

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

**Protótipo de um ambiente de
monitoramento e representação
de programas Java utilizando
Reflexão Computacional**

Acadêmico - Romeu Gadotti

Orientador - Prof. Marcel Hugo

Roteiro de Apresentação

- Introdução
- Orientação a Objetos em Java
- Reflexão Computacional
- Reflexão Computacional em Java
- Desenvolvimento do Trabalho
- Conclusões
- Referências Bibliográficas

Introdução

- Orientação a Objetos
- Java e linguagens orientadas a objetos
- O protótipo proposto
- Reflexão Computacional

Objetivos

- desenvolver o protótipo de um ambiente que apresente classes e monitore objetos
 - aprofundar os conceitos da tecnologia de orientação a objetos em Java
 - analisar as classes do pacote *java.lang.reflect* e suas extensões
 - pesquisar e aplicar a tecnologia de reflexão computacional

Orientação a Objetos em Java

- Encapsulamento
- Herança
- Polimorfismo

Orientação a Objetos em Java

- Criação de classes
- Declaração dos membros da classe
- Criação e destruição de objetos
- Troca de mensagens entre objetos
- Subclasses e herança
- Polimorfismo e sobrecarga
- Interfaces

Orientação a Objetos em Java

```
class Inter
{
    public static void main(String[] args) {
        ContagemRegressiva cr = new ContagemRegressiva();
        cr.setValorInicial(10);
        System.out.println("Setou o contador com 10");
        while (cr.getValor() > 0) {
            cr.decrementa();
            System.out.println("Valor atual: " + cr.getValor());
        }
    }
}
```

Orientação a Objetos em Java

```
public class Contador {  
    protected long valor;  
    public void setValorInicial(long v) { valor = v; }  
    public long getValor() { return valor; }  
    public void incrementa() { valor++; }  
}
```

```
public interface Regressivo { public void decrementa(); }
```

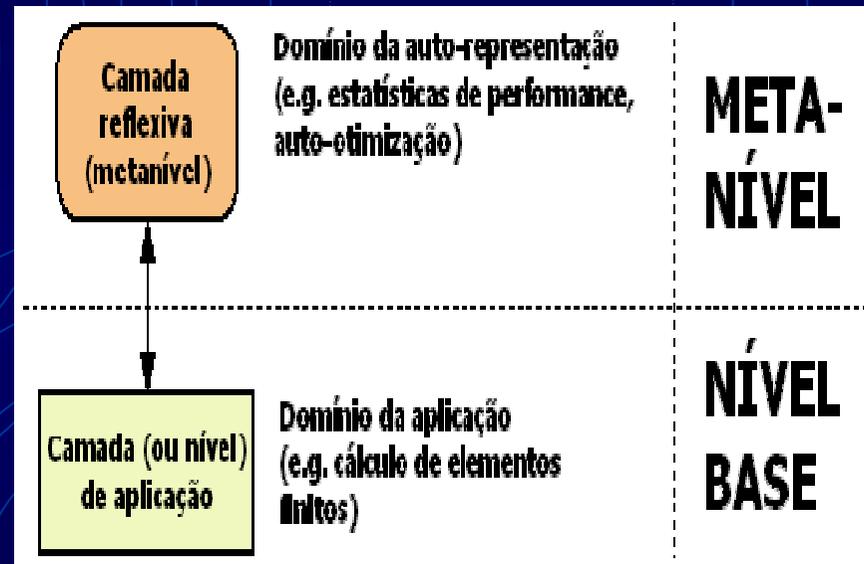
```
public class ContagemRegressiva extends Contador implements  
Regressivo {  
    public void decrementa() { super.valor--; }  
    public void incrementa(short n) { super.valor += n; }  
}
```

Reflexão Computacional

- Origem
- Conceitos
 - Estudo da autoconsciência (IA)
 - Divisão de interesses (Engenharia de Software)
 - Refletir e ajustar-se (Kleinoder, 2000)
 - Introspecção e redirecionamento (Senra, 2001)

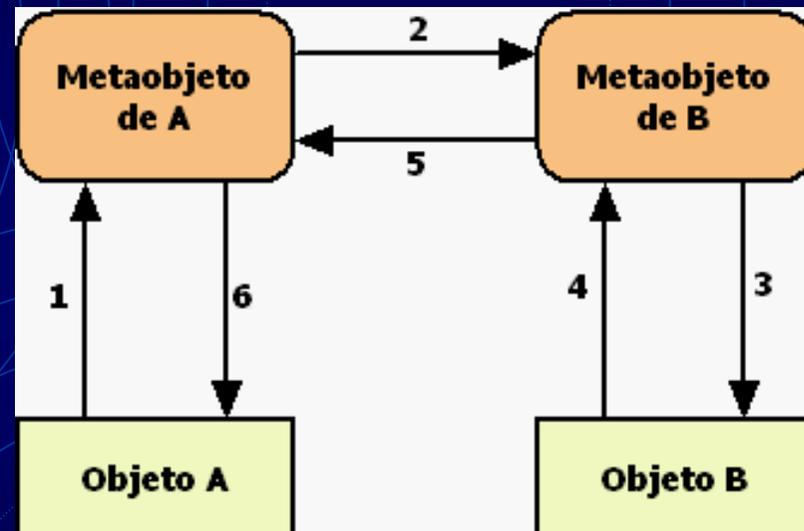
Reflexão Computacional

- Domínio da aplicação
- Meta-domínio
- Nível base
- Metaníveis
- Arquitetura de metaníveis

