



**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**

---

## **PROTÓTIPO DE UM SISTEMA TUTOR DE ORIENTAÇÃO A OBJETOS**

**ORIENTANDO: Kim W. Forest**

**ORIENTADOR: Prof. Mauro Marcelo Mattos Dr.**



# Roteiro

---

- Introdução
- Objetivos
- Fundamentação teórica;
- Trabalhos Correlatos;
- Desenvolvimento do trabalho;
- Operacionalidade;
- Resultados e discussão;
- Conclusão;
- Extensões;



# Introdução

---

- crescimento do uso da orientação a objetos nas fábricas de software;
- dificuldade na troca de paradigma, aprender a pensar OO;
- proposta apresentada por Gubler (2002);



## Objetivos

---

- desenvolver um protótipo nos moldes do trabalho desenvolvido em Gubler (2002) que conduza o aluno no processo de desenvolvimento de uma simples aplicação OO;
- produzir um código fonte em JAVA que corresponderá as intenções do aluno;



# A Orientação a Objetos

---

- “um modo natural de pensar sobre o mundo e de escrever programas de computador” (Deitel e Deitel, 2003);
- objetos possuem seus atributos e métodos (ou ações, mensagens);
- vantagens sobre a análise estruturada;
- Encapsulamento e herança;



# Conceitos básicos de OO

---

- Classe;
- Objeto ou instância;
- Atributos;
- Métodos;
- Acesso;
- Herança;
- Associação;



# Informática na Educação

---

- vantagens no ensino utilizando um sistema de tutoria;
- no Brasil a primeira iniciativa na área de informática na educação aconteceu em 1981 em Brasília, no I Seminário de Informática na Educação (Campos, 1994);
- utilização da tecnologia para transpor conceitos estudados para estudos de casos reais;
- aulas mais motivadoras e interativas;
- o aluno, através da interação com a máquina tem a oportunidade de construir o seu próprio conhecimento (Silva C., 2000);



# Classificação (Sistemas tutores)

---

- **Sistemas Tutores:** apresentação de informações e proposta de atividades para se adquirir conhecimento sobre um assunto (Campos, 1994);
- **Sistemas Tutores Inteligentes:** a tentativa de integrar técnicas de inteligência artificial (IA) e uma teoria de aquisição de conhecimento dentro de um plano de ensino (Campos et al, 1994);





## Trabalho de Mattos (2000)

---

- Protótipo de um sistema que formula perguntas ao aluno;
- conduz o aluno no processo de desenvolvimento do algoritmo desejado;



## Extensão de Gubler (2002)

---

- melhora da interface com o usuário;
- suporte as estruturas de repetição em algoritmos;
- reformulação das perguntas aplicadas em Mattos (2000);



# Desenvolvimento do Trabalho

---

- o objetivo maior desse projeto é fazer uma extensão aos trabalhos de Mattos (2000) e Gubler (2002);
- implementar a extensão da parte de orientação a objetos;



# Requisitos

---

- manter as características funcionais do protótipo desenvolvido no trabalho de Gubler (2002);
- o protótipo deverá permitir a geração de código fonte JAVA para uma determinada área de aplicação;
- implementar a ferramenta utilizando a linguagem de programação JAVA;
- conduzir o aluno no processo de desenvolvimento de aplicação OO;

