

Aluno: Fabricio Augusto Pereira

Mau Cheiro segundo Martin Fowler, comando IF e SWITCH, melhora entendimento do código tirado NEGAÇÕES dos IF e eliminando os ELSE.

```
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 usuário com nome vazio!");
    } else
        throw new UsuarioInexistenteException("--->Não pode registrar usuário inexistente!");
}
```

Teste do código inicial

The screenshot displays the JUnit test results in IntelliJ IDEA. The top panel shows the test suite 'pSABbyCRC_UnitTestingSuite' with 5/5 tests passed. The bottom panel shows the 'UsuarioTest' suite with 5/5 tests passed. The 'BibliotecaTestContextoInicial' suite is also shown with 7/7 tests passed. The 'Failure Trace' panel is empty, indicating no failures.

Package Explorer: JUnit

Finished after 0,022 seconds

Runs: 5/5 Errors: 0 Failures: 0

pSABbyCRC_UnitTestingSuite.LivroTest [Runner: JUnit 4] (0,000 s)

- whenAnexaUsuarioNuloAoLivroEntaoLivroFicaDisponivelParaEmprestimo (0,000 s)
- whenDesanexaUsuarioNuloDoLivroEntaoLivroContinuaDisponivelParaEmprestimo (0,000 s)
- whenSituacaoInicialEntaoLivroDisponivelParaEmprestimo (0,000 s)
- whenDesanexaUsuarioNaoNuloDoLivroEntaoLivroFicaDisponivelParaEmprestimo (0,000 s)
- whenAnexaUsuarioNaoNuloAoLivroEntaoLivroFicaIndisponivelParaEmprestimo (0,000 s)

Runs: 5/5 Errors: 0 Failures: 0

pSABbyCRC_UnitTestingSuite.UsuarioTest [Runner: JUnit 4] (0,000 s)

- whenAnexaLivroNuloAListaDeLivrosEmprestadosPeloUsuarioEntaoTamanhoDeListaDeLivrosEmprestadosNaoMuda (0,001 s)
- whenSituacaoInicialEntaoLivroDisponivelParaEmprestimo (0,000 s)
- whenDesanexaLivroNaoNuloDeLivrosEmprestadosPeloUsuarioDecrementaHumDeListaDeLivrosEmprestados (0,000 s)
- whenAnexaLivroNaoNuloAListaDeLivrosEmprestadosPeloUsuarioEntaoIncrementaHumDeListaDeLivrosEmprestados (0,000 s)
- whenDesanexaLivroNuloDeLivrosEmprestadosPeloUsuarioEntaoTamanhoDeListaDeLivrosEmprestadosNaoMuda (0,000 s)

Failure Trace

Runs: 7/7 Errors: 0 Failures: 0

pSABbyCRC_UnitTestingSuite.BibliotecaTestContextoInicial [Runner: JUnit 4] (0,000 s)

- whenAdicionoLivroNuloEntaoAdicionarLivroInexistenteExceptionEhLancada (0,000 s)
- whenAdicionoUsuarioJaExistenteEntaoUsuarioJaRegistradoExceptionEhLancada (0,000 s)
- whenAdicionoUmLivroEntaoIncrementaTamListaLivrosDisponiveisDeUm (0,000 s)
- whenAdicionoUsuarioInexistenteEntaoUsuarioInexistenteExceptionEhLancada (0,000 s)
- whenAdicionoUmUsuarioEntaoIncrementaTamListaUsuariosDeUm (0,000 s)
- whenAdicionoOutroLivroComMesmoTituloEAutorEntaoIncrementaTamListaLivrosDisponiveisDeUm (0,000 s)
- whenAdicionoUsuarioComNomeVazioEntaoUsuarioComNomeVazioExceptionEhLancada