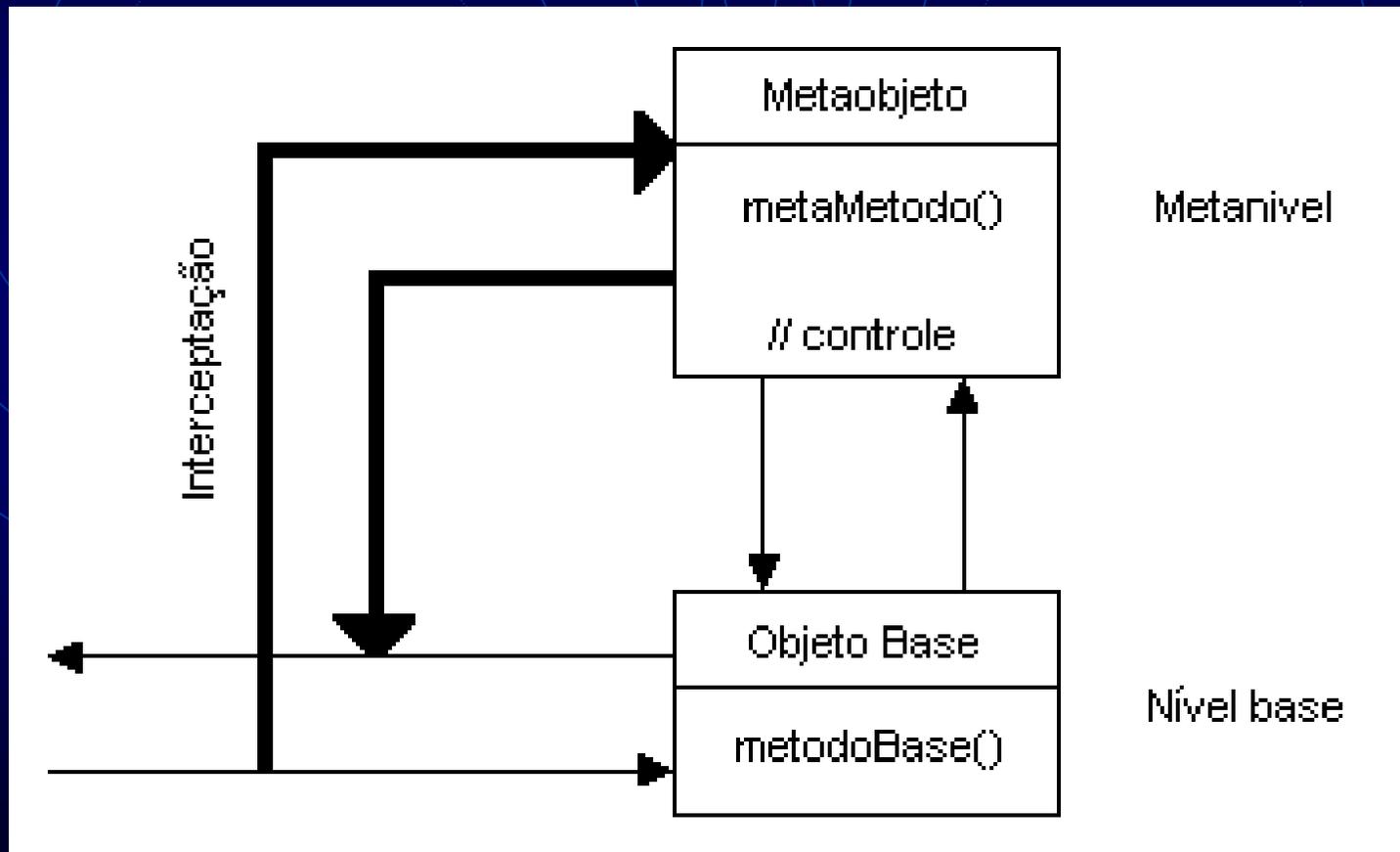


Reflexão em Orientação a Objetos

- Vantagens da união OO e RC
- Metaobjetos
- Eventuais problemas
- Aspectos funcionais e não funcionais



Reflexão em Orientação a Objetos



Metaobject Protocol

- Conexão entre níveis
- Objetivos
 - Simplicidade
 - Poder de programação
 - Compatibilidade
 - Desempenho