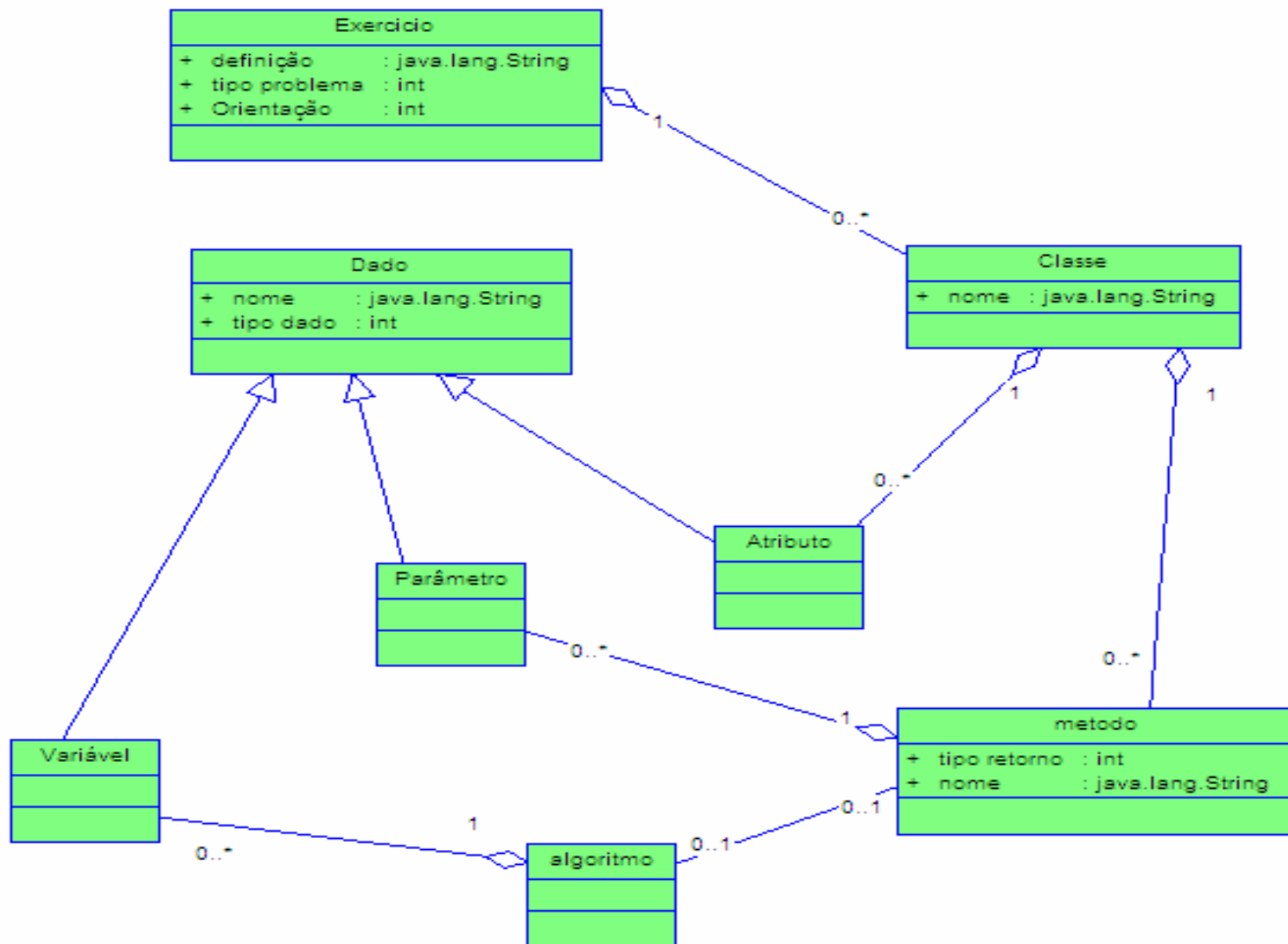


# Especificação

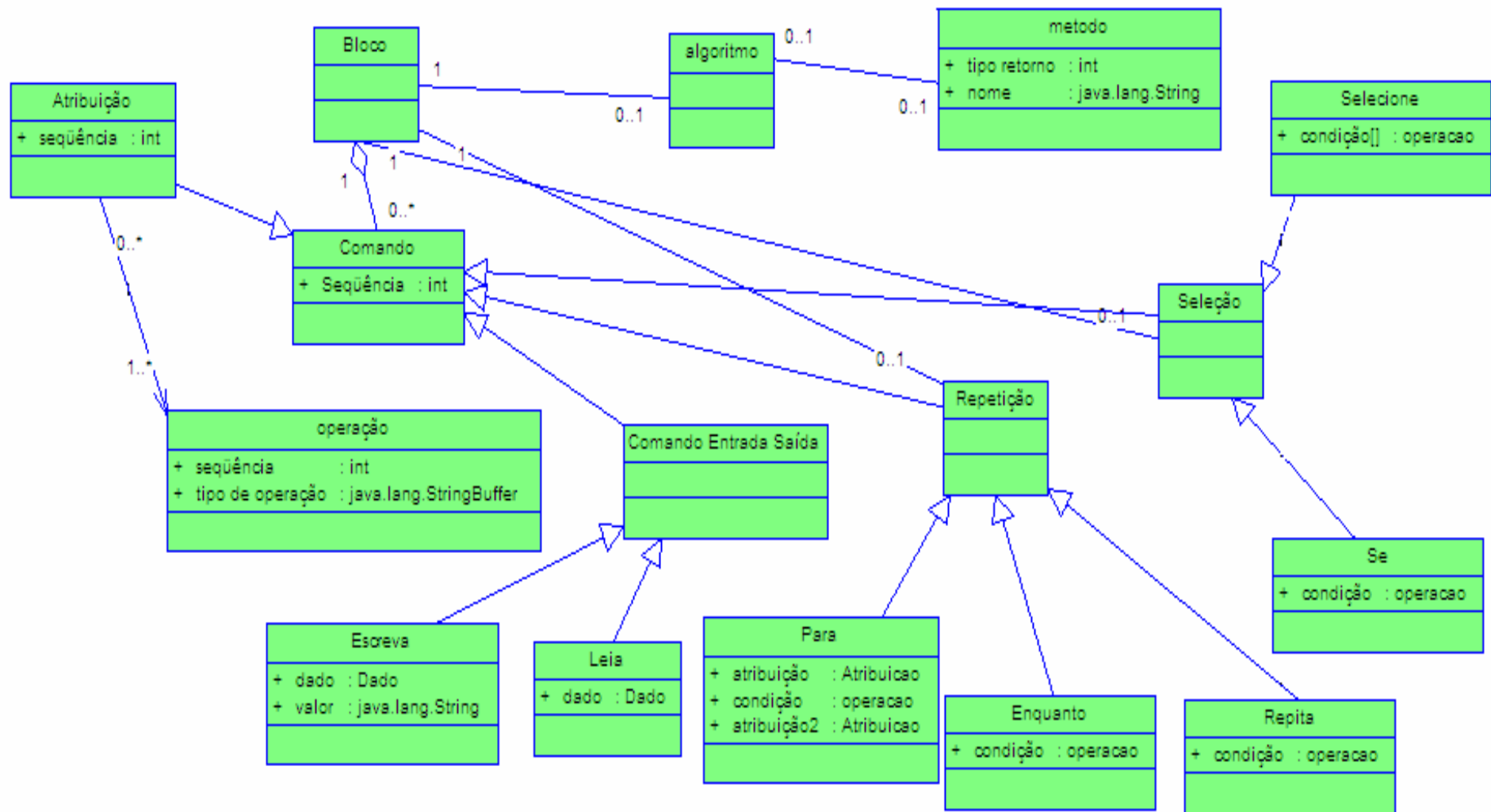
---

- um exercício deve ser cadastrado pelo professor;
- aluno deve realizar o exercício gerando código fonte;
- professor avalia código fonte;
- 2 protótipos:
  - Módulo professor: cadastro exercício;
  - Módulo aluno: realização do exercício;

