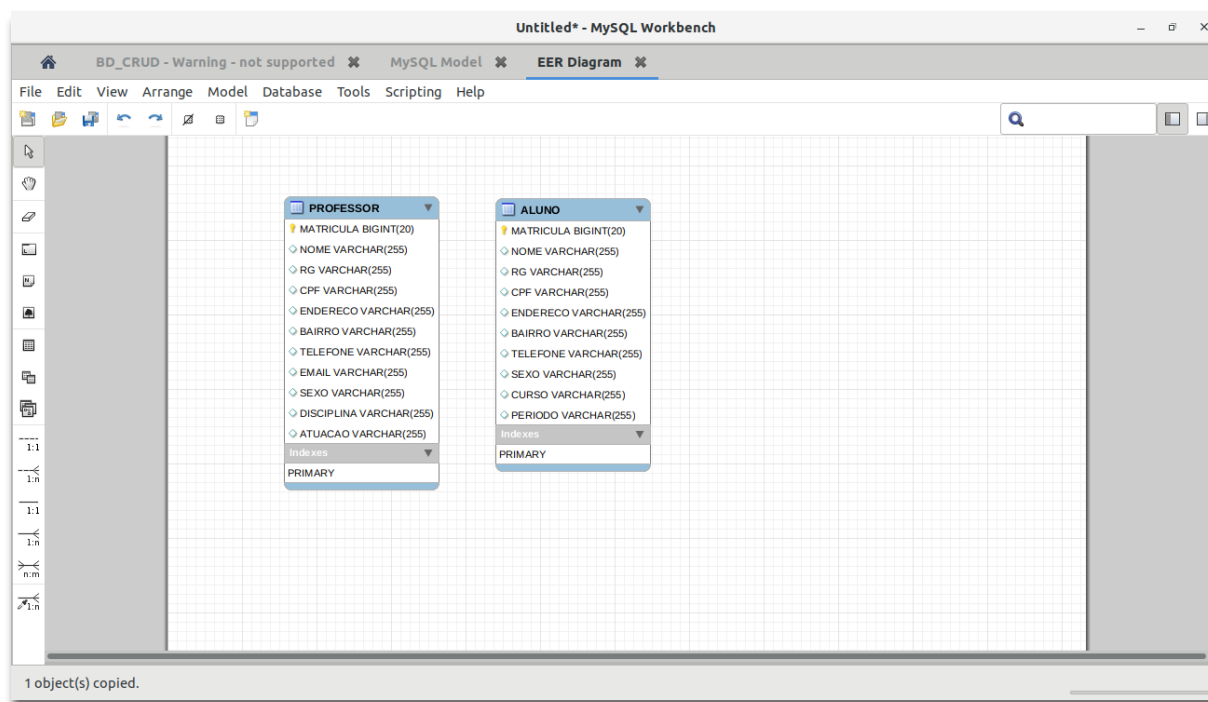


Imagem 1.3

## LINGUAGEM SQL

Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional.

A linguagem SQL foi desenvolvida pela IBM no final dos anos 70, tendo sido a linguagem adotada em seu protótipo de banco de dados relacional denominado System R. O primeiro banco de dados comercial a utilizar a linguagem SQL foi apresentado em 1979, pela Oracle.

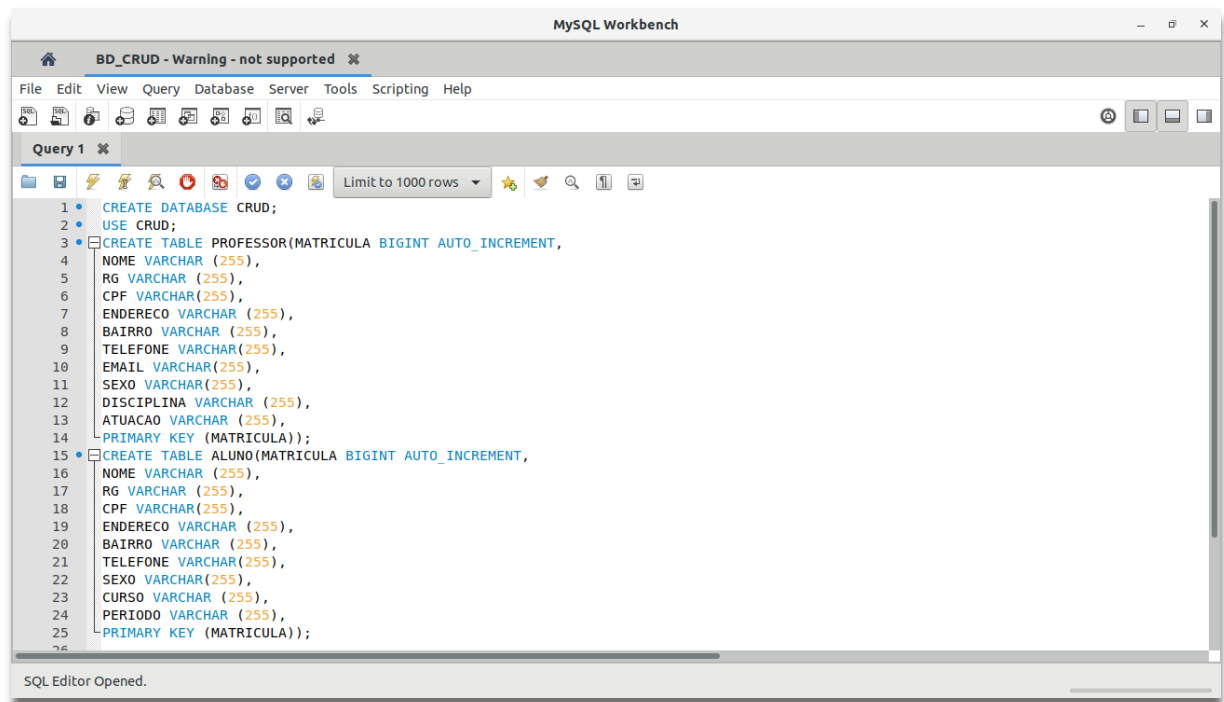


Imagem 1.4

## XAMPP

Foi utilizado também o XAMPP v. 7.3.5-0 que é um servidor independente de plataforma, software livre, que consiste principalmente na base de dados MySQL, o qual foi substituído pelo MariaDB, o servidor web Apache e os interpretadores para linguagens de script: PHP e Perl. O nome provem da abreviação de X, Apache, MySQLDB, PHP, Perl.