Reflexão Computacional em Java

- *java.lang.class* e *java.lang.reflect*
 - Java Reflection API
 - Representação dos tipos de dados da Java
 - Capacidade de introspecção
 - Não provê capacidade reflexiva completa

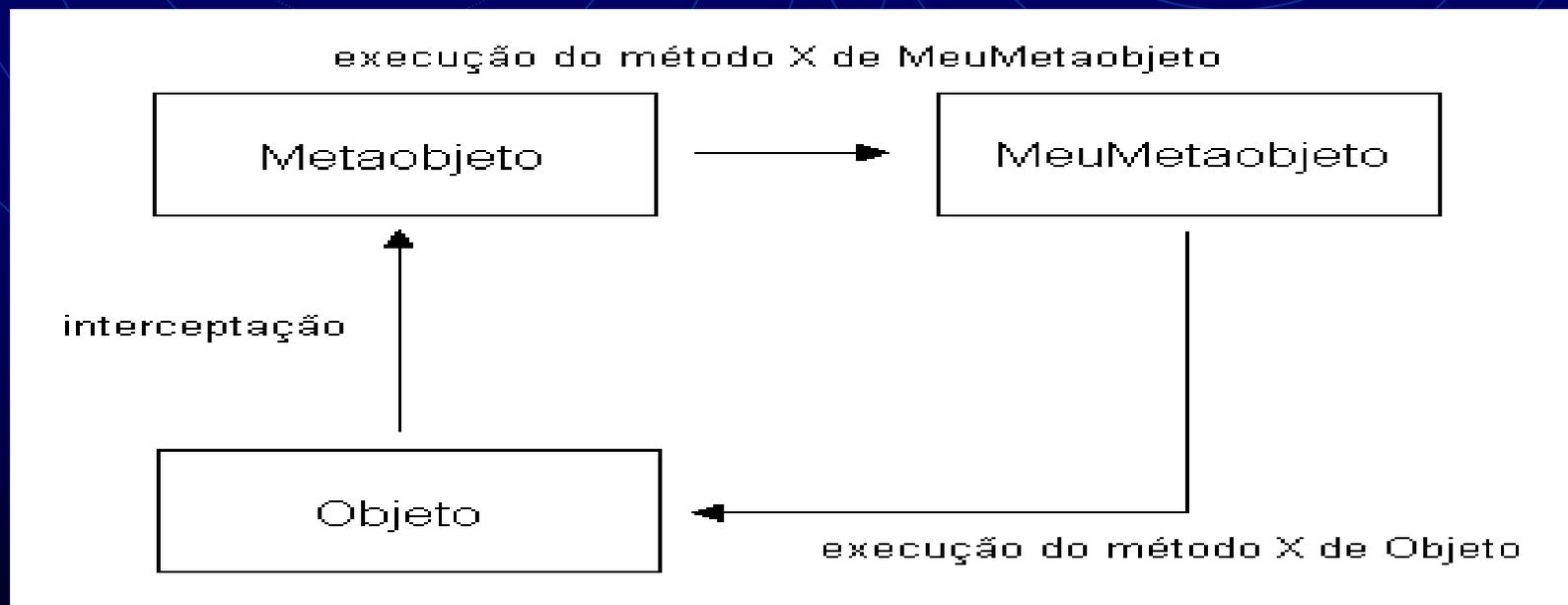
Reflexão Computacional em Java

```
import java.lang.reflect.*;

public class Collect
{
    public static void main(String args[])
    {
        try
        {
            Class c = Class.forName(args[0]);
            Method m[] = c.getDeclaredMethods();
            for (int i = 0; i < m.length; i++)
                System.out.println(m[i].toString());
        }
        catch (Throwable e)
        {
            System.err.println(e);
        }
    }
}
```

Reflexão Computacional em Java

- *Class Loader*
- *Extensões*



Reflexão Computacional em Java

- Extensões
 - Javassist
 - Sistema reflexivo em tempo de execução
 - Possibilita introspecção e interceptação
 - Na interceptação o sistema de *runtime* chama um método associado
 - As principais classes são CtClass e ClassPool

Reflexão Computacional em Java

- Extensões
 - Javassist
 - O metaobjeto mantém métodos de interceptação

```
public class MyMetaobject extends Metaobject
{
    public Object trapMethodcall(int identifier, Object[] args)
    {
        // código executado antes do código do nível base
    }

    public Object trapFieldRead(String fieldName)
    {
        // código executado antes do código do nível base
    }

    public void trapFieldWrite(String fieldName, Object value)
    {
        // código executado antes do código do nível base
    }
}
```