# Diagrama de classes inicial



# Diagrama de classes - Algoritmo





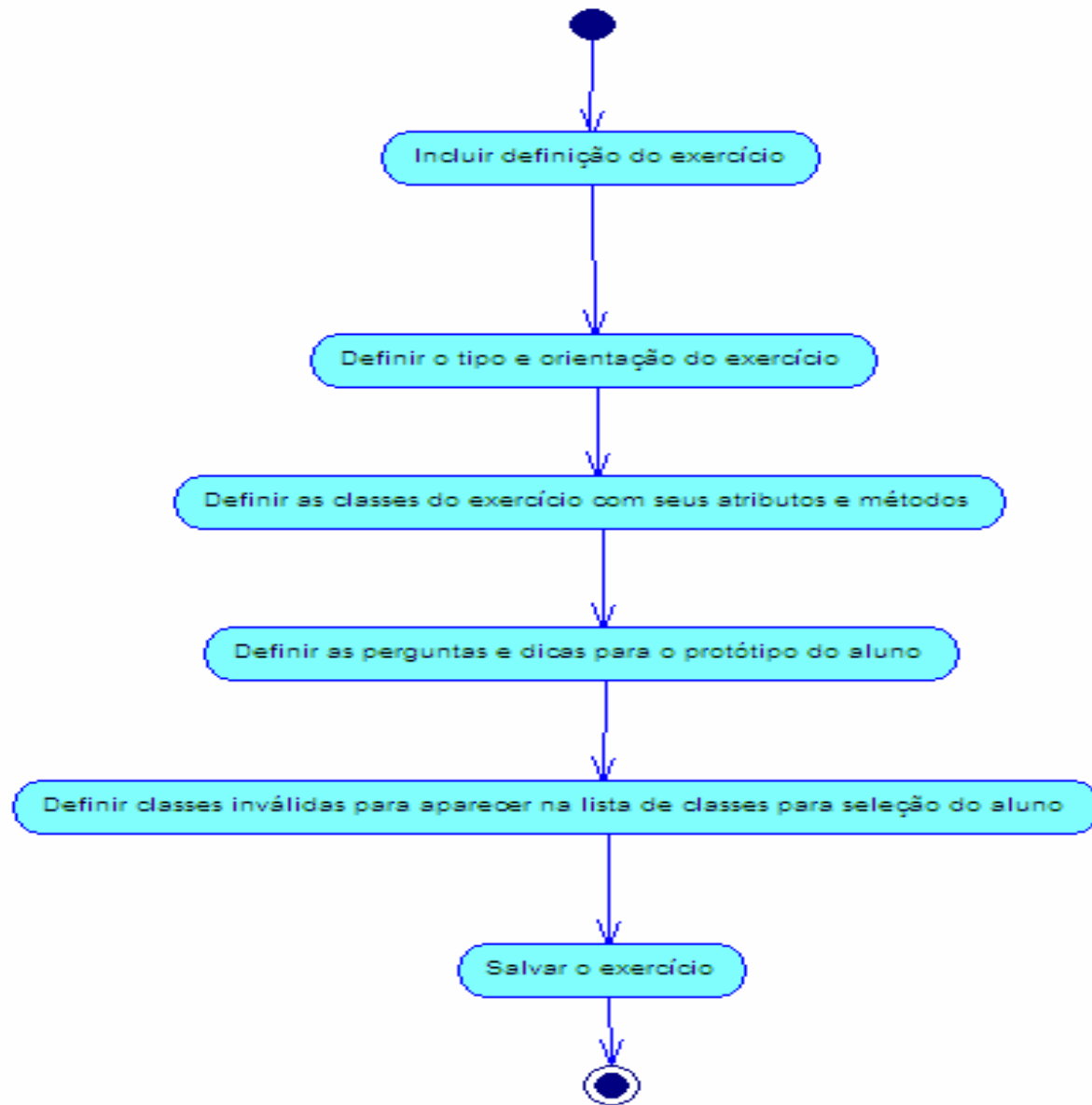
## Módulo professor

---

- permitir o cadastramento de um exercício que mais tarde ser resolvido pelo aluno;
- definir as classes que deverão ser identificadas pelo aluno;