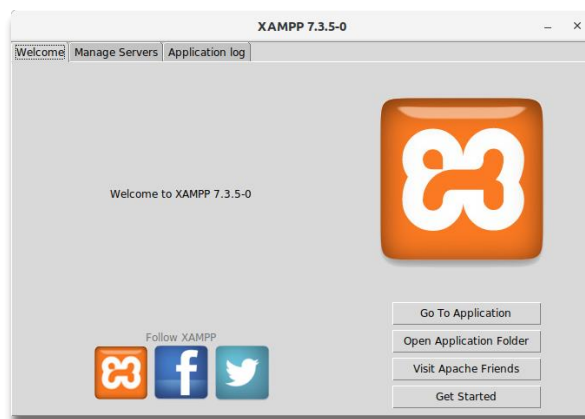


Imagem 2.0

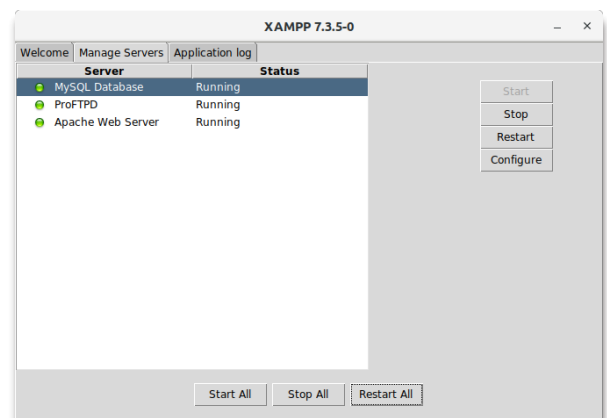


Imagem 2.1

Após a criação do Banco de Dado (BD\_CRUD), partimos para o desenvolvimento do Sistema.

## JDK + NETBEANS

Nessa etapa foi utilizado o Java Development Kit (JDK) que é uma ferramenta disponibilizada pela Oracle que tem por finalidade distribuir todos os recursos necessários para desenvolver uma aplicação Java, esses recursos são: Java Virtual Machine (JVM), bibliotecas e o compilador. Em outras palavras, quando queremos desenvolver uma aplicação na linguagem Java, precisamos instalar o JDK.

Foi utilizado também a IDE oficial da Oracle, o NetBeans IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software nas linguagens Java, JavaScript, HTML5, PHP, C/C++, Groovy, Ruby, entre outras. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS.

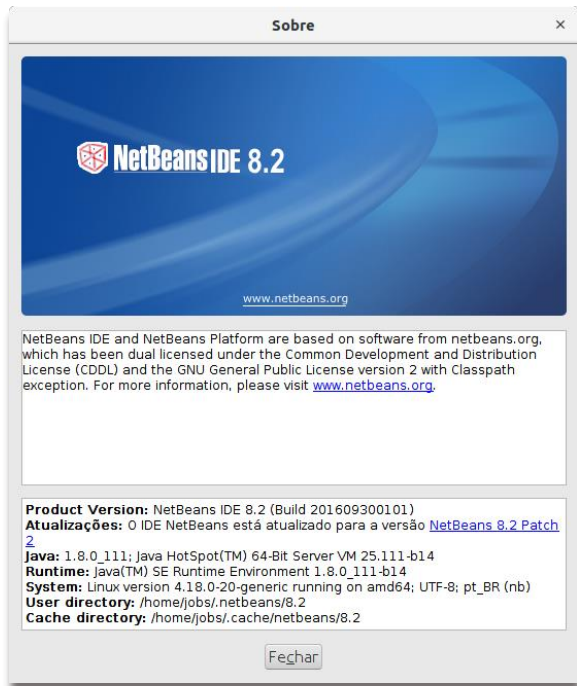
Tanto o JDK (Kit de Desenvolvimento Java) e o NetBeans foram adquirido no próprio site da Oracle:

The screenshot shows the Oracle Java SE Downloads page for JDK 8u111 with NetBeans 8.2. The page is in Portuguese and features the Oracle logo at the top. The main content area is titled "JDK 8u111 with NetBeans 8.2" and includes a description of the distribution, a license agreement, and a table of download links for various operating systems. The table has three columns: Product / File Description, File Size, and Download. The download links are provided for Linux x86, Linux x64, Mac OS X x64, Windows x86, and Windows x64. The page also includes a sidebar with navigation links and a right sidebar with additional resources.

Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	<a href="#">jdk-8u111-nb-8_2-linux-i586.sh</a>
Linux x64	282.57 MB	<a href="#">jdk-8u111-nb-8_2-linux-x64.sh</a>
Mac OS X x64	342.99 MB	<a href="#">jdk-8u111-nb-8_2-macosx-x64.dmg</a>
Windows x86	317.21 MB	<a href="#">jdk-8u111-nb-8_2-windows-i586.exe</a>
Windows x64	326.03 MB	<a href="#">jdk-8u111-nb-8_2-windows-x64.exe</a>

Imagem 3.0

<https://www.oracle.com/technetwork/pt/java/javase/downloads/jdk-netbeans-jsp-3413153-ptb.html>



## VERSÃO NETBEANS

O NetBeans IDE e a Plataforma NetBeans são baseados no software do [netbeans.org](http://netbeans.org), que foi licenciado sob a Licença de Desenvolvimento Comum e Distribuição (CDDL) e a Licença Pública Geral GNU versão 2 com a exceção Classpath. Para mais informações, visite

[www.netbeans.org](http://www.netbeans.org).

Product Version: NetBeans IDE 8.2 (Build 201609300101)

Atualizações: O IDE NetBeans está atualizado para a versão NetBeans 8.2 Patch 2

Java: 1.8.0\_111; Java HotSpot(TM) 64-Bit Server VM 25.111-b14

Runtime: Java(TM) SE Runtime Environment 1.8.0\_111-b14

System: Linux version 4.18.0-20-generic running on amd64; UTF-8; pt\_BR (nb)

User directory: /home/jobs/.netbeans/8.2

Cache directory: /home/jobs/.cache/netbeans/8.2

## CRIANDO O PROJETO JAVA CRUD

O Projeto Java CRUD foi elaborado primeiramente com a criação 10 Pacotes e suas respectivas classes como mostra a imagem 4.0.

O projeto tem como finalidade o desenvolvimento de um sistema de cadastro de informações, que cadastre em um banco de dados informações relativas aos alunos e professores para uma instituição de ensino (universidade ou faculdade). A linguagem de programação utilizada foi o Java que é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Em 2008 o Java foi adquirido pela empresa Oracle Corporation.

CRUD (acrônimo do inglês Create, Read, Update and Delete) são as quatro operações básicas (criação, consulta, atualização e destruição de dados).