Reflexão Computacional em Java

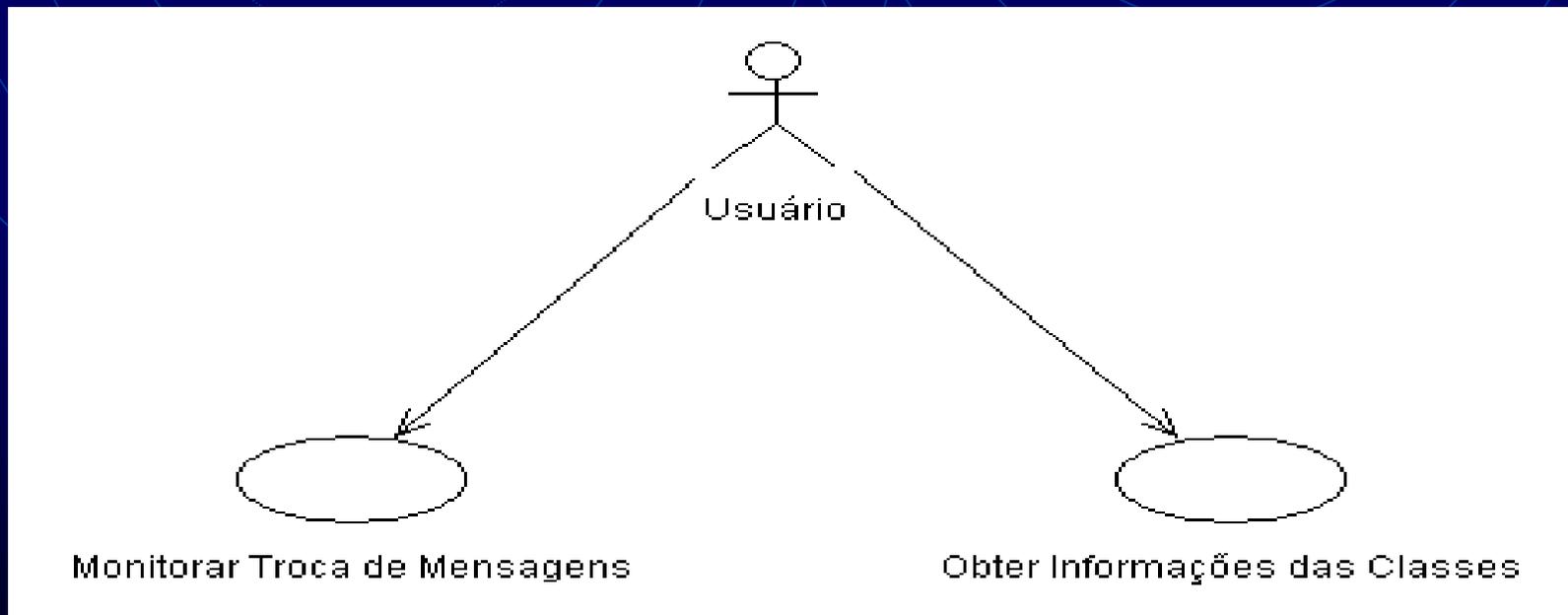
- Extensões
 - Guaraná

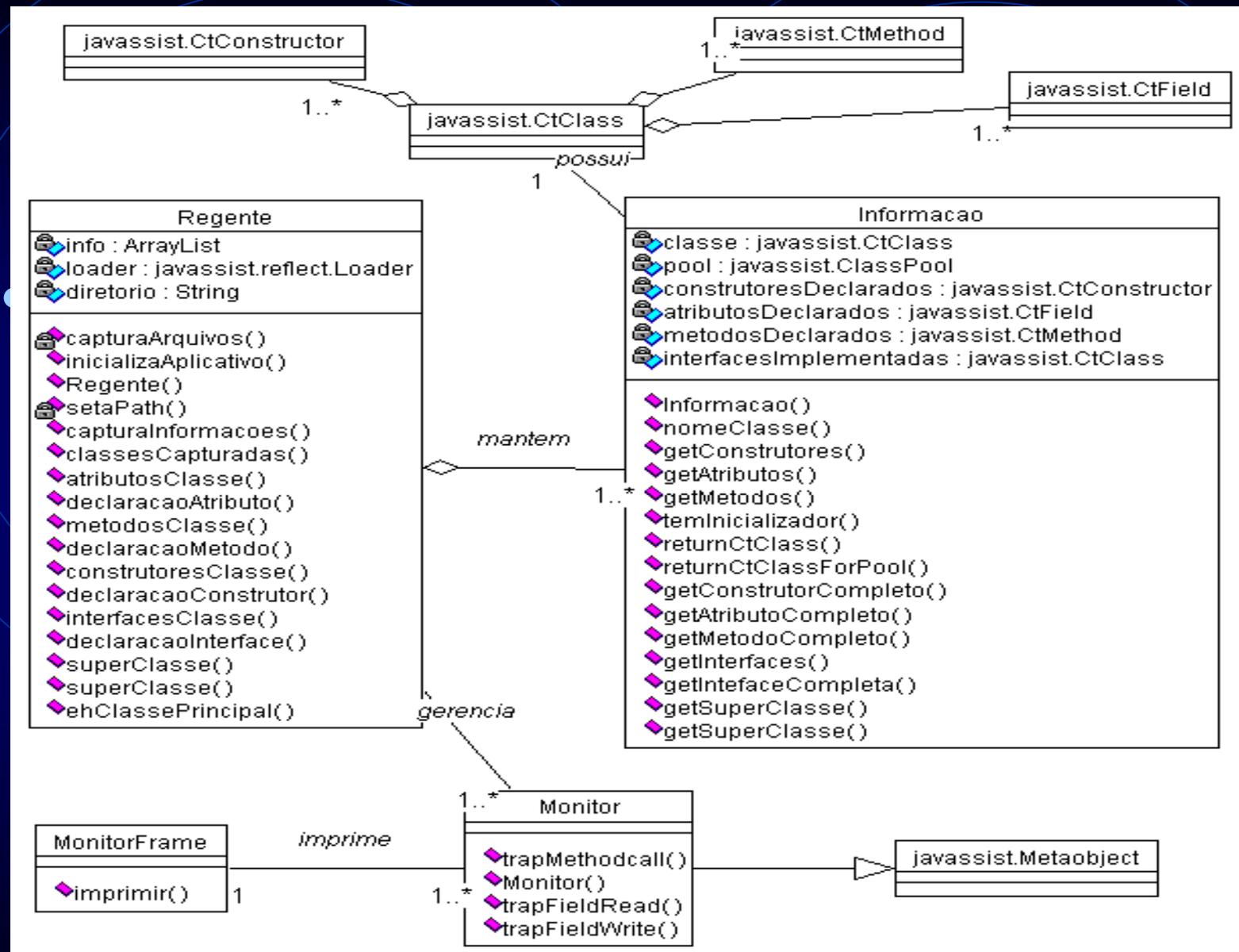
Desenvolvimento do Trabalho

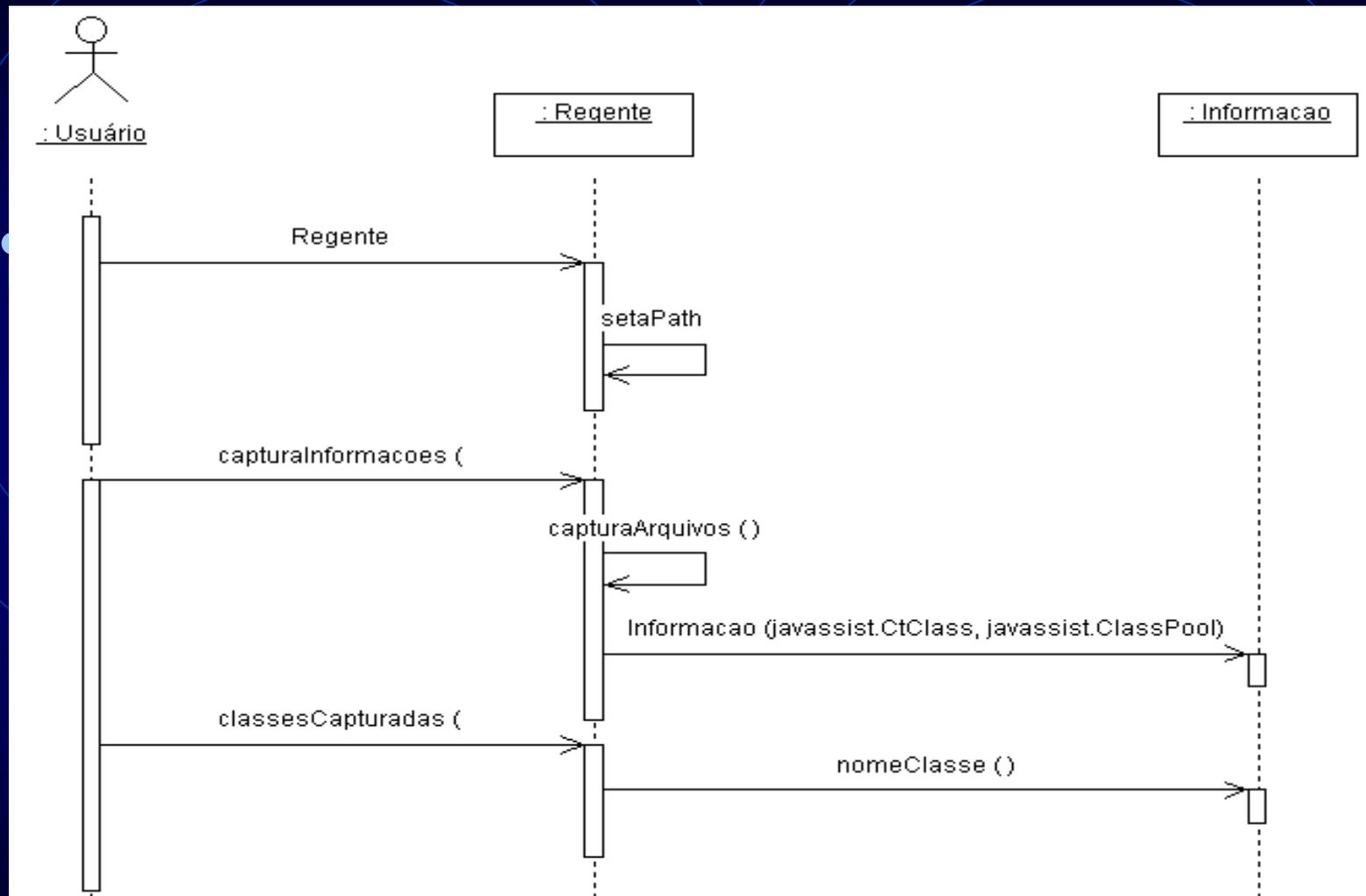
- Especificação
 - UML (*Unified Modeling Language*)
 - Ferramenta Rational Rose
- Implementação
 - Ambiente Borland JBuilder 7 Enterprise Trial
 - Linguagem Java
- Biblioteca
 - Javassist 2.0