# Módulo professor – diagrama de atividades



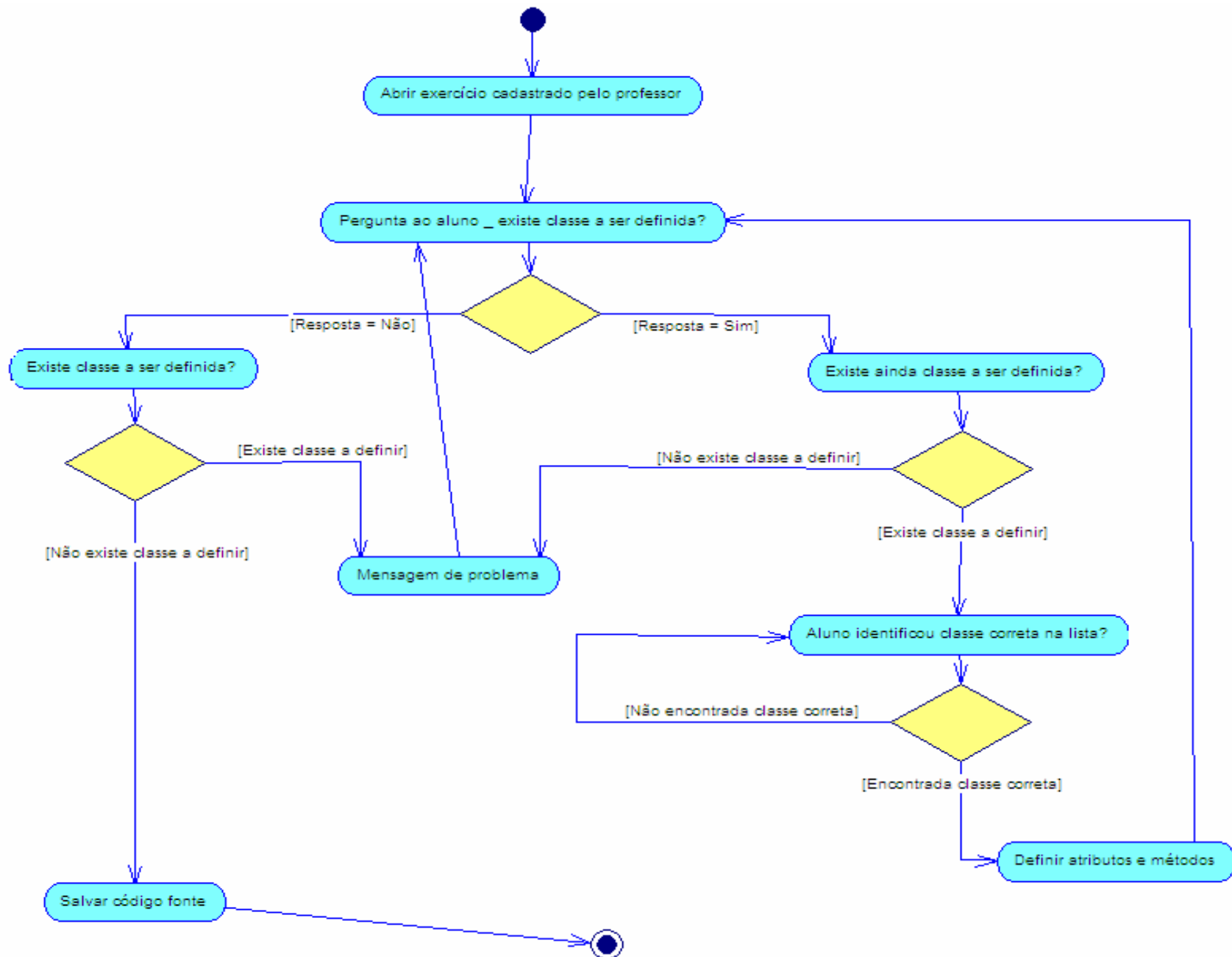


## Módulo do aluno

---

- realização de um exercício previamente cadastrado pelo professor;
- interação com o aluno;
- identificação das classes definidas pelo professor;
- detalhamento de métodos;

# Módulo do aluno – Diagrama de atividades





# Implementação

---

- ambiente de desenvolvimento: Eclipse – Versão 3.0, disponível no CD;
- Desenvolvimento em JAVA utilizando o Swing;
- Interface: necessário à criação de muitos painéis (classe JPanel do Swing);