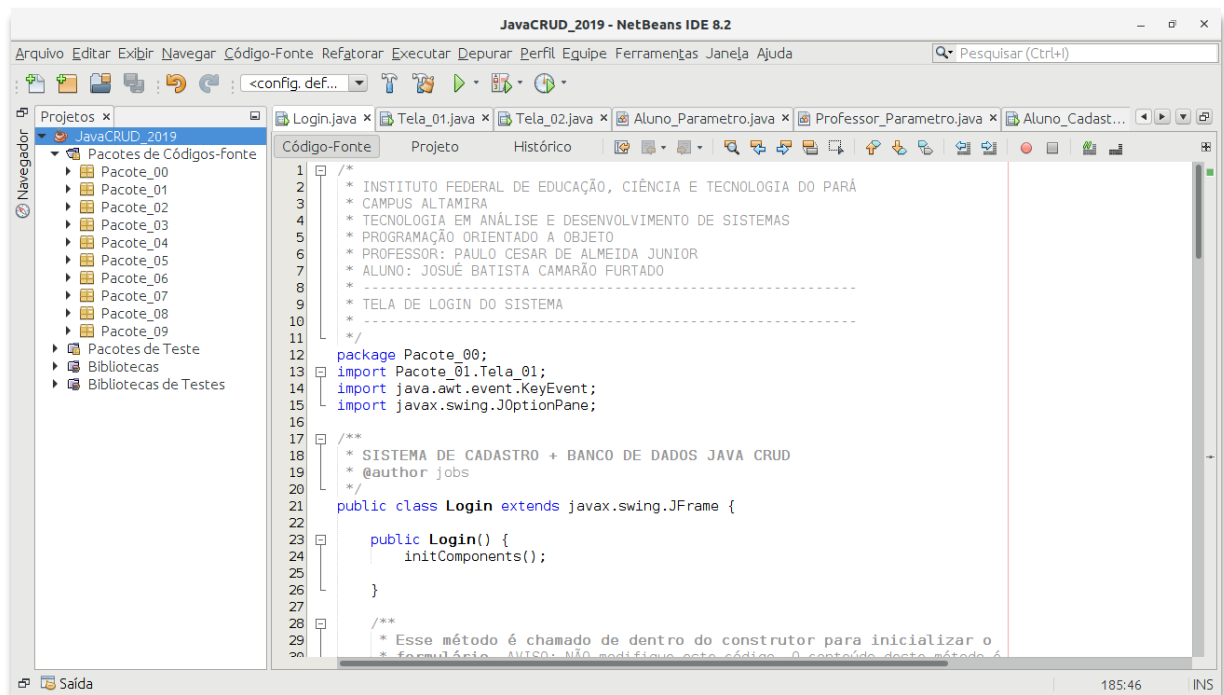


Imagem 4.0

## PACOTE\_00

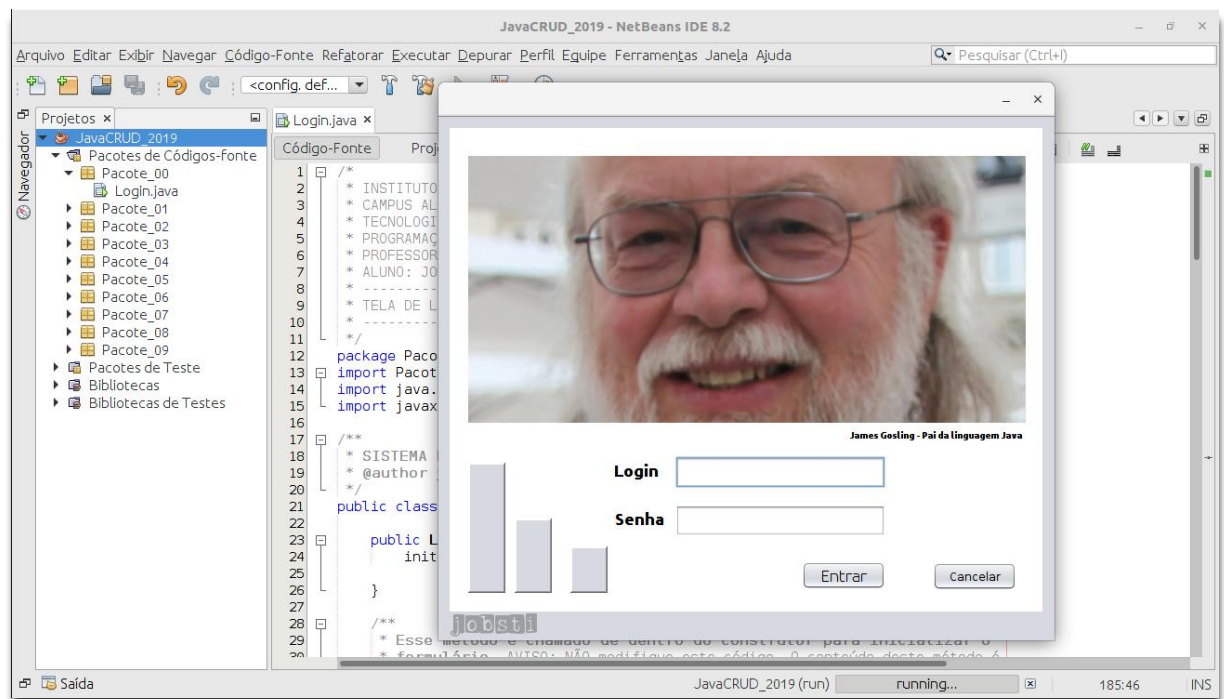


Imagem 4.1

O Pacote 00, traz a primeira Tela do Sistema (TELA DE LOGIN), onde o Administrador (root) do Sistema irá logar para efetuar os cadastros.

## MÉTODOS TELA LOGIN

```
private void BT_ENTERActionPerformed(java.awt.event.ActionEvent evt) {  
    /**  
     * O código abaixo tem a função de chamar a tela Principal se a condição  
     * for satisfeita. Nesse caso se o login e senha forem "root". Obs. toda  
     * função com um traço como por exemplo o (getText) abaixo, significa  
     * que existe que já existe uma atualização para a mesma, nesse caso  
     * usarei ela mesma.  
     */  
    if (CX_LOGIN.getText().equals("root") && CX_SENHA.getText().equals("root")) {  
        Tela_01 TelaPrincipal = new Tela_01();  
        TelaPrincipal.show();  
        dispose();  
    } else {  
        JOptionPane.showMessageDialog(null, "Login ou Senha incorretos!");  
        /**  
         * O setText abaixo é para que quando o usuário errar o login ou a  
         * senha ele chame novamente a tela login com o campo Login e Senha  
         * vazios para que o usuário tente novamente.  
         */  
        CX_LOGIN.setText("");  
        CX_SENHA.setText("");  
    }  
}
```

Imagem 4.2

### Método equals:

if (CX\_LOGIN.getText().equals("root") && CX\_SENHA.getText().equals("root"))

Esse método compara dois objetos quanto à igualdade e retorna true se eles forem iguais; caso contrário, retorna false . O método aceita qualquer Object como um argumento. Quando os objetos de uma classe particular precisarem ser comparados em relação à igualdade, a classe deve sobrescrever o método equals a fim de comparar o conteúdo dos dois objetos.

### Métodos getText, setText

O método getText pode ser usado para recuperar o texto atual do JLabel, o método setText pode ser chamado em qualquer JLabel para alterar seu texto. Esse JLabel exibe tanto o ícone como o texto.

O código printado acima tem a função de chamar a tela Principal se a condição for satisfeita, nesse caso se o login e senha forem "root".

CX\_LOGIN.setText("");

CX\_SENHA.setText("");

É para que quando o usuário errar o login ou a senha ele chame novamente a tela login com o campo Login e Senha vazios para que o usuário tente novamente.

```

if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
    if (CX_LOGIN.getText().equals("root") && CX_SENHA.getText().equals("root")) {
        Tela_01 TelaPrincipal = new Tela_01();
        TelaPrincipal.show();
        dispose();
    } else {
        JOptionPane.showMessageDialog(null, "Login ou Senha incorretos!");
    }
}

```

Imagem 4.3

Código acima (if(evt.getKeyCode() == KeyEvent.VK\_ENTER)) tem a função de acessar o sistema sem precisar clicar no Botão Enter, bastando assim apenas a tecla Enter do teclado.