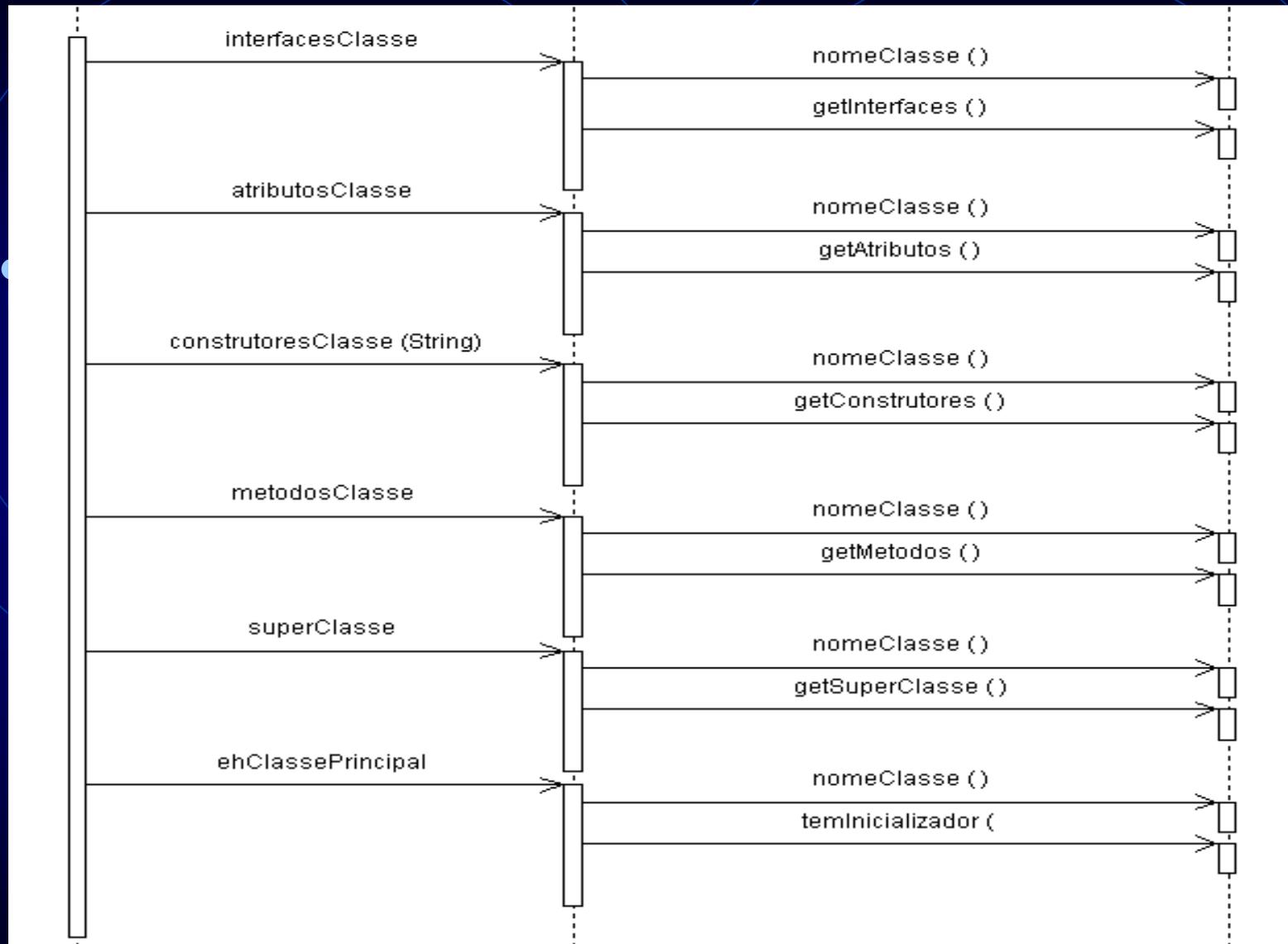
Desenvolvimento do Trabalho

- Especificação
 - Diagrama de Casos de Uso



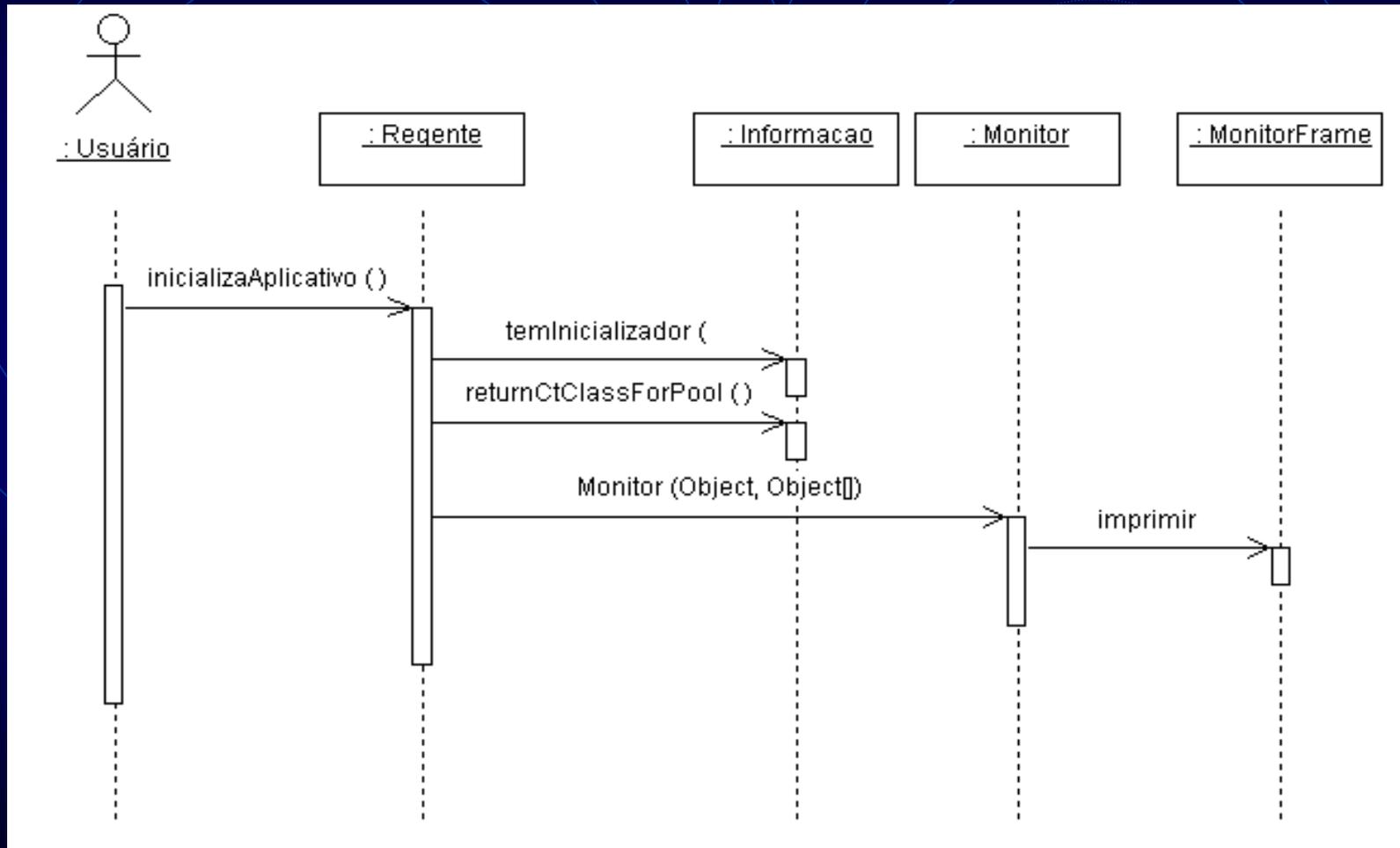




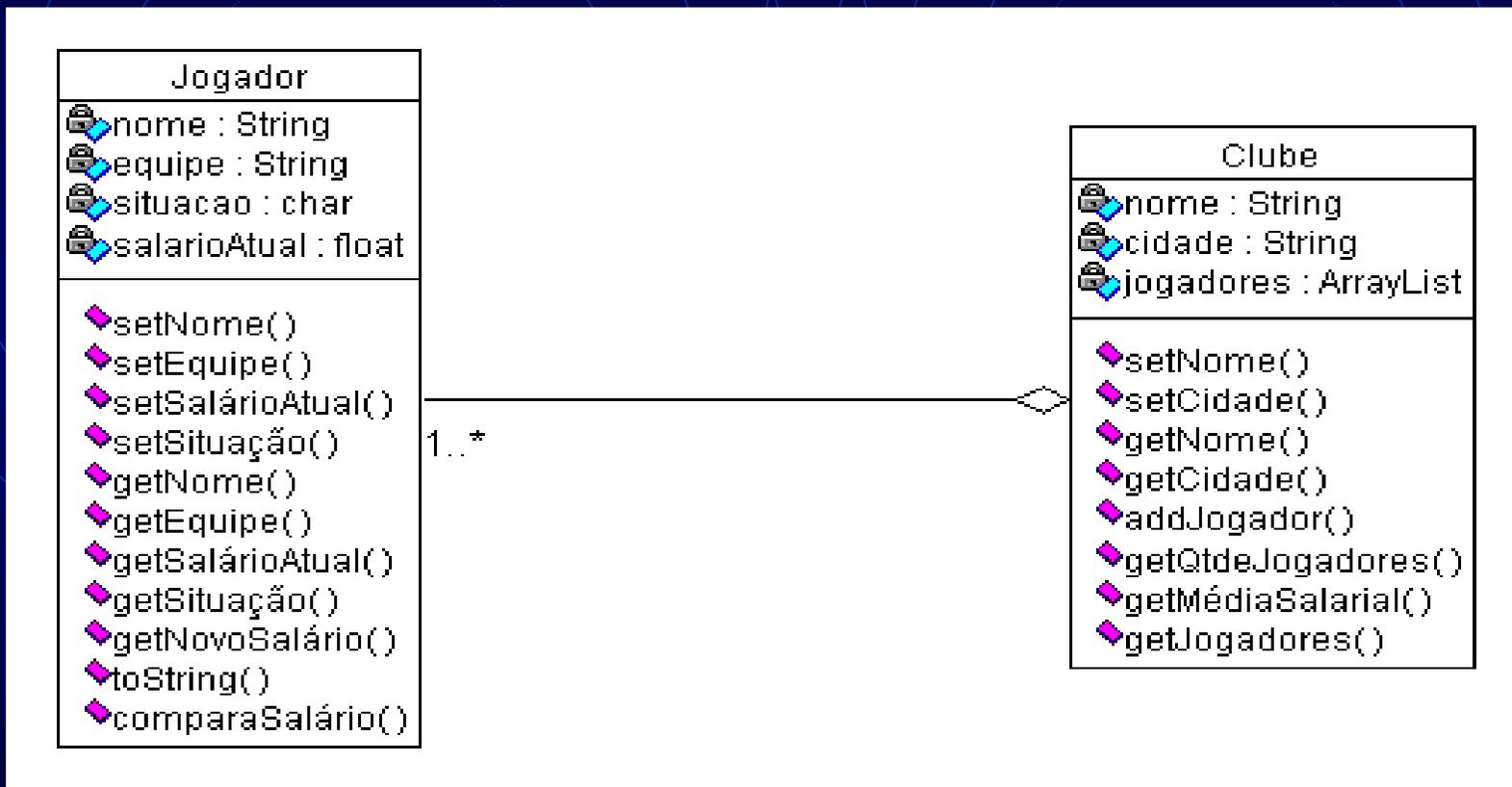




Desenvolvimento do Trabalho



Estudo de Caso



Resultados

- • Inspeção das classes
- • Demonstração dos membros das classes
- • Definição das classes bases e interfaces
- • Indicação da classe principal
- • Inicialização do aplicativo inspecionado
- • Monitoramento das mensagens

Conclusões