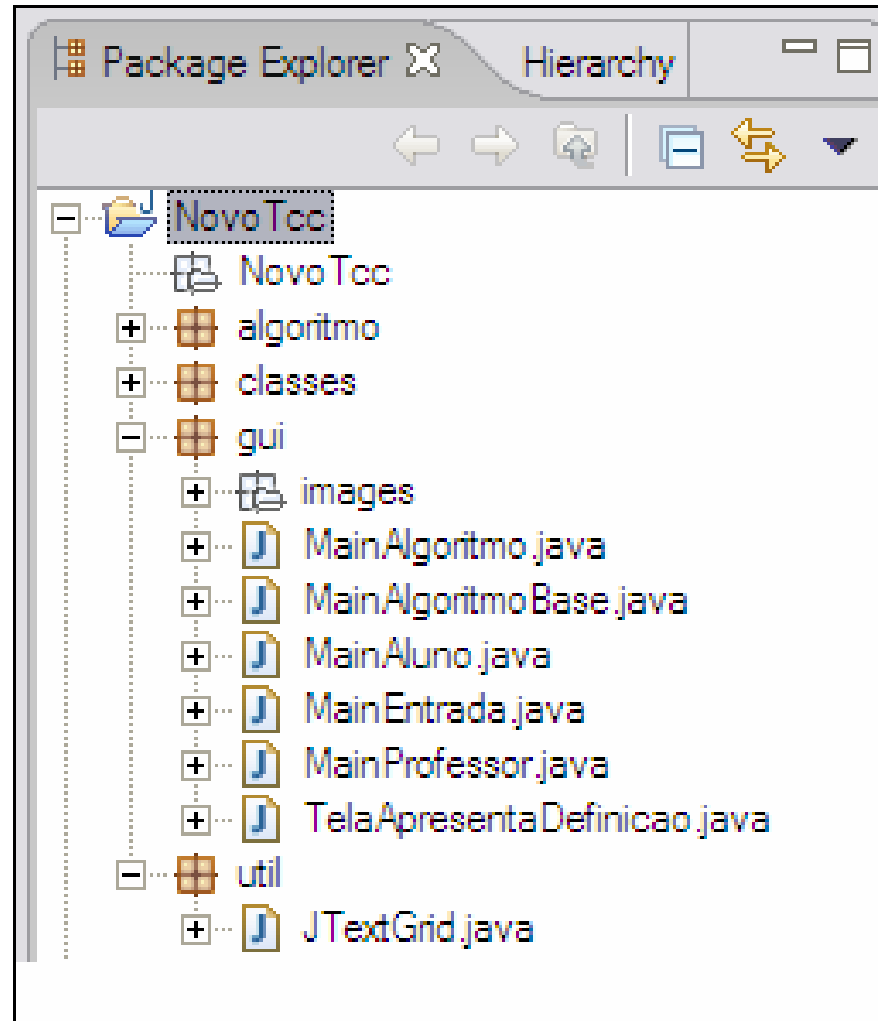


# Estruturação

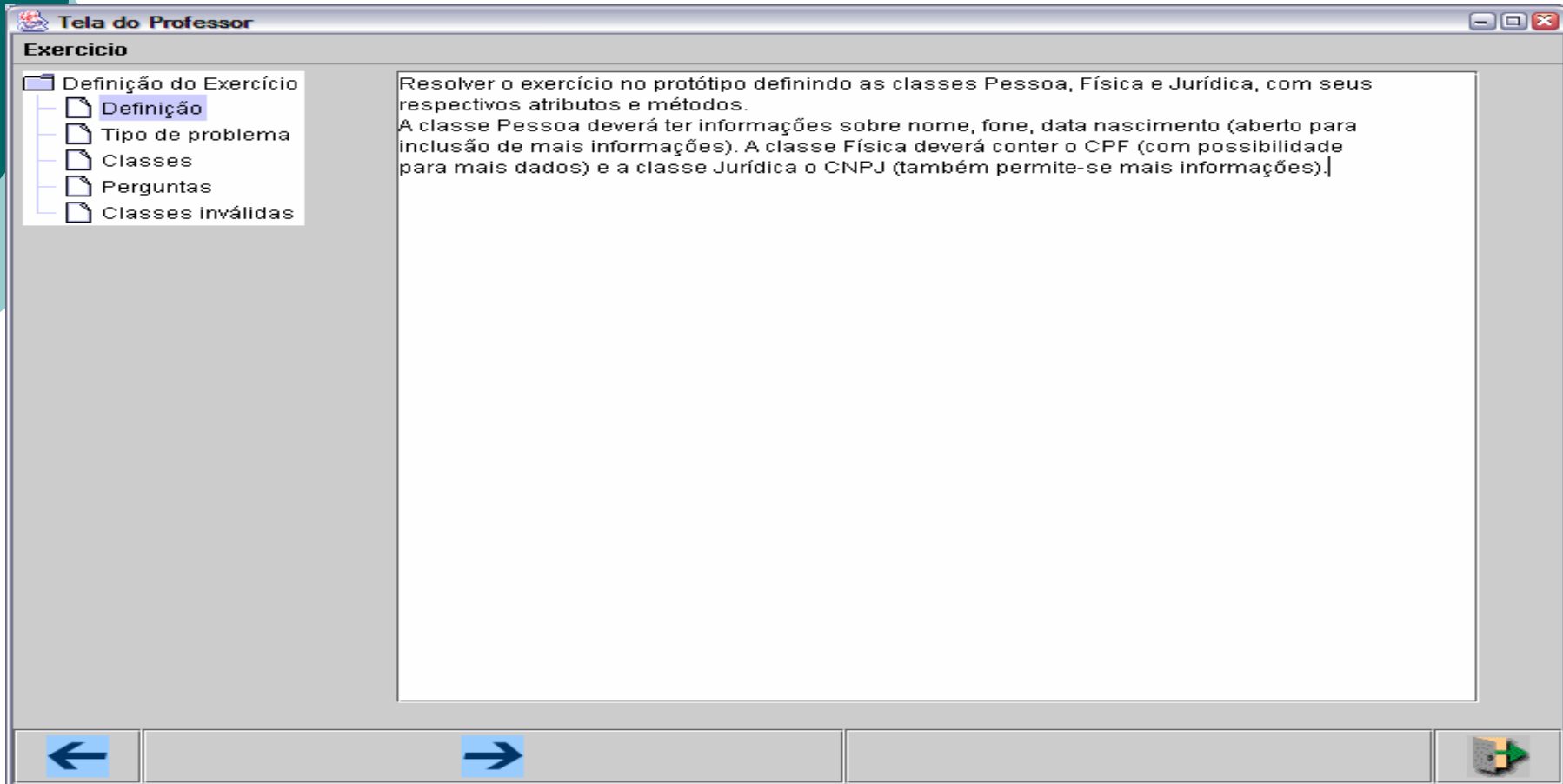
---

- criados os pacotes (packages) **algoritmo**, **classes**, **gui** e **util**;
- pacote **algoritmo** contém as classes referente a algoritmos;
- pacote **classes** contém as classes referente a parte OO;
- pacote **gui** contém as classes referente a interface com usuário;
- pacote **util** contém classes gerais que precisaram ser implementadas;

