# Avaliação - Exercício Prático da Semana 2

Zoraia R. Dantas

Usar a Modelagem CRC, para o SAB - Sistema de Automação de Biblioteca.

Assim solicitamos que examine o método abaixo registraUsuario(String) e faça o seguinte:

a) Identifique uma lista de maus cheiros que você encontra no código, relacionando cada um deles com o correspondente tipo de mau cheiro exercitado nesta parte do curso: [mau cheiro no código (trecho do código)/tipo de mau cheiro (de acordo com Fowler, pode estar em português)].

## 1. Lista de Maus Cheiros encontrados

## 1.0 Mau cheiro 1

- 1.1 Tipo de mau cheiro: Ifs com expressões inapropriadas (expressões com booleanas negativas)
- 1.2 Trecho do código

```
..if (nome != null) {
..if (!nome.isEmpty()) {
..if (!_usuarios.contains(usuario))
```

### 2.0 Mau cheiro 2

- 2.1 Tipo de mau cheiro: ifs aninhados
- 2.2 Trecho do código

```
if (nome != null) {
  if (!nome.isEmpty()) {
    Usuario usuario = new Usuario(nome);
    if (!_usuarios.contains(usuario)) {
        _usuarios.add(usuario);
    } else
    throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
        + nome + "\"! Use outro nome!");
    } else
    throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
    } else ..
```

b) Realize o Ciclo de Refatoração apresentado, eliminando cada um dos maus cheiros encontrado no código do método, considerando apenas os tipos de mau cheiro exercitados nesta Semana 2 do curso.

# Mau cheiro 1

Tipo de mau cheiro: Ifs com expressões inapropriadas (expressões com booleanas negativas)

# Ciclos de Fatoração

#### **Antes**

```
.. if (nome != null) { ...
```

```
1º Ciclo
```

```
if (nome == null) {
        throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
   if (!nome.isEmpty()) { //não esta vazio
   Usuario usuario = new Usuario(nome);
    if (!_usuarios.contains(usuario)) {
      _usuarios.add(usuario);
    } else
      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
                                                        + nome + "\"! Use outro nome!");
 } else
     throw new UsuarioComNomeVazioException("--->Não pode registrar usuário com nome vazio!");
}
Antes
.. if (!nome.isEmpty()) ..
2º Ciclo
if (nome == null) {
        throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
} else {
 if (nome.isEmpty()){
 throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
  }else{
    Usuario usuario = new Usuario(nome);
        if (!_usuarios.contains(usuario)) {
           _usuarios.add(usuario);
           throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
                                                        + nome + "\"! Use outro nome!");
        }
}
Antes
... if (!_usuarios.contains(usuario)) ...
3º Ciclo
if (nome == null) {
        throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
} else {
   if (nome.isEmpty()){
      throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
   }else{
      Usuario usuario = new Usuario(nome);
        if (_usuarios.contains(usuario)) {
           throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
                + nome + "\"! Use outro nome!");
          _usuarios.add(usuario);
   }
```

# Mau cheiro 2

Tipo de mau cheiro: ifs aninhados

## 3 Ciclos de Fatoração

```
Antes
if (nome == null) {
        throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
} else {
1º Ciclo - Remoção do else da primeira condição
if (nome == null)
   throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
if (nome.isEmpty()){
   throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
}else{
        Usuario usuario = new Usuario(nome);
        if (_usuarios.contains(usuario)) {
        throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
                + nome + "\"! Use outro nome!");
        } else
         _usuarios.add(usuario);
2º Ciclo - Remoção do else da segunda condição
if (nome == null)
   throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
if (nome.isEmpty())
   throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
Usuario usuario = new Usuario(nome);
if (_usuarios.contains(usuario)) {
   throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
} else
  _usuarios.add(usuario);
2º Ciclo - Remoção do else da terceira condição interna
if (nome == null)
   throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
if (nome.isEmpty())
   throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
Usuario usuario = new Usuario(nome);
if ( usuarios.contains(usuario))
 throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""+nome + "\"! Use outro nome!");
_usuarios.add(usuario);
```

- c) Entregue um documento em que você apresenta o seguinte:
- A) Código anterior do método registraUsuario(String), antes de iniciar o Ciclo de Refatoração.