- Possibilidade de inspeção na estrutura das classes e monitoramento de objetos
- Reflexão + OO = menos complexidade e facilidade de manutenção
- Resolução de problemas simples através da introspecção

Referências Bibliográficas

DEVEGILI, Augusto Jun. Tutorial sobre reflexão em orientação a objetos. Florianópolis, abr. 2000. Disponível em: <http://www.uvm.edu/~dewey/reflection_manual/>. Acesso em: 23 jun. 2002.

SENRA, Rodrigo Dias Arruda. Programação reflexiva sobre o protocolo de meta-objetos Guaraná, São Paulo, nov. 2001. Disponível em: <<http://www.ic.unicamp.br/~921234/dissert/node5.html>>. Acesso em: 12 maio 2002.

Referências Bibliográficas

LEE, Arthur H.; SHIN, Ho-Yun. **Building a persistent object store using the java reflection API**. Seoul, 1998. Disponível em: <<http://www4.informatik.uni-erlangen.de/Publications/>>. Acesso em: 28 ago. 2002.

KLEINODER, Jurgem; GOLM, Michael. **metaXa and the future of reflection**. Erlangen, Janeiro 2000. Disponível em: <<http://www4.informatik.uni-erlangen.de/Publications/>>. Acesso em: 28 ago. 2002.

Demonstração do Protótipo

Obrigado!

romeu@wk.com.br