# Estrutura



# Operacionalidade



**Tela inicial, módulo do professor.**

# Operacionalidade Módulo professor

The screenshot shows a software window titled "Tela do Professor". On the left is a tree view under "Exercicio" with items: "Definição do Exercício", "Definição", "Tipo de problema", "Classes", "Perguntas" (highlighted), and "Classes inválidas". The main area is divided into three sections:

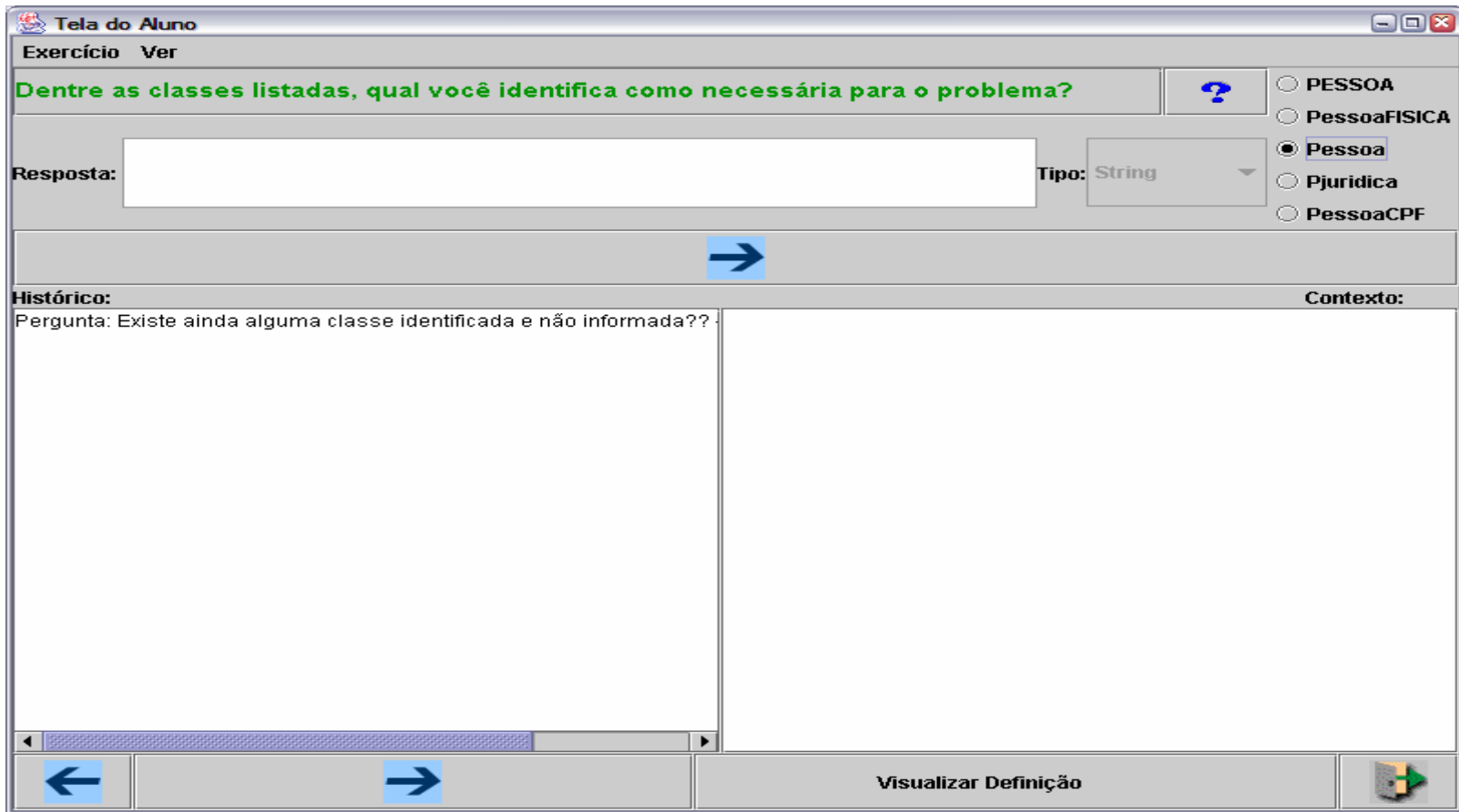
- Section 1:** "Pergunta para Classes:" with a text input field containing "Dentre as classes listadas, qual você identifica como necessária para o problema?". Below it, "Ajuda para as classes" reads: "Reveja definição, definir as classes Pessoa, Fisica e Juridica de acordo com o padrão passado."
- Section 2:** "Selecione a classe:" with a dropdown menu showing "Pessoa". Below it, "Pergunta para atributos de classe:" has a text input field with "Identifique atributo para a classe Pessoa:". "Ajuda para os atributos dessa classe:" reads: "As informações que deverão ser armazenadas são: Nome, fone e data nascimento."
- Section 3:** "Pergunta para métodos de classe:" with a text input field with "Identifique método para a classe Pessoa:". "Ajuda para os métodos dessa classe:" reads: "Métodos get e set somente."

