Henrique Machado Müller

RunGroovy: Extensão do BlueJ para execução de linhas de código

Orientador:
Prof. Adilson Vahldick

Roteiro de Apresentação

- Introdução
- Fundamentação Teórica
- Desenvolvimento da Ferramenta
- Estudo de caso
- Resultados e Discussões
- Conclusão

Introdução

- Problemas
  - Novos paradigmas de programação
  - Estímulos durante o aprendizado

Introdução

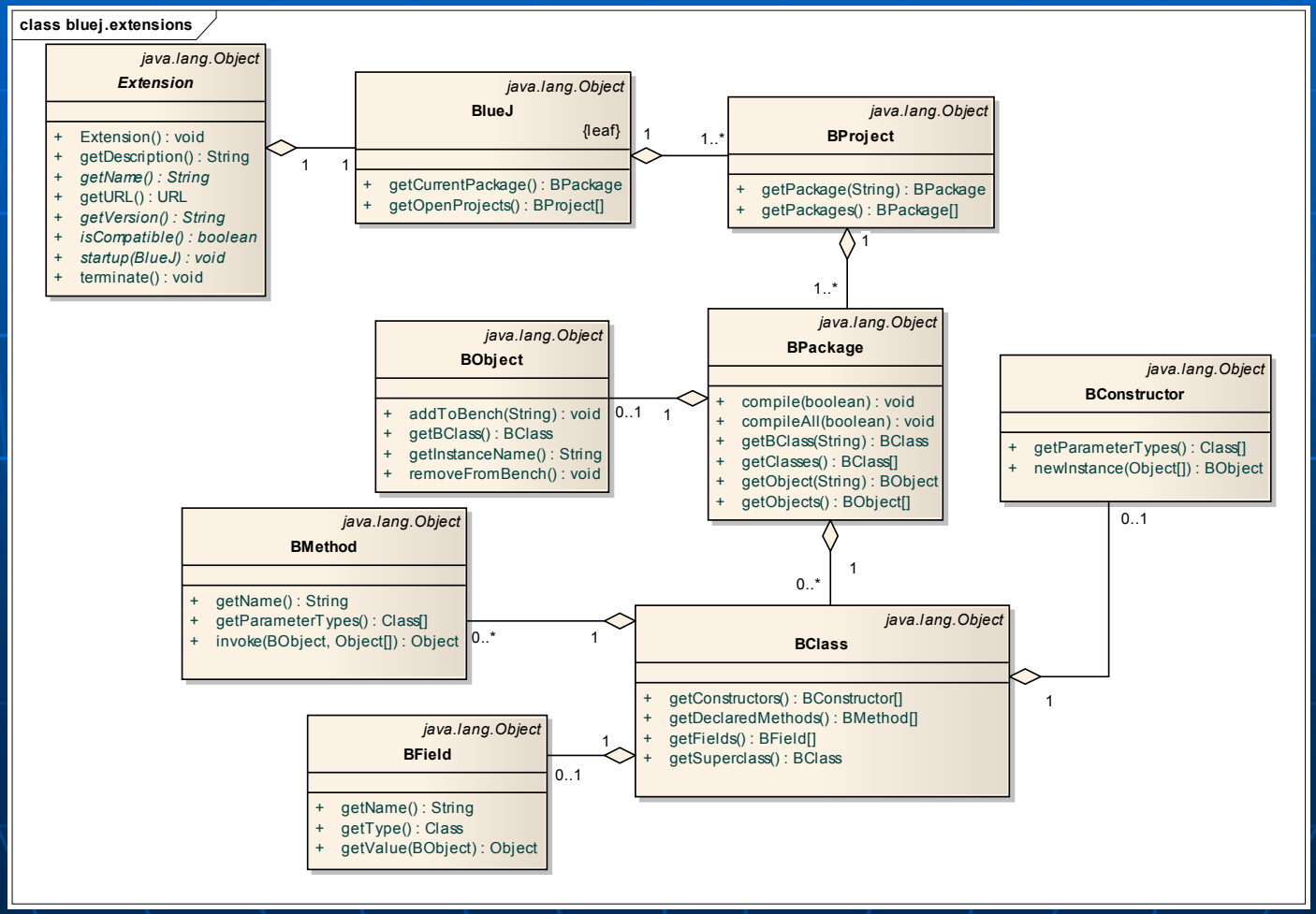
- Objetivos
  - Execução de linhas de códigos
  - Reflexão no ambiente de desenvolvimento

Fundamentação Teórica

- API de extensão do BlueJ
  - Desenvolvimento extensão no ambiente
  - Manipulação dos objetos
  - Criação de menus e preferências

Fundamentação Teórica

API de Extensão do BlueJ



Fundamentação Teórica

- Groovy
  - Compilada ou interpretada
  - Compatível com a JVM

Fundamentação Teórica

Groovy – Hello World

```
def name='World';
println "Hello $name!"

Hello World
```

```
class Greet {
  def name
  Greet(who) {
    name = who[0].toUpperCase() + who[1..-1] }
  def salute() { println "Hello $name!" }
}

def g = new Greet('world')  // create object
g.salute()                  // Output "Hello World!"
```

Fundamentação Teórica

Groovy com integração ao Java

```
private static void exemplo7() throws CompilationFailedException
{
  Date dataDeHoje = new Date();
  Binding binding = new Binding();
  binding.setVariable("hoje",
     SimpleDateFormat.getInstance().format(dataDeHoje));
  GroovyShell shell = new GroovyShell(binding);
  shell.evaluate (
    "def a='variavel definida dentro do script' ; " +
    " println 'Dia de hoje : '+ hoje ");
  Object variavelDoScript = binding.getVariable("a");
  System.out.println(variavelDoScript);
}
```

Desenvolvimento da Ferramenta

- Requisitos principais
- Cenários do caso de uso
- Diagrama de Atividade
- Diagramas de classes
- Diagrama de Seqüência

Desenvolvimento da Ferramenta

- Requisitos principais
  - Implementar uma janela de console para execução de comandos
  - Criar, alterar, modificar estado de objetos na ferramenta BlueJ
  - Salvar, carregar e executar arquivos
  - Linguagem Groovy para execução dos comandos
  - Ser compatível com qualquer sistema operacional que tenha suporte a JVM

Desenvolvimento da Ferramenta

Fluxo Principal {Principal}.