### PACOTE\_01



Imagem 5.0

```

private void BT_ACESSOActionPerformed(java.awt.event.ActionEvent evt) {
    Tela_02 PuxaTela = new Tela_02();
    PuxaTela.setVisible(true);
    dispose();
}

```

Imagem 5.1

Botão que chama a tela 2 para escolha do cadastro do Professor ou Aluno ou até mesmo Sair do Sistema.

```
try {
    Desktop.getDesktop().browse(new URI("http://www.jobsti.net.br"));
} catch (URISyntaxException | IOException ex) {
    Logger.getLogger(Link.class.getName()).log(Level.SEVERE, null, ex);
}
```

Imagem 5.2

O código acima direciona para meu site (jobsti.net.br) onde postei todo projeto passo a passo.

```
try {
    Desktop.getDesktop().browse(new URI("http://www.github.com/JBCFurtado"));
} catch (URISyntaxException | IOException ex) {
    Logger.getLogger(Link.class.getName()).log(Level.SEVERE, null, ex);
}
```

Imagem 5.3

O código acima direciona para meu github ("http://www.github.com/JBCFurtado") onde está todo Projeto Java CRUD

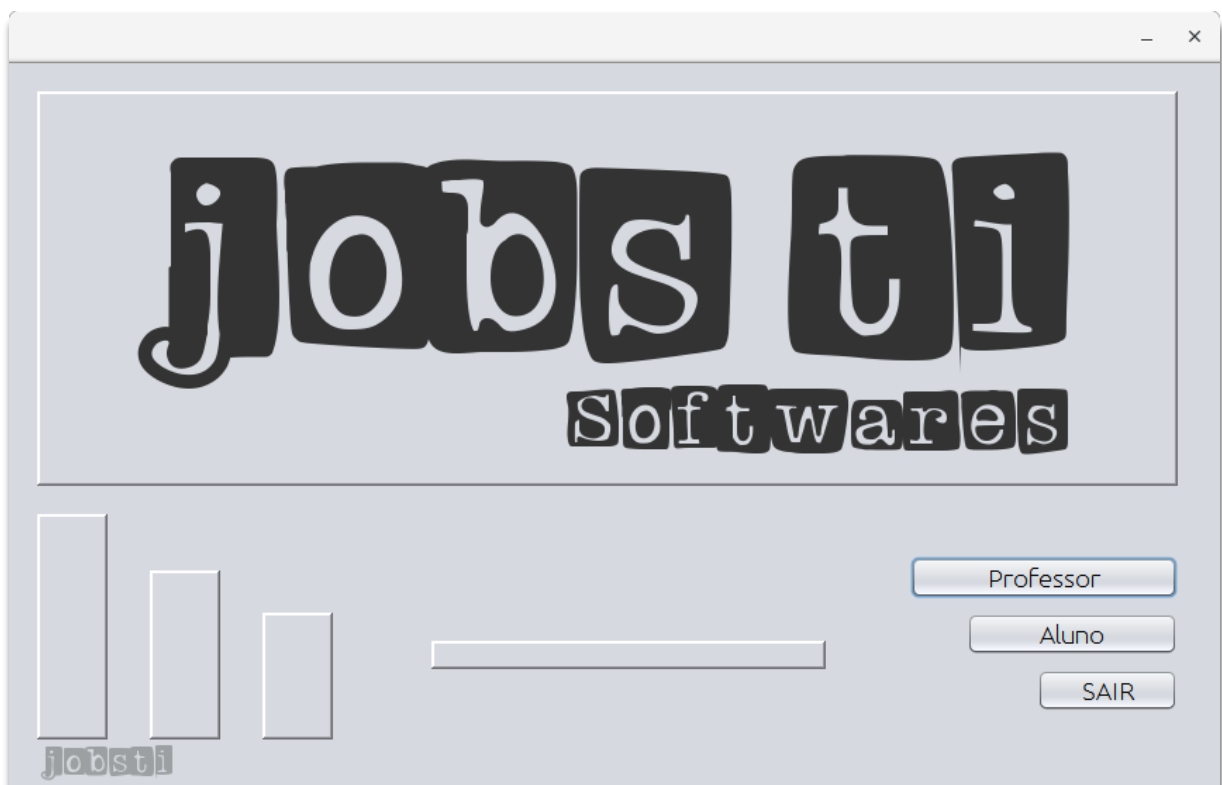


Imagem 5.4

```

private void BT_PROFESSORActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        /**
         * Tratamento responsável em avisar para o usuário se o banco de
         * dados estiver offline.
         */
        Professor_Cadastro PuxaTela = new Professor_Cadastro();
        PuxaTela.setVisible(true);
        dispose();
    } catch (Exception ti) {
        JOptionPane.showMessageDialog(null, "Banco de Dado OFFLINE, verifique sua conexão!");
    }
}

private void BT_ALUNOActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        /**
         * Tratamento responsável em avisar para o usuário se o banco de
         * dados estiver offline.
         */
        Aluno_Cadastro PuxaTela = new Aluno_Cadastro();
        PuxaTela.setVisible(true);
        dispose();
    } catch (Exception i) {
        JOptionPane.showMessageDialog(null, "Banco de Dado OFFLINE, verifique sua conexão!");
    }
}

```

Imagem 5.5

Códigos acima chamam as duas telas de cadastro “PROFESSOR” e “ALUNO” abaixo com tratamento caso o BD esteja offline.

## CADASTRO PROFESSOR

\* Campos Obrigatórios

Matrícula  \*

Nome  \*

Sexo

RG  \*

CPF  \*

Endereço  \*

Bairro

Telefone  \*

Email  \*

Disciplina

Atuação

Salvar

Professores Cadastrados

Voltar

jobsti

Imagem 5.6

**CADASTRO ALUNO**

\* Campos Obrigatórios

Matrícula  \*

Nome  \* Sexo

RG  \*

CPF  \*

Endereço  \* Período

Bairro

Telefone  \*

Curso

Imagem 5.7

Telas acima fazem a inserção dos cadastros professores e alunos, e permitem além de salvar o cadastro, levam a lista de professores e alunos cadastrados.

Os campos com \* (asteriscos) são obrigatórios.

## PACOTE\_02

O Pacote\_02 armazena todos os métodos “get” e “set” para acesso a tudo o que é privado dividido em duas classes: Aluno\_Parametro e Professor\_Parametro, segue tela abaixo:

- Usamos o get para obter informações.
- Usamos set para definir valores.