At the bottom, there are navigation buttons: a left arrow, a right arrow, and a green arrow pointing right.

**Definindo perguntas, módulo do professor.**



# Operacionalidade módulo aluno



The screenshot shows a software window titled "Tela do Aluno". At the top, there is a tab labeled "Exercício Ver". Below the tab, a green question is displayed: "Dentre as classes listadas, qual você identifica como necessária para o problema?". To the right of the question is a blue question mark icon. Below the question is a text input field labeled "Resposta:". To the right of the input field is a dropdown menu labeled "Tipo:" with "String" selected. To the right of the dropdown menu is a list of radio buttons with the following labels: "PESSOA", "PessoaFISICA", "Pessoa", "Pjuridica", and "PessoaCPF". The "Pessoa" option is selected. Below the input field and dropdown menu is a blue arrow button pointing right. Below this is a section labeled "Histórico:" on the left and "Contexto:" on the right. The "Histórico:" section contains the text "Pergunta: Existe ainda alguma classe identificada e não informada??". At the bottom of the window, there are two blue arrow buttons (left and right) and a button labeled "Visualizar Definição" with a small icon to its right.

**Realizando exercício cadastrado, módulo do aluno.**

# Operacionalidade módulo aluno

The screenshot shows a window titled "Tela Aluno - Definindo Classe Pessoa". The interface is divided into several sections:

- Exercício Ver**: A header section.
- Identifique atributo para a classe Pessoa:** A green instruction text.
- Resposta:** A text input field containing "nome".
- Tipo Atributo:** A dropdown menu currently set to "String".
- Contexto:** A list of radio buttons with options:  Sim,  Não,  Pessoa,  Pjuridica, and  PessoaCPF.
- Histórico:** A scrollable area containing a list of questions:
  - Pergunta: Existe ainda alguma classe identificada e não informada??
  - Pergunta: Dentre as classes listadas, qual você identifica como neces
  - Pergunta: Existe ainda algum atributo identificado e não informado par
- Contexto:** A scrollable area containing the text "Pessoa".
- Navigation:** A central blue arrow button and a bottom bar with left and right arrow buttons.
- Buttons:** A "Visualizar Definição" button and a small icon button in the bottom right corner.

**Identificando novo atributo, módulo do aluno.**

# Operacionalidade módulo aluno

The screenshot shows a window titled "Tela Aluno - Definindo Classe Pessoa". At the top, there is a tab labeled "Exercício Ver". Below this, the instruction "Identifique método para a classe Pessoa:" is displayed. A text input field contains the response "setNome", and a dropdown menu for "Tipo Retorno:" is set to "void". To the right, there are radio buttons for "Sim" (selected), "Não", "Pessoa", "Pjuridica", and "PessoaCPF". A blue arrow button is positioned below the input fields. The "Histórico:" section on the left contains a list of questions and answers. The "Contexto:" section on the right shows the class definition for "Pessoa" with attributes "nome", "fone", and "dataNascimento". At the bottom, there are navigation arrows and a "Visualizar Definição" button.

**Exercício Ver**

Identifique método para a classe Pessoa:  

Resposta:  Tipo Retorno:

Sim  
 Não  
 Pessoa  
 Pjuridica  
 PessoaCPF

**Histórico:**

Pergunta: Existe ainda alguma classe identificada e não informada??  
Pergunta: Dentre as classes listadas, qual você identifica como neces  
Pergunta: Existe ainda algum atributo identificado e não informado par  
Pergunta: Identifique atributo para a classe Pessoa: - sua resposta -->  
Pergunta: Existe ainda algum atributo identificado e não informado par  
Pergunta: Identifique atributo para a classe Pessoa: - sua resposta -->  
Pergunta: Existe ainda algum atributo identificado e não informado par  
Pergunta: Identifique atributo para a classe Pessoa: - sua resposta -->  
Pergunta: Existe ainda algum método identificado e não informado par

**Contexto:**

```
Pessoa
  ATRIBUTOS
    nome
    fone
    dataNascimento
```

**Identificando novo método, módulo do aluno.**

# Operacionalidade módulo aluno

Tela do Aluno - Definindo Método setName da classe Pessoa

Exercício Ver

Neste ponto, tem algum retorno que pode ser feito??  Sim  Não

Resposta:  Tipo: String

→ Fim contexto(Return)