1. O usuário digita um script.
2. O usuário seleciona operação de executar.
3. Sistema executa o(s) comando(s) digitado(s).
3.1 Sistema altera e/ou cria objetos do BlueJ conforme o(s) comando(s) digitado(s).
3.2 Sistema retorna mensagens de cada comando digitado, caso existir.

Salvar arquivo { Alternativo }.

1. No passo 1 do Fluxo Principal o usuário pode salvar o arquivo.
2. O usuário seleciona operação de salvar arquivo.
3. O Sistema mostra tela de pastas do sistema.
4. O usuário seleciona a pasta para salvar o arquivo.
5. O usuário informa o nome do arquivo para salvar.
6. O Sistema grava arquivo no disco.
7. Retorna para o passo 1 do Fluxo Principal.

Cancelamento de operação Salvar Arquivo {Alternativo}.

1. Nos passos 3 e 4 do Salvar arquivo o usuário pode cancelar a operação.

Abrir arquivo {Alternativo}.

1. O usuário seleciona a operação de abrir arquivo.
2. O sistema mostra uma tela de seleção de arquivo.
3. O usuário seleciona o arquivo que deseja abrir.
4. O sistema carrega conteúdo do arquivo na tela.
5. Retorna para o passo 1 do Fluxo Principal.

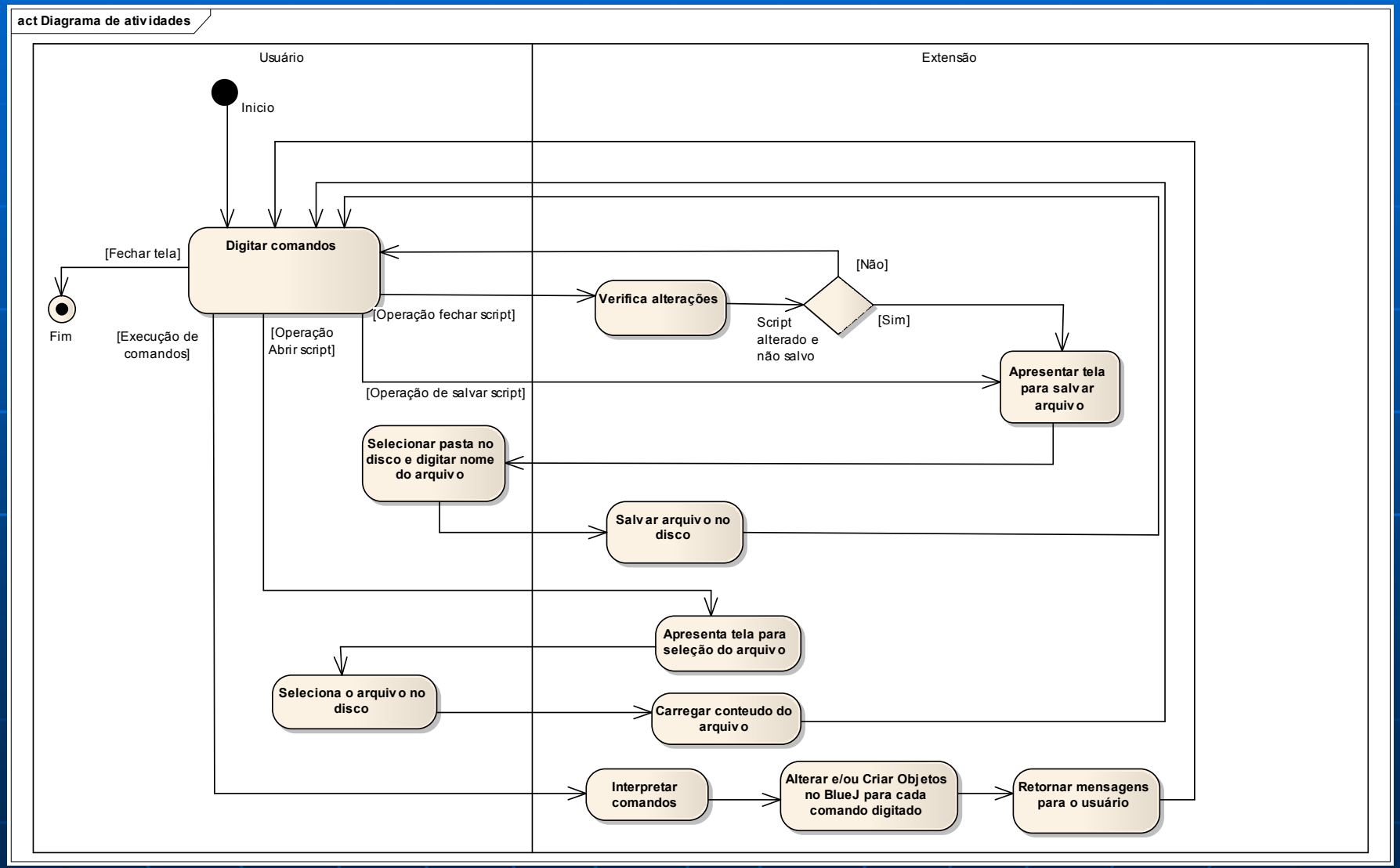
Cancelamento de operação Abrir Arquivo {Alternativo}.

1. No passo 3 do Abrir arquivo o usuário pode cancelar a operação.

Fechar arquivo {Alternativo}.