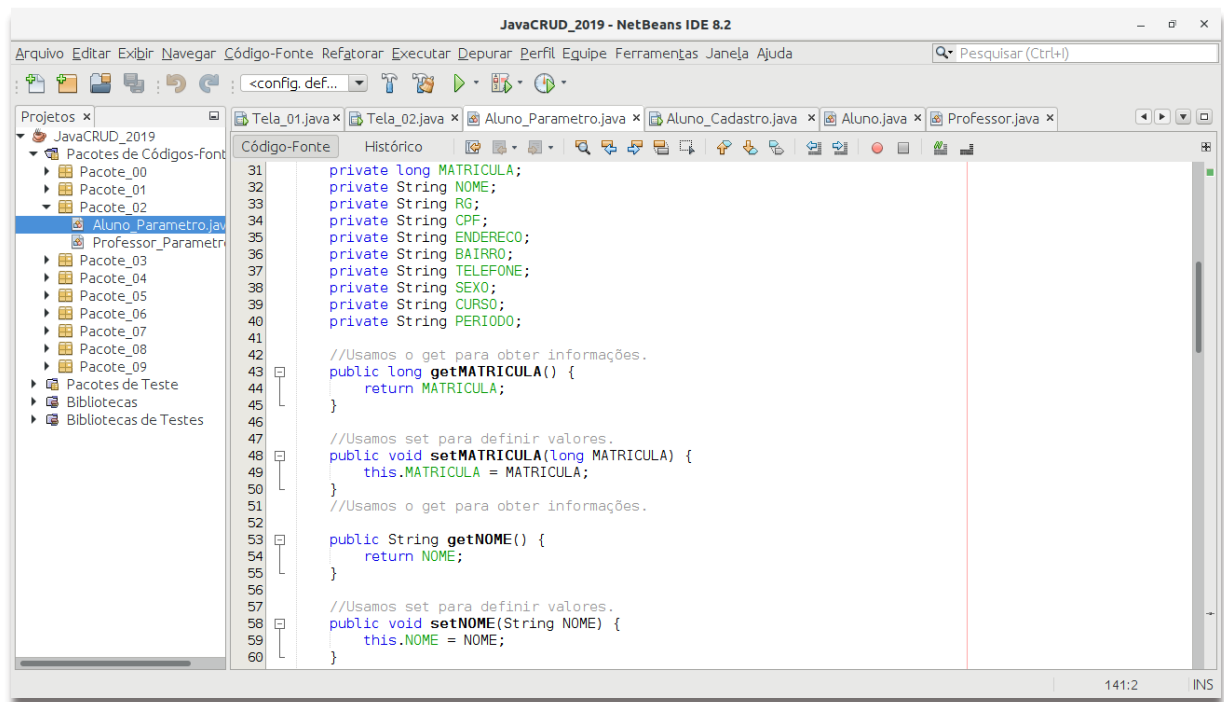


Imagem 6.0

## PACOTE\_03

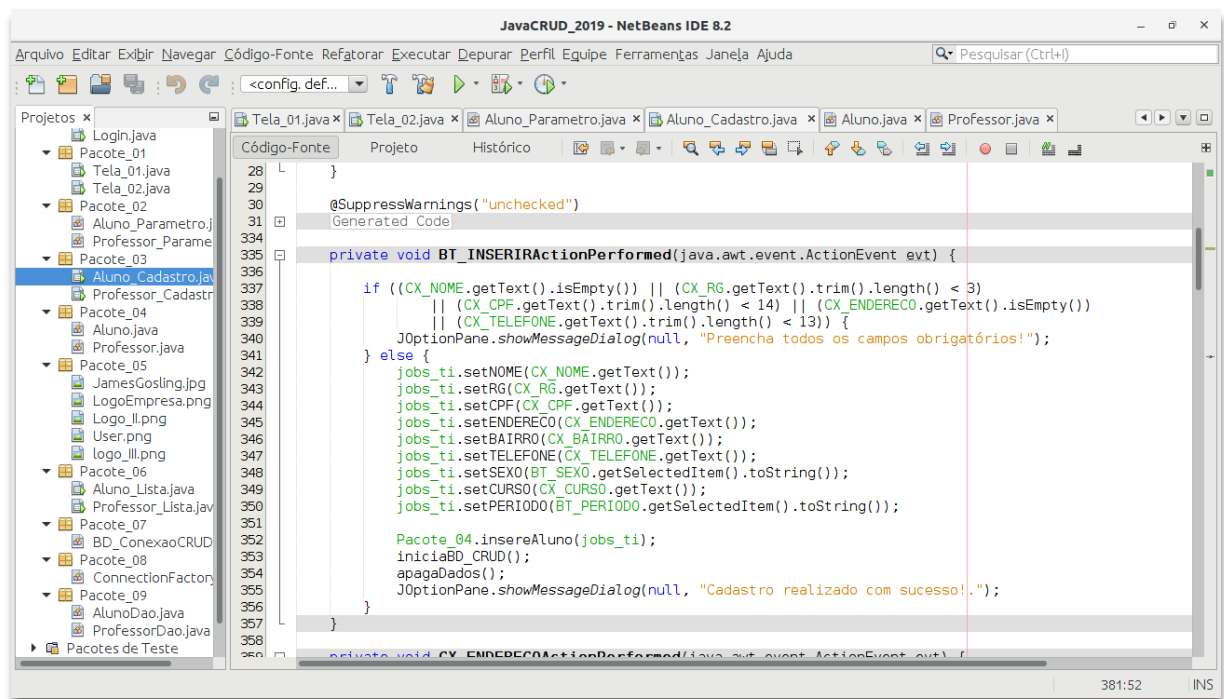


Imagem 7.0

Esse Pacote recebe todos os atributos do cadastro “Professor” e “Aluno”, e define os campos obrigatórios das telas abaixo 2.1 e 2.0 através dos métodos get’s e set’s.

## PACOTE\_04



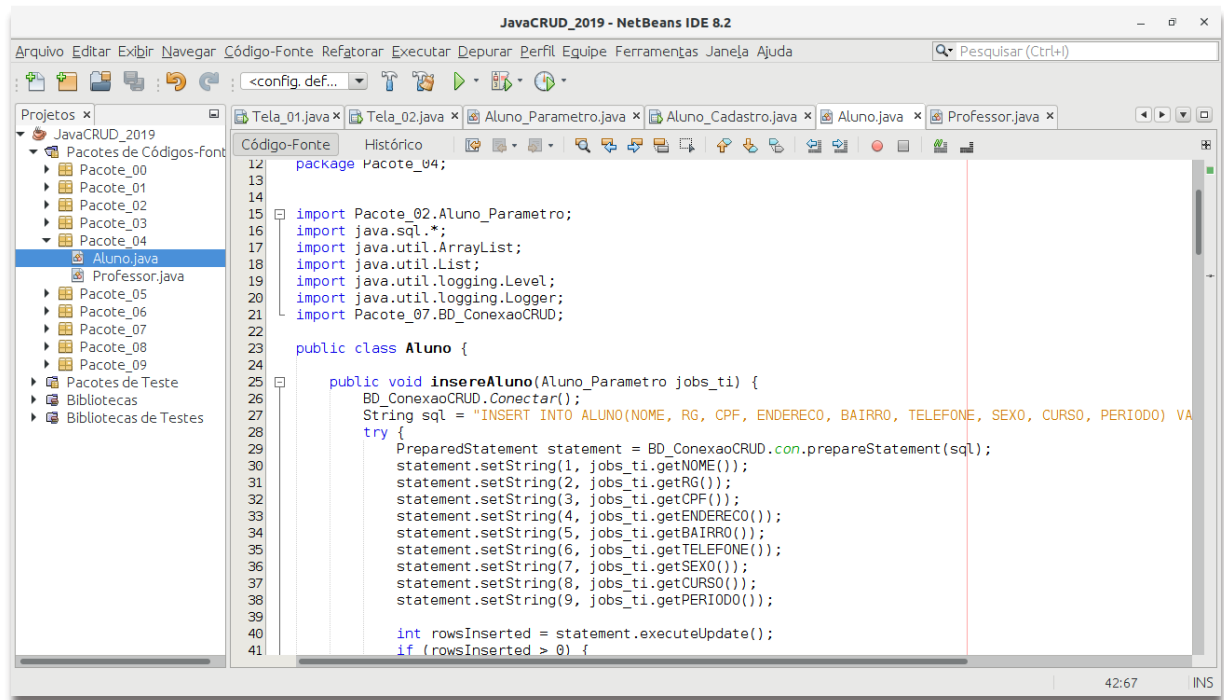


Imagem 8.0

O Pacote\_04 traz todos os “SET’s” que utilizamos para obtermos todas as informações dos “Professores” e “Alunos” cadastrados no sistemas através do BD.A linha 29 invoca o método Connection createStatement para obter um objeto que implementa a interface Statement (pacote java.sql ). O programa utiliza o objeto Statement para enviar instruções de SQL ao banco de dados.