Método: Contexto:

```
/**
 *
 *@author null
 *Comente aqui o seu método
 */
this.nome=pnome;
```

```
Classe Pessoa
Definindo Método setName

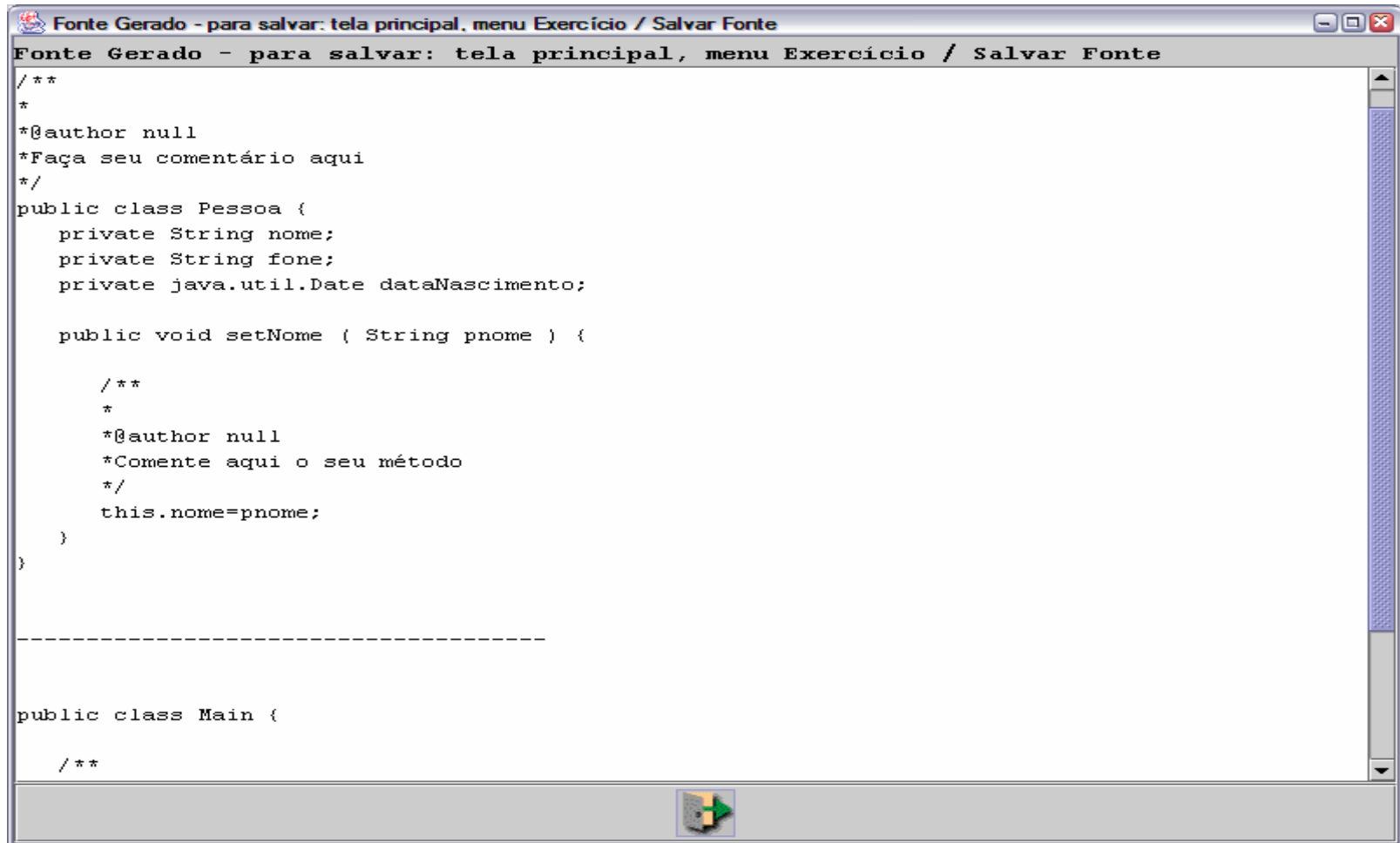
Parametros:
-----
    pnome
-----

Bloco Principal <<<<< Você está aqui
```

← →

**Detalhando método, módulo do aluno.**

# Operacionalidade módulo aluno



```
Fonte Gerado - para salvar: tela principal, menu Exercício / Salvar Fonte
Fonte Gerado - para salvar: tela principal, menu Exercício / Salvar Fonte
/**
 *
 *@author null
 *Faça seu comentário aqui
 */
public class Pessoa {
    private String nome;
    private String fone;
    private java.util.Date dataNascimento;

    public void setNome ( String pnome ) {

        /**
         *
         *@author null
         *Comente aqui o seu método
         */
        this.nome=pnome;
    }
}

-----

public class Main {

    /**
```

**Visualizando fonte gerado, módulo do aluno.**



## Resultados e discussão

---

- atingido o objetivo de estender o trabalho de Gubler (2002), e contemplar a implementação de um simples problema de OO;
- novidade: tela de definição de exercício (módulo do professor);
- geração de código fonte em JAVA;
- não contempla a geração de diagramas;
- não abrange muitos conceitos importantes de OO;



## Conclusões

---

- auxílio do aluno em disciplinas de difícil compreensão que são as disciplinas de introdução a programação e programação II;
- objetivos atingidos com algumas limitações já citadas;
- algumas outras limitações como a impossibilidade de declaração de um objeto do tipo de uma classe;



# Extensões

---

- geração de diagramas;
- implementação das funcionalidades referente a dificuldade do problema;
- implementação de outros tipos de comandos possíveis no JAVA;
- a possibilidade de se definir tipos como tipos de classes ;
- melhorar esquema dos comentários;
- extensões de código fonte para outras linguagens;
- contemplar demais conceitos de OO;





# Demonstração do Sistema

---

Questões?