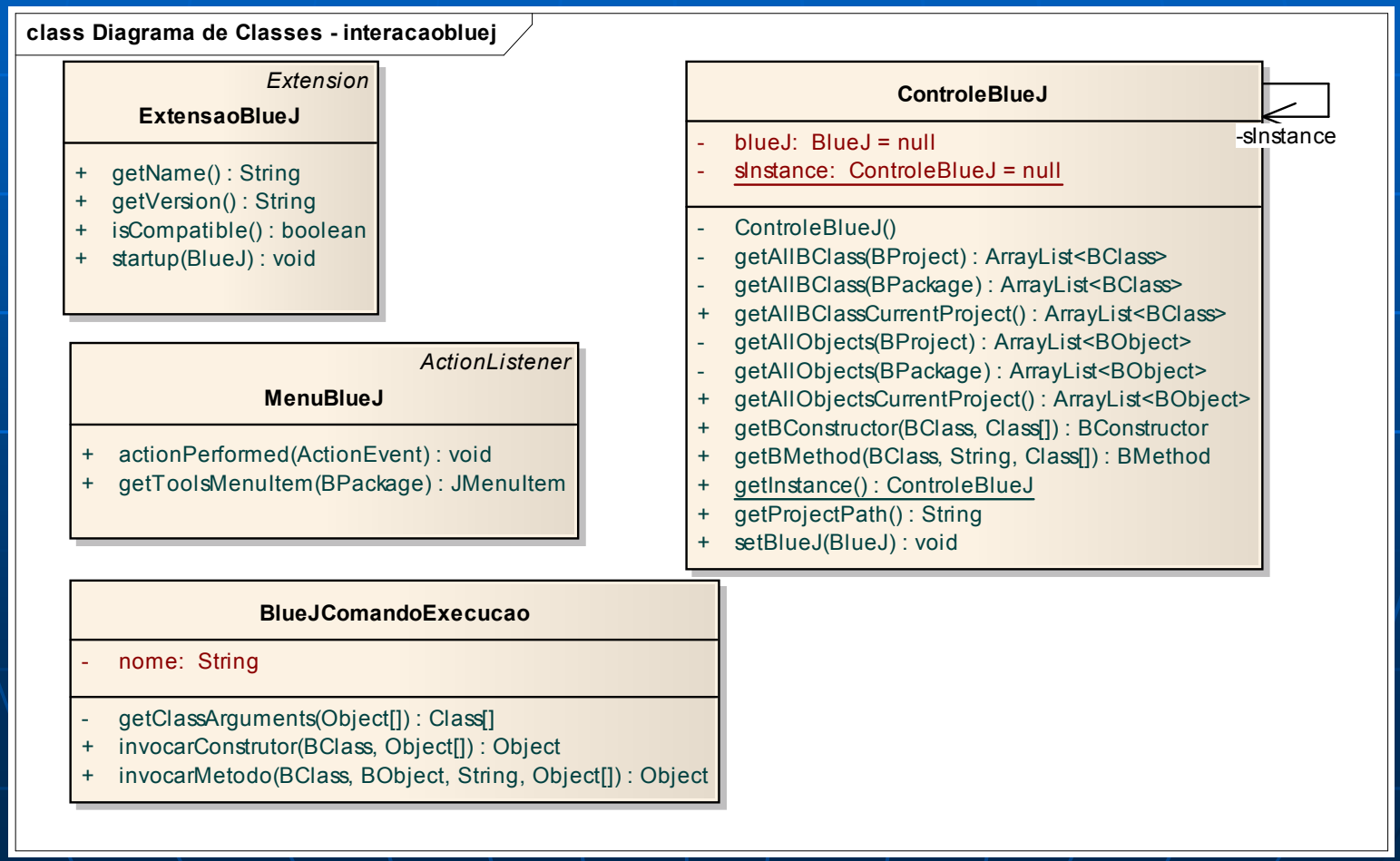
1. No passo 1 do Fluxo Principal se o arquivo foi aberto ou salvo o usuário pode fechar arquivo.
2. O usuário seleciona operação de Fechar arquivo.
3. O Sistema verificar se ouve alteração no arquivo
3.1 Caso houver alteração segue fluxo de Salvar arquivo.
4. O Sistema fecha o arquivo aberto e limpa tela de comandos.
5. Retorna para o passo 1 do Fluxo Principal.