## PACOTE\_05

O Pacote 05 contém todas as imagens utilizadas que foram utilizadas nesse projeto:

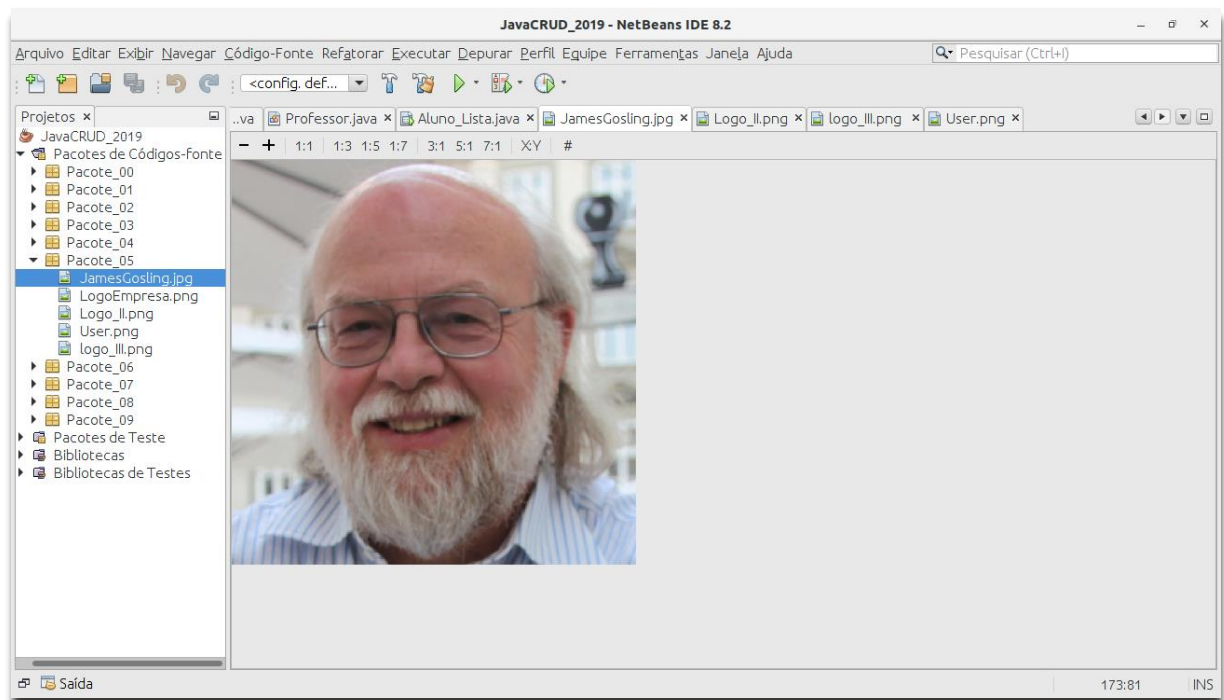


Imagem 9.0  
**PACOTE\_06**

**PROFESSORES CADASTRADOS**

MA...	NOME	RG	CPF	ENDEREÇO	BAIRRO	TELEFONE	EMAIL	SEXO	DISCIPLINA	ATUAÇÃO
1	Root									
2	Paulo Cesar Jr.	4556456411	456.454.546-54	Rua 7, Nº 1524	Centro	(93) 9 9216-5...	paulo.cesar@gmail.com	Masculino	Programação	Educação
3	Eduardo Krause	4156454564	564.564.545-45	Rua 20, Nº 255	Jatobá	(87) 6 8455-3...	edu@gmail.com	Masculino	Geografia	Educação
4	Josué Batista	2342343	321.211.212-12	Rua Blitz, Nº 666	Independente I	(23) 4 3242-3...	jobs.ti@gmail.com	Masculino	Programação	Analista de Sistema
6	Jardel Castro	1234654	541.564.654-64	Rua 7	Centro	(89) 7 8979-8...	jardel@gmail.com	Masculino	Redes	Educação
7	Gessé Jorge	654654124	456.465.456-45	Rua 10	Central	(65) 4 5646-5...	gescafe@gmail.com	Masculino	Java	Tecnologia
9	Diego Rafael	789454115	564.545.645-64	Av. Salin, 154	Park	(98) 4 5616-5...	diego@gmail.com	Masculino	Programação	Tecnologia
10	Paulo Cesar	1564156646	545.645.646-54	Rua 7, N 255	Centro	(93) 4 6545-6...	paulo@gmail.com	Masculino	Java	Educação

**Atualizar Cadastro**

Matrícula:  Endereço:  Sexo:

Nome:  Bairro:  Atuação:

RG:  Telefone:  Disciplina:

CPF:  Email:

Imagem 10.0

O Pacote 06 mostra em tela todos os cadastros quando selecionados com as opções dos botões remover ou atualizar o cadastro “Professor” ou “Aluno”, além do botão voltar a tela anterior ou a tela de início do sistema.