### 2. Método completo

Código Antes da refatoração, com o trecho a ser refatorado com as letras coloridas ou com fundo amarelo

public void registraUsuario(String nome) throws UsuarioJaRegistradoException,
UsuarioComNomeVazioException, UsuarioInexistenteException {

```
if (nome != null) {
   if (!nome.isEmpty()) {
      Usuario usuario = new Usuario(nome);
      if (!_usuarios.contains(usuario)) {
            usuarios.add(usuario);
      } else
        throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""+ nome +
      "\"! Use outro nome!");
      } else
      throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
      } else
        throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
   }
}
```

B) Imagem: Imagem da execução bem-sucedida (verde) no Eclipse ou outro ambiente Java, comprovando que código atual do SAB, incluindo o método registraUsuario(String) está funcionando direito (pelo menos de acordo com a bateria de testes atual).

```
Package Explorer 23 1 Type Hierarchy 10 Junit

☑ Biblioteca java 
☑ HashSet.class 
☑ Conversor.java 
☑ TesteCamelCase.java 
☑ CentralNumeracaoLivro.java
                                                                 21
                                            E 8 9 V
                                                                                   throw new AdicionarLivroInexistenteException(
                                                          ٨
> B PalavrasEmbaralhadas
                                                                                              --->N<o pode adicionar livro inexistente!");
> B PilhaEncapsulada
                                                                         }
                                                          > @ Pontuação
                                                                 25
26
>  Produtos
                                                                          > ProjetoAula5
                                                                 28
29
30
31
> ProjetoPizzaria
> @ TabuleiroIMD
> 🐸 TDDArtigo
                                                                32
33
34
35
36
37
38
39
40
41
42
43
44
45⊖
v 🕭 src
                                                                                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");

→ 

⊕ pSABbyCRC_UnitTestingSuite

           AdicionarLivrolnexistenteException.java
          ) Biblioteca.java
         BibliotecaTestContextolnicial.java
         BuscaUsuarioComNomeNuloException.iava
         > D BuscaUsuarioComNomeVazioException.java
         > DevolveLivroDisponivelParaEmprestimoException
         > DevolveLivroNuloParaEmprestimoException.java
                                                                          public void emprestaLivro(Livro livro, Usuario usuario)
         ) [] Livro.java
                                                                                   throws LivroIndisponivelParaEmprestimoException, LivroOuUsuarioNulosException {
                                                                 46
47
48
49
50
51
52
53
54
55

    LivroIndisponivelParaEmprestimoException.java

          ) LivroOuUsuarioNulosException.java
                                                                              if ((livro == null) && (usuario == null))
throw new LivroOuUsuarioNulosException
          > I TituloOuAutorNuloException.java
                                                                                              --->Livro e Usu‡rio inexistentes!");
         > 1 TituloOuAutorVazioException.java
                                                                             if (livro != null) {
   if (usuario != null) {
          ) Usuario java
         UsuarioComNomeVazioException.iava
                                                                                       if (livro.getUsuario() == null) {
    usuario.anexaLivroAoUsuario(livro);

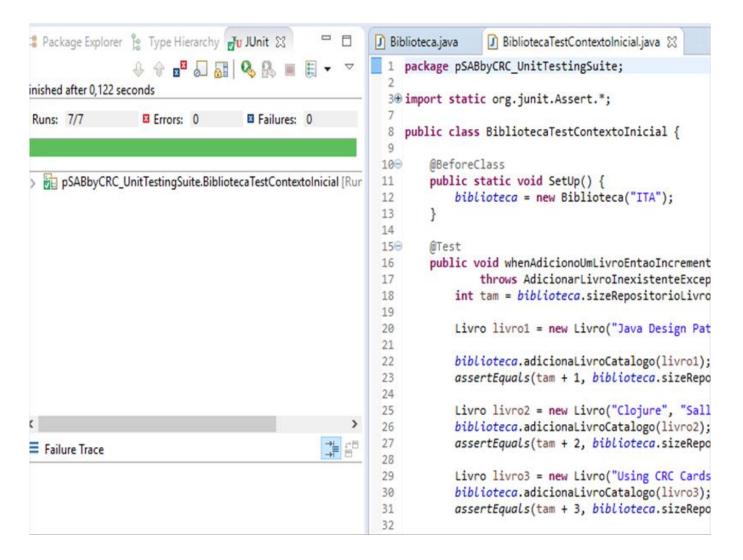
    UsuariolnexistenteException.iava

    UsuarioJaRegistradoException.java

                                                                                            livro.anexaUsuarioAoLivro(usuario);

▼ M JRE System Library [JavaSE-1.7]

        resources.jar - C:\Program Files\Java\jre7\lib
                                                                                            throw new LivroIndisponivelParaEmprestimoException(
                                                                                                      "--->Livro " + livro
```



O código Depois da refatoração, com o trecho refatorado com as letras coloridas ou com fundo verde

D) Código Depois final do método abaixo registraUsuario(String), sem letras ou fundo coloridos!