Desenvolvimento da Ferramenta



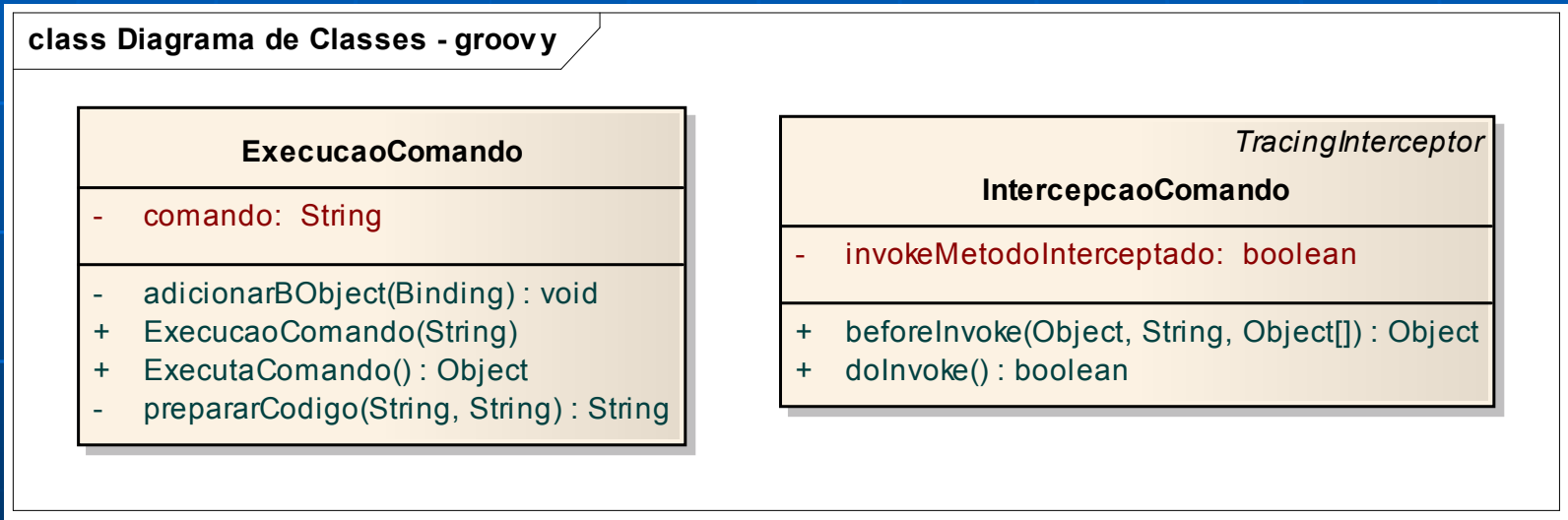
Desenvolvimento da Ferramenta

Diagrama de classes - interacaobluej

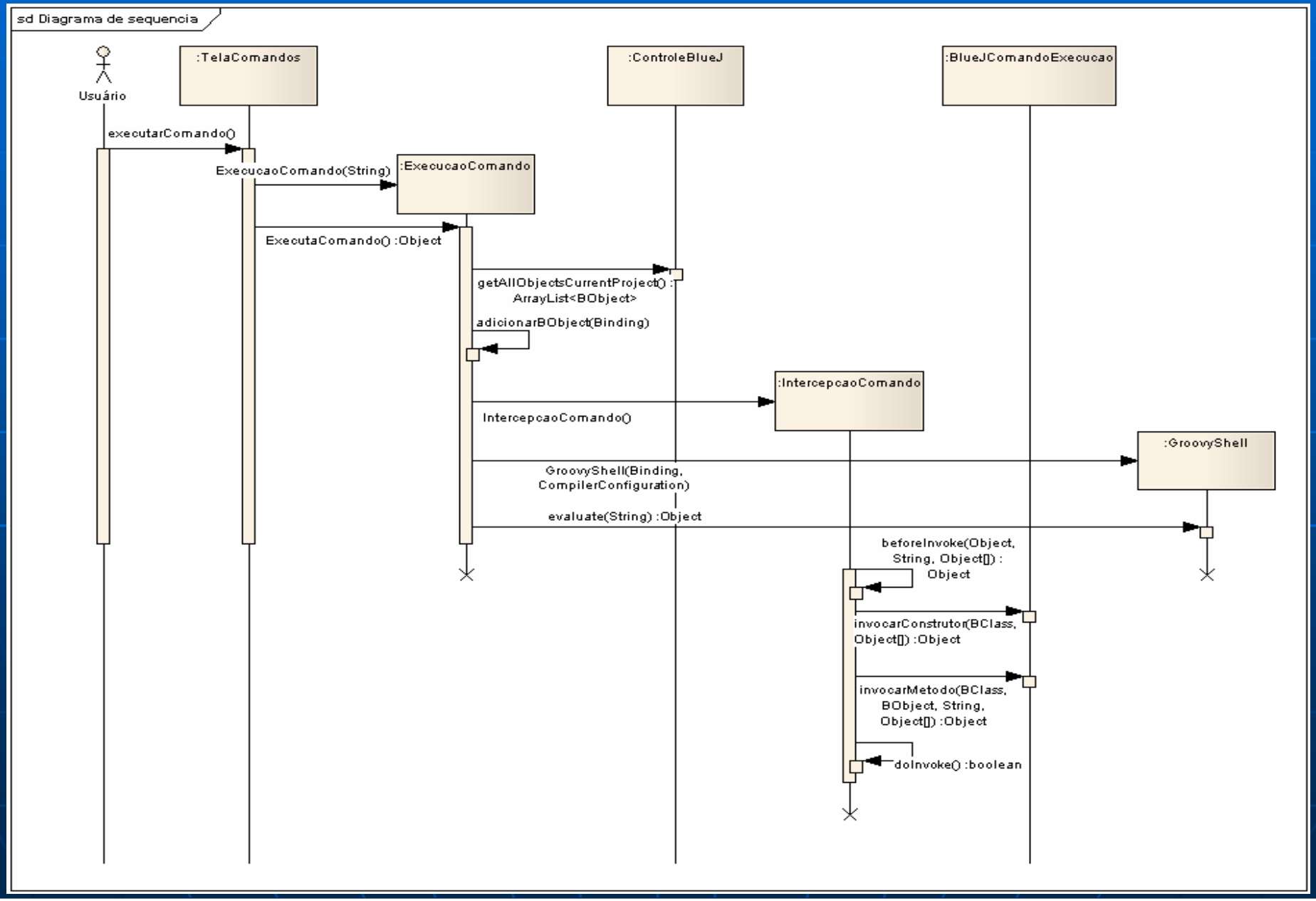


Desenvolvimento da Ferramenta

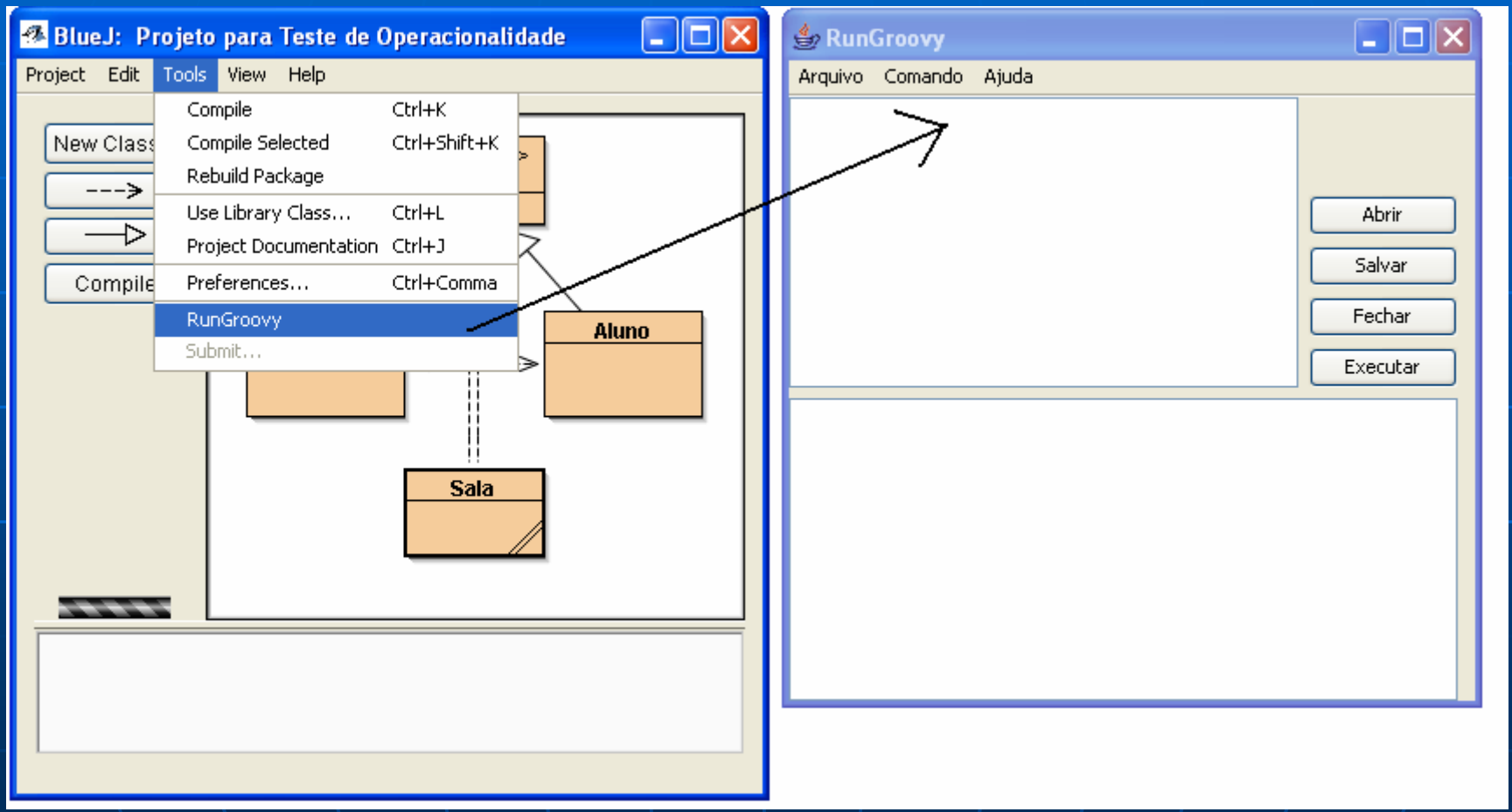
Diagrama de classes - groovy



Desenvolvimento da Ferramenta