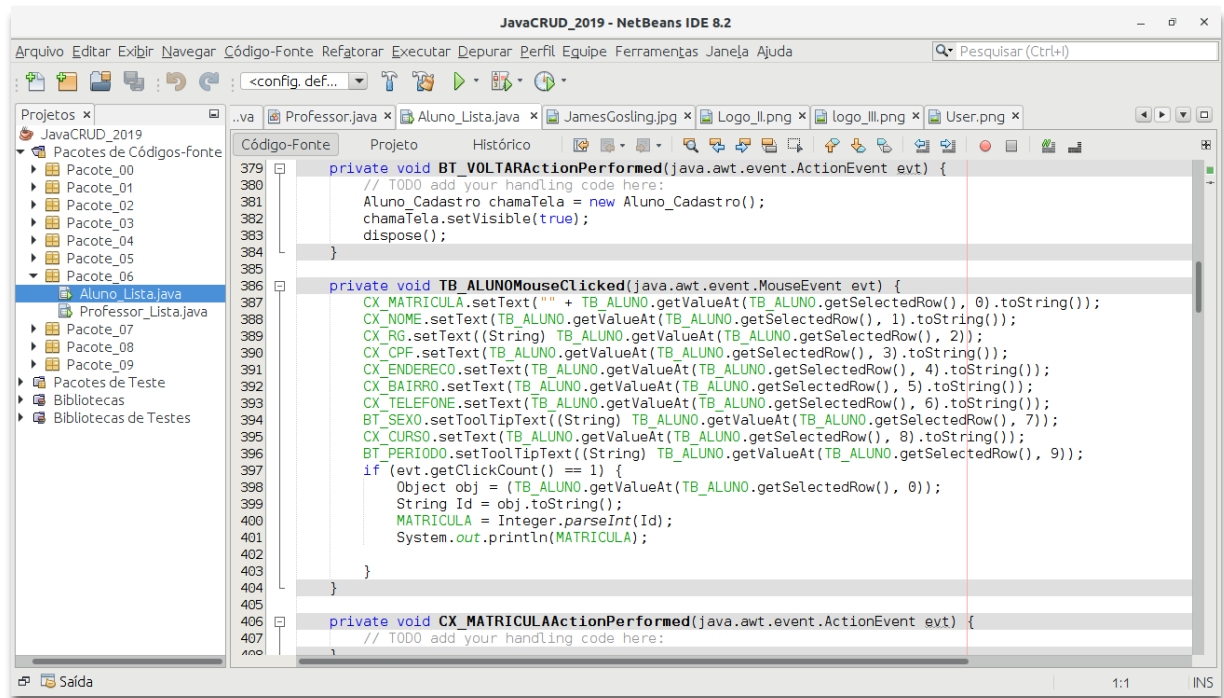


Imagem 10.1

Os métodos acima (get's e set's) são para “Obter” e “Definir” as informações inseridas nesse caso pelos Alunos.

## PACOTE\_07

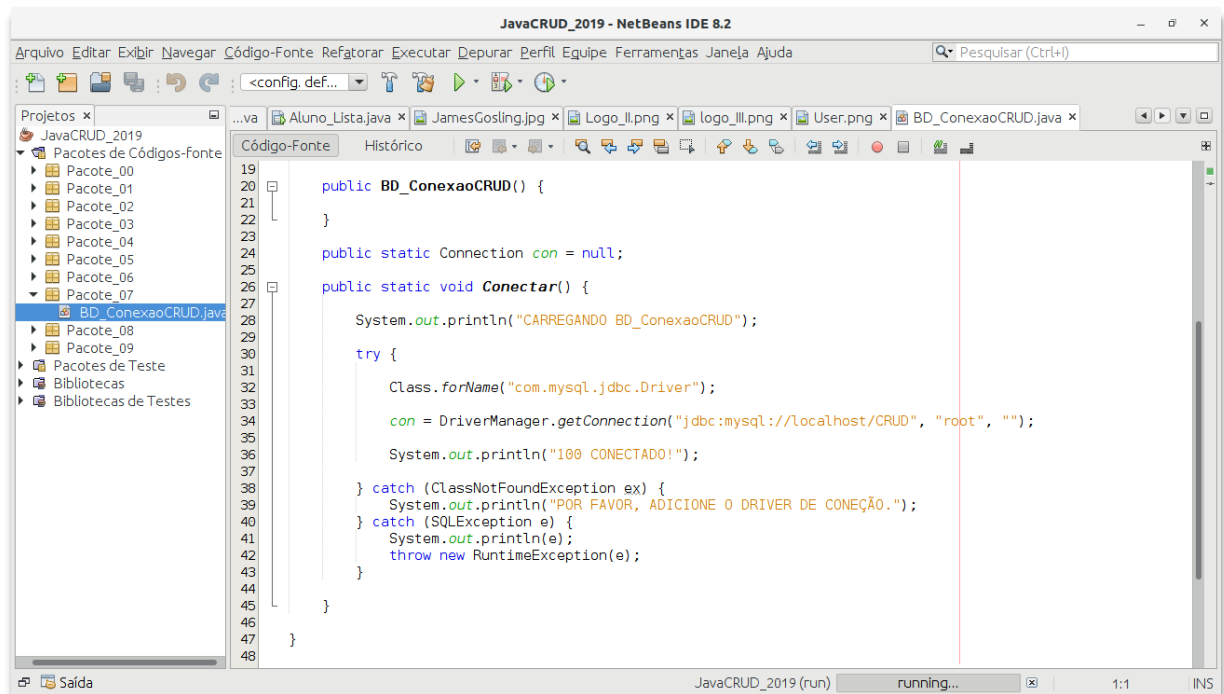


Imagem 11.0

O Pacote 07 traz a conexão do BD\_CRUD, através da classe JDBC do MySQL importada da `import java.sql.DriverManager;`

## PACOTE\_08

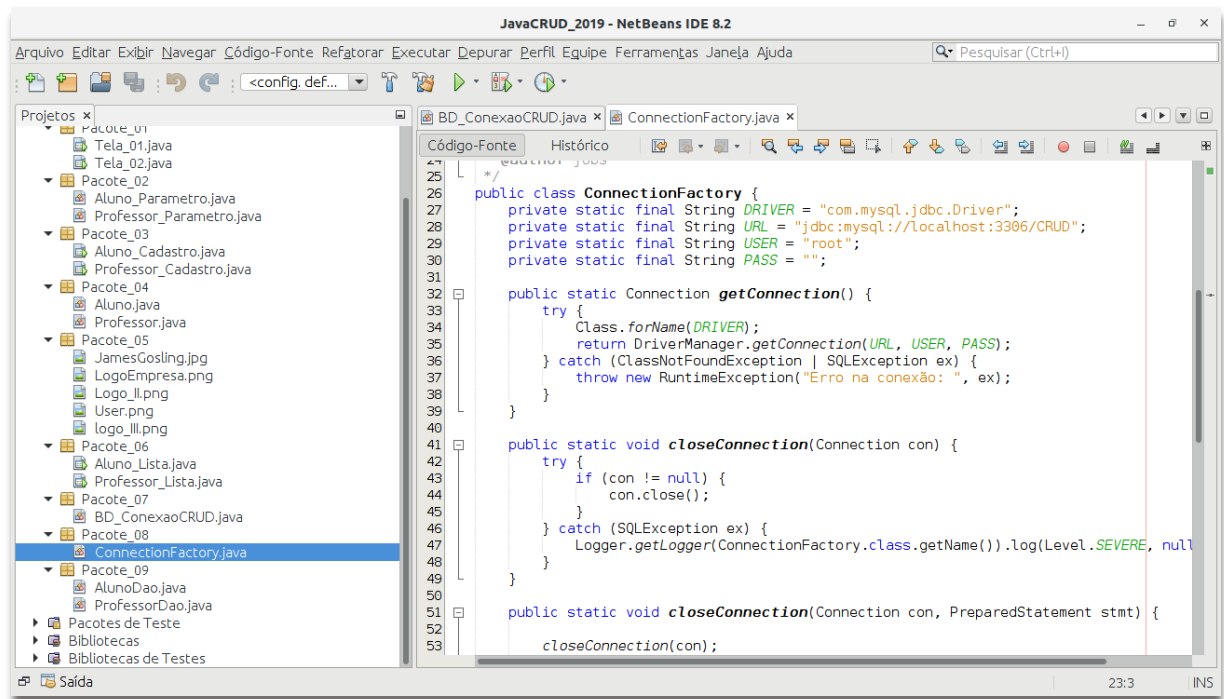


Imagem 12.0

Acessando o BD através do JDBC, Programas Java interagem com bancos de dados usando a API Java Database Connectivity (JDBC™). Um driver JDBC permite aos aplicativos Java conectar-se a um banco de dados em um DBMS particular e permite a você manipular esse banco de dados utilizando o JDBC API.