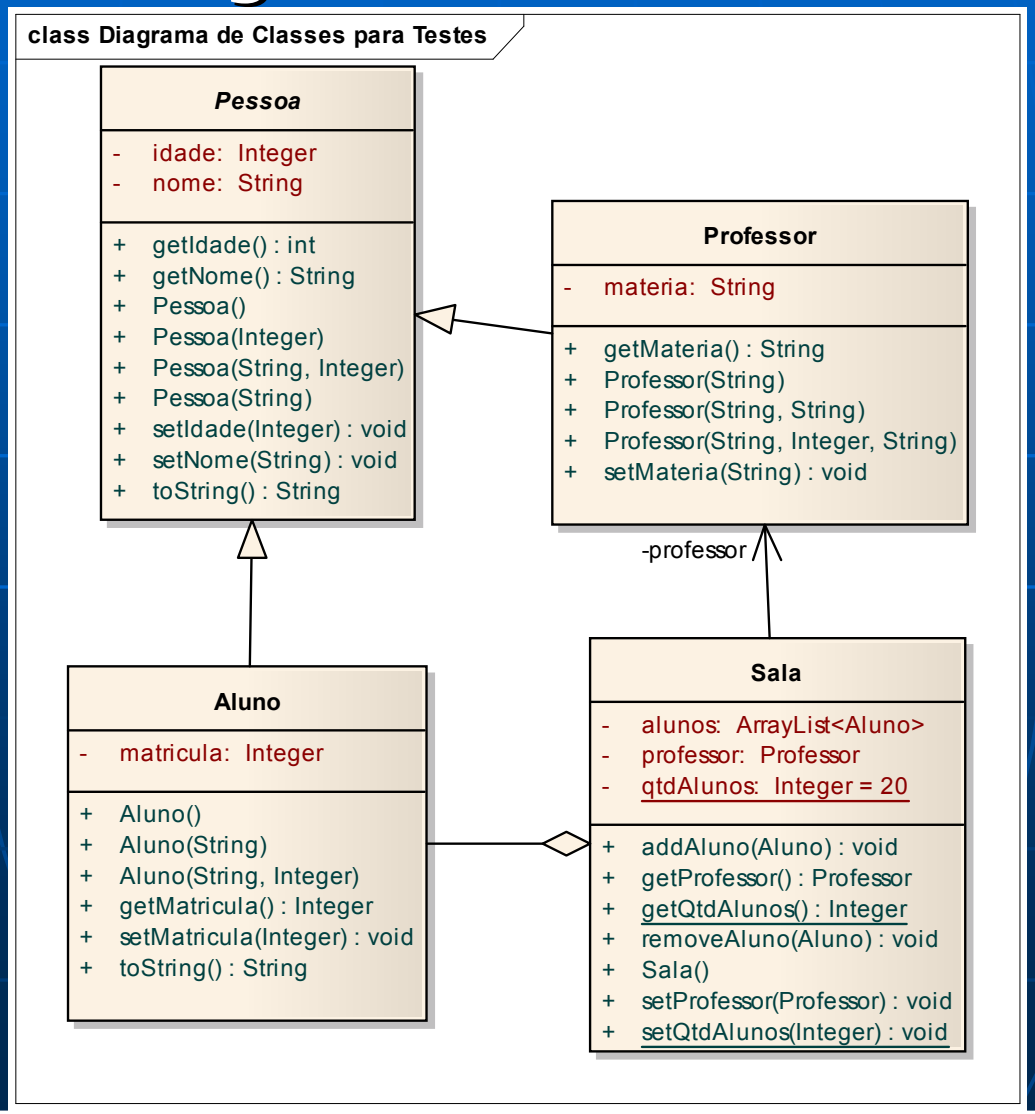


Desenvolvimento da Ferramenta



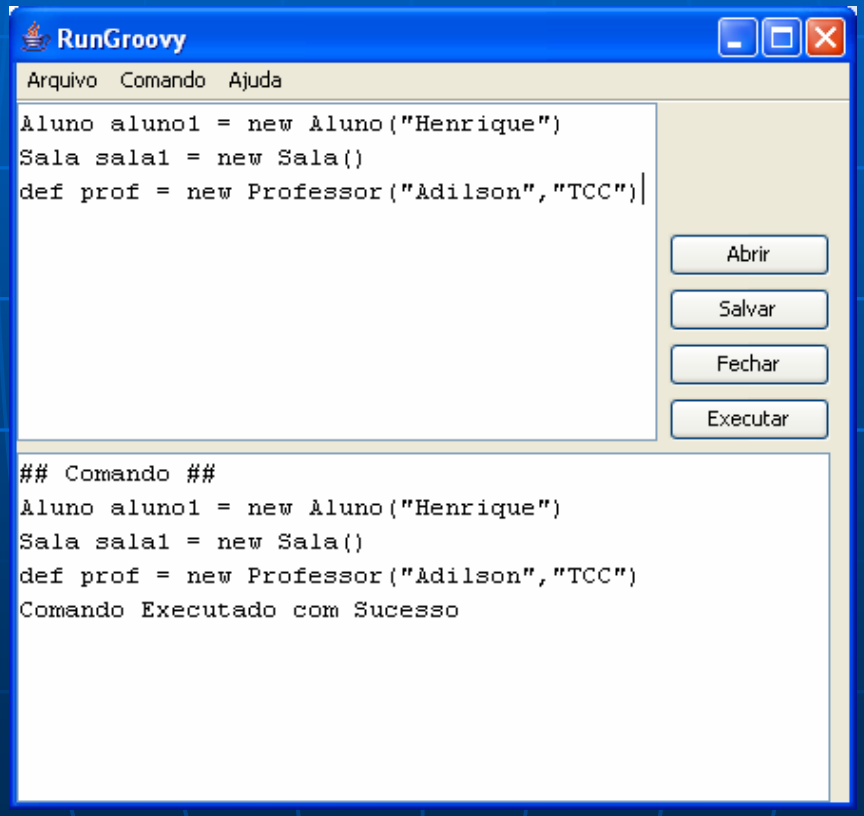
Estudo de caso

Diagrama de classes



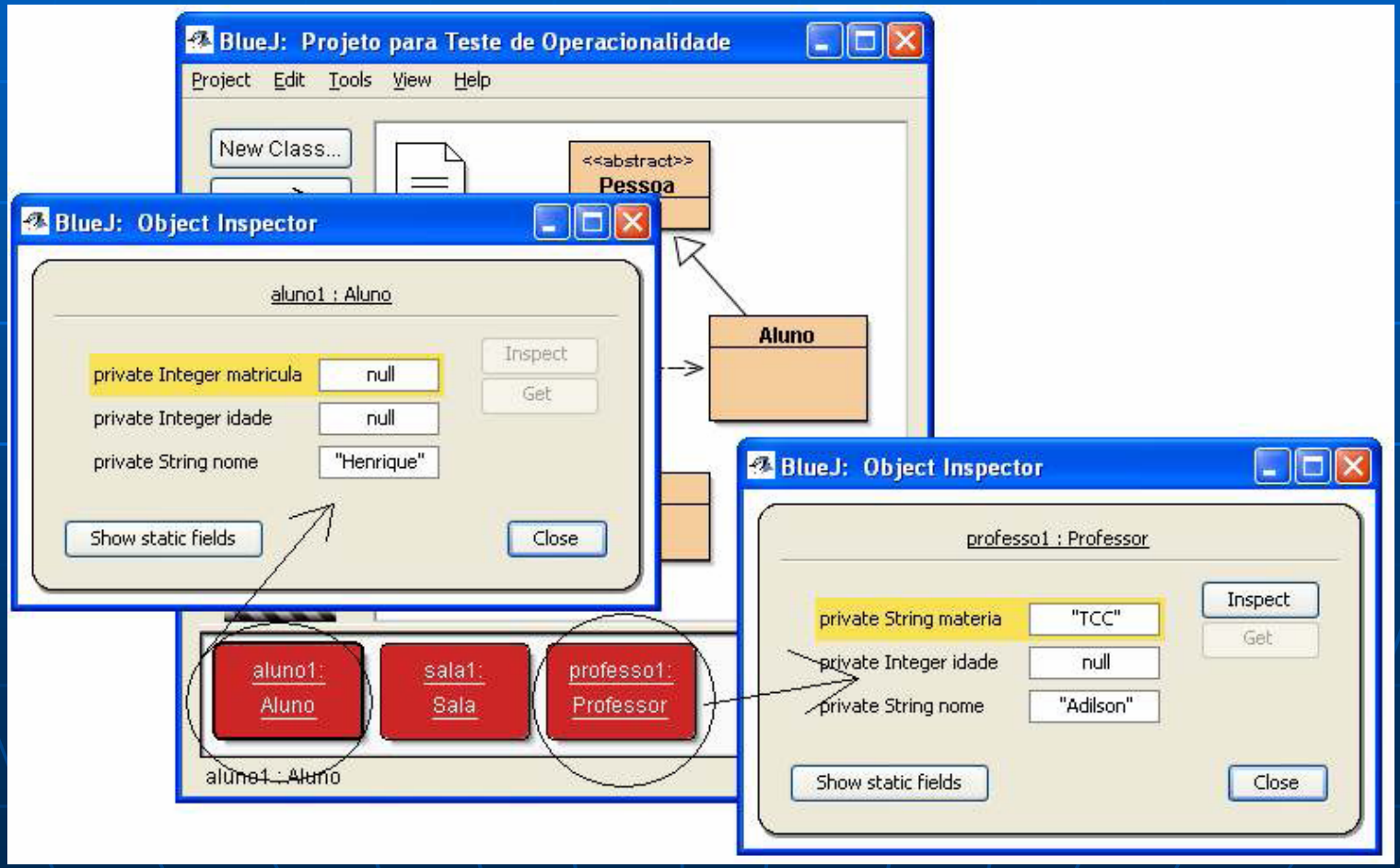
Estudo de caso

Execução de códigos



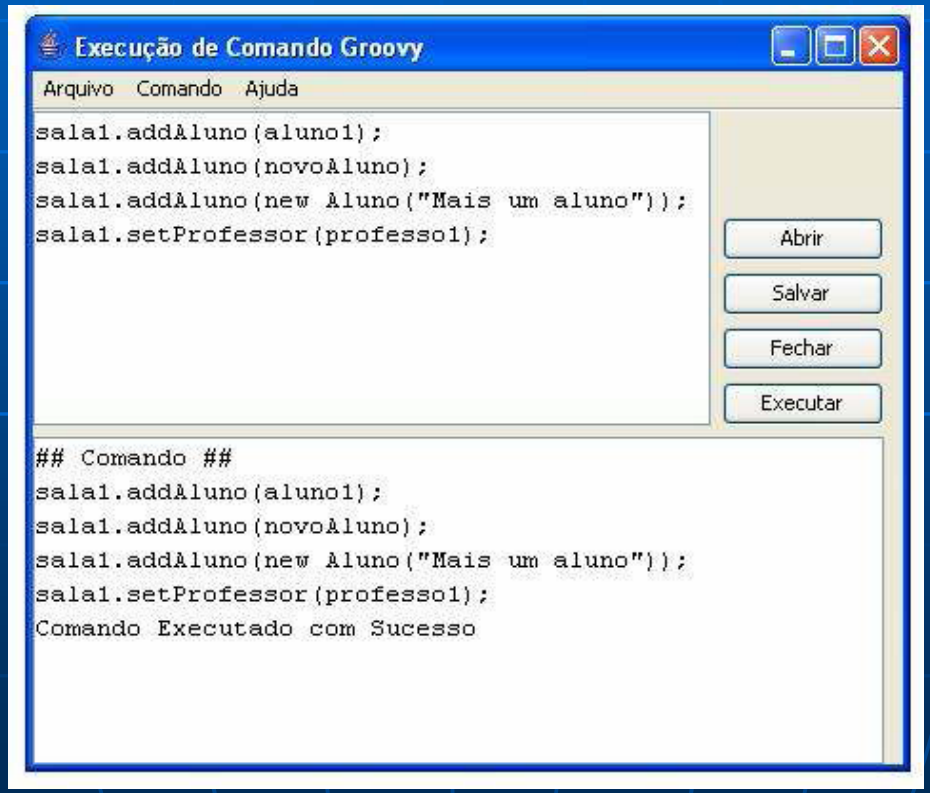
Estudo de caso

Análise dos valores

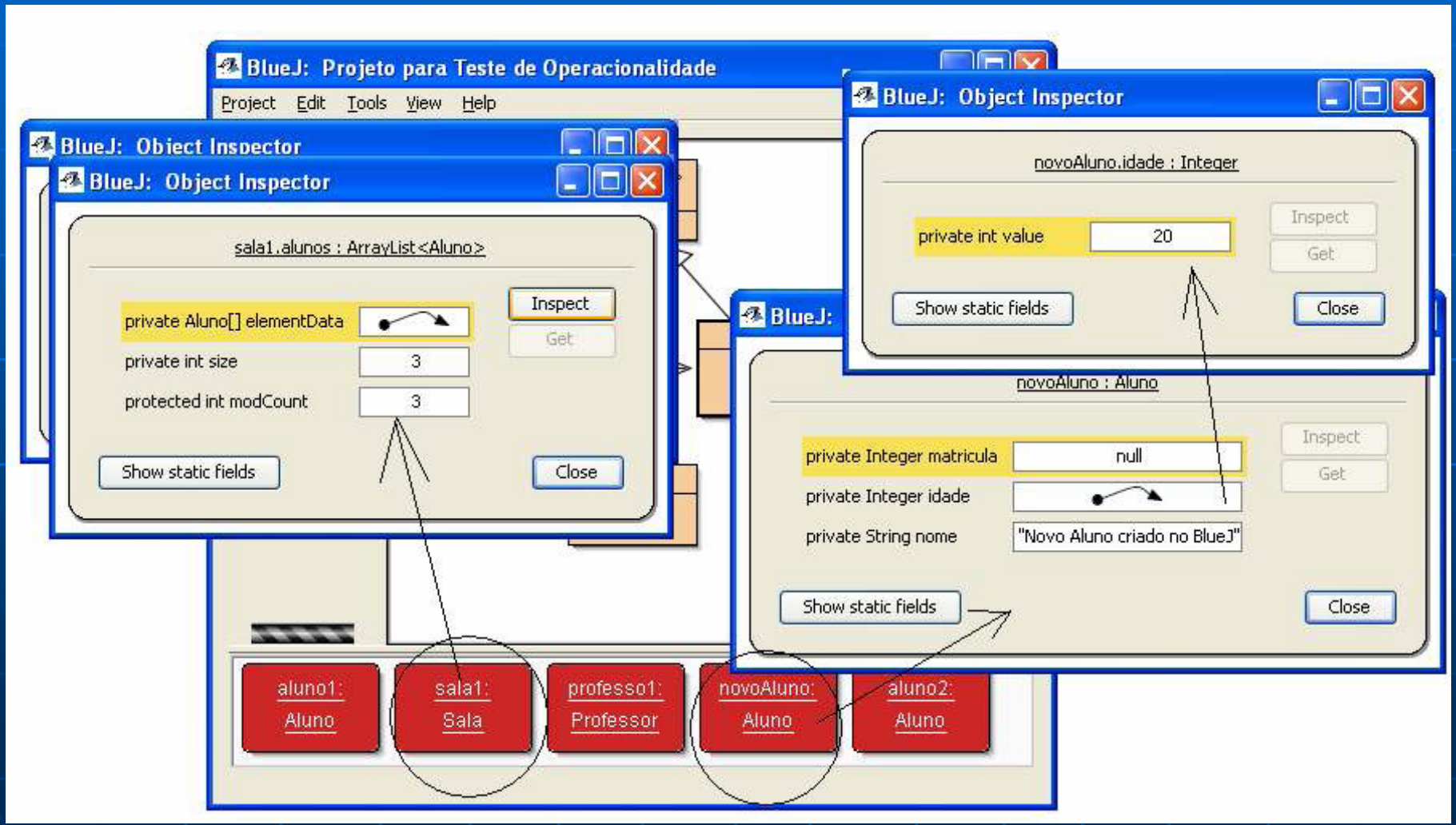