Define também o User e Password do BD\_CRUD: "root", "sem senha".

## PACOTE\_09

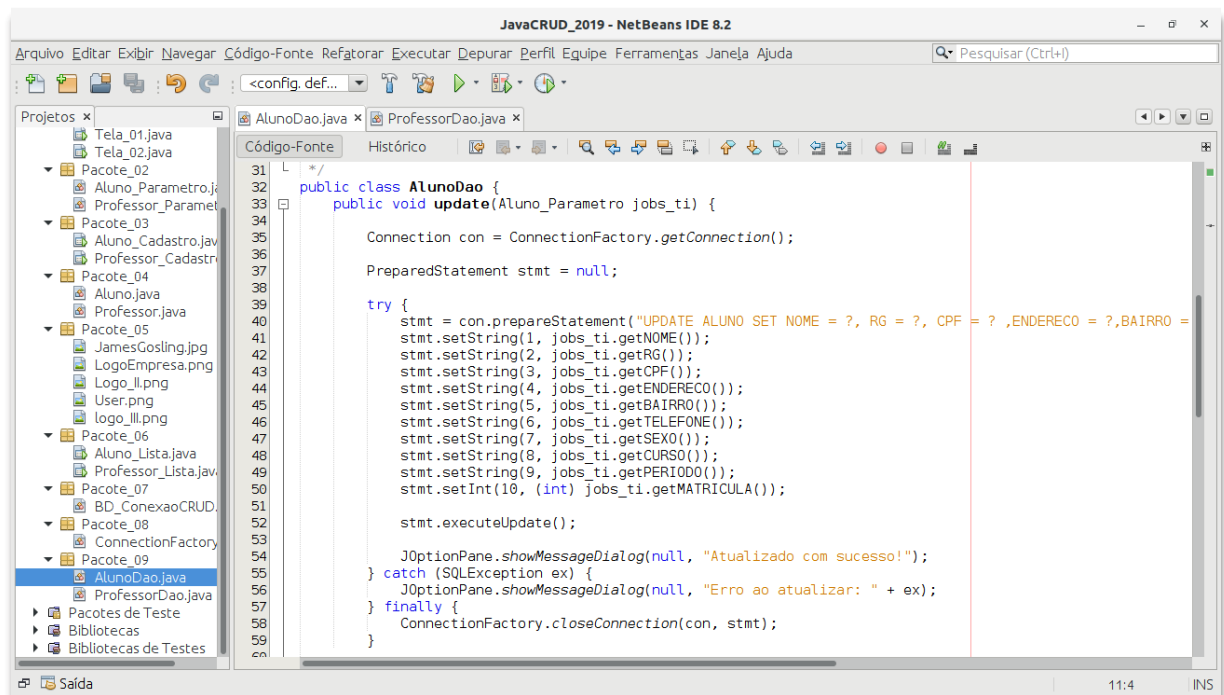


Imagem 13.0

Esse pacote é responsável por conectar o BD\_CRUD e fazer as alterações realizadas pelo administrador nos cadastros “Professores” e “Alunos”.

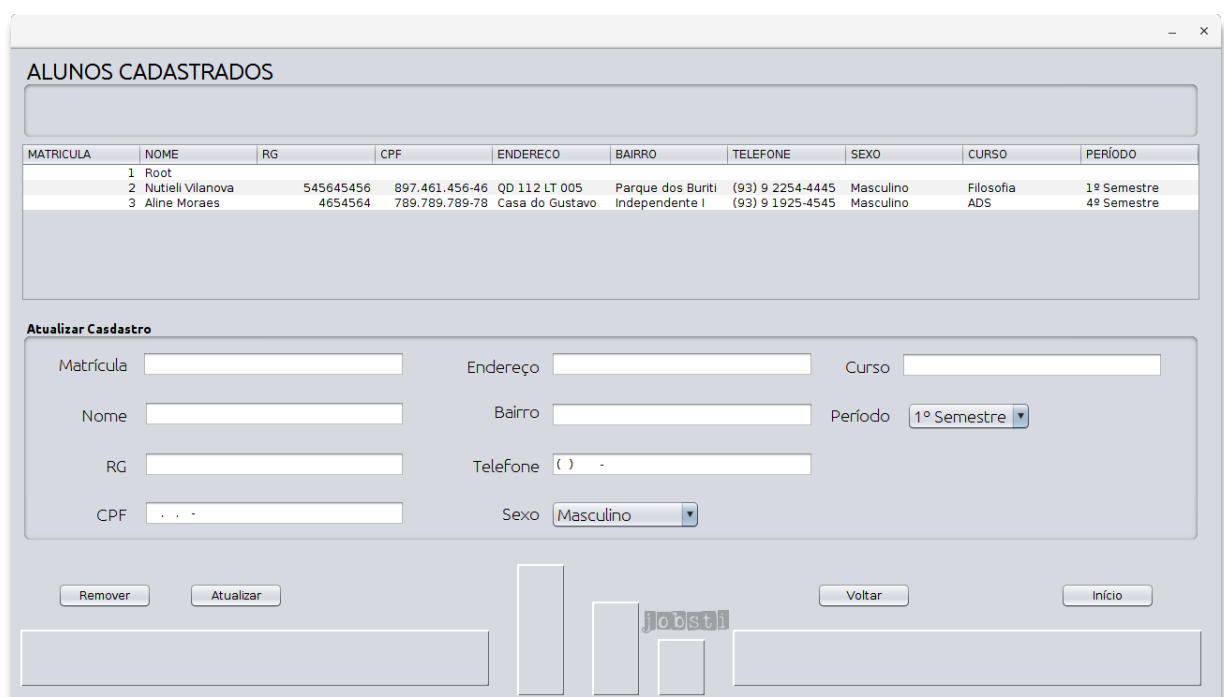


Imagem 13.1

## BIOGRAFIA

### Livros e Apostilas:

#### **JAVA - Como Programar 10ª Edição**

Paul Deitel

Deitel & Associates, Inc.

Harvey Deitel

Deitel & Associates, Inc.

Tradutor:

Edson Furmankiewicz

Docware Traduções Técnicas

Revisão técnica:

Fábio Luis Picelli Lucchini

Bacharel em Ciência da Computação pelo Centro Universitário

Padre Anchieta

Mestre em Engenharia da Computação pela Unicamp (FEEC)

Professor Universitário do Centro Universitário Padre Anchieta

#### **APOSTILA JAVA - Progressivo**

[www.javaprogressivo.net](http://www.javaprogressivo.net)

### Sites e Blogs:

#### **- Pinhata Dev**

<https://pinhatadev.wordpress.com/2016/04/12/sistema-basico-de-cadastro-usando-java-em-netbeans-e-mysql-parte-3/>

#### **- Java Documentation**

<https://www.oracle.com/technetwork/java/javase/documentation/index.html>

### Canais no YouTube:

#### **- Canal Descompila**

<https://www.youtube.com/user/CanalSamuelson>

#### **- Canal G-Tec**

<https://www.youtube.com/user/gtechnfor>