Estudo de caso

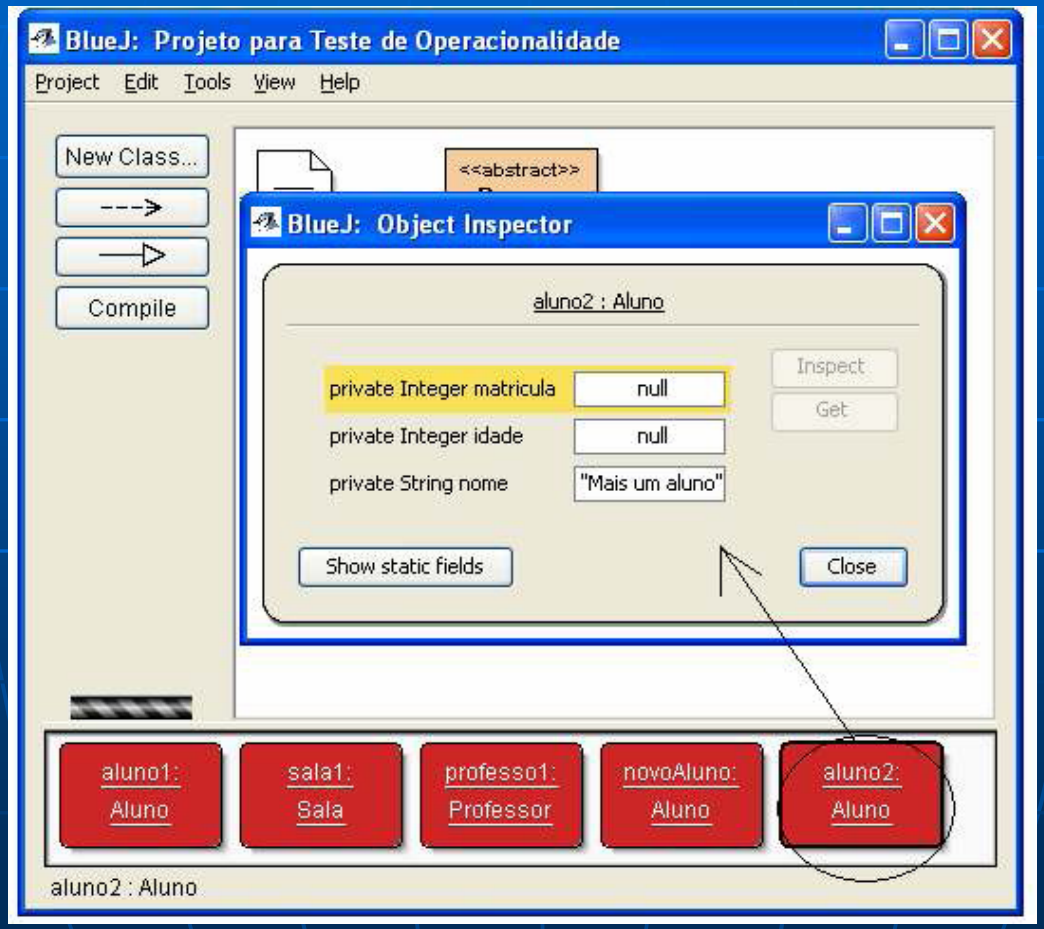
Execução de novos códigos



Estudo de caso



Estudo de caso



Resultados e Discussão

	RunGroovy	*Cod Pad* do BlueJ
Permite a interação com o usuário para criação de objetos	X	X
Permite a execução de códigos	Groovy	Java
Salva e carrega arquivos de scripts	X	
Permite a execução de estruturas de repetição	X	X
Criação de objetos no *ObjectBench*	Através dos códigos digitados	Necessita de mais interações com o Usuário

Conclusão

- Ferramenta simples de utilização
- Fácil integração entre Groovy e Java
- Groovy diminui curva de aprendizagem
- Documentação para extensão do ambiente BlueJ

Conclusão

- Limitações
  - Restrições no acesso direto a atributos
  - Os nomes dos objetos no script não refletem no ambiente BlueJ

Conclusão

- Extensões
  - Melhoria na interface console da Ferramenta
  - Novas ferramentas podem executar script Groovy pelo RunGroovy
  - Implementação de segurança no código digitado
  - Base na produção de novos trabalhos de extensão no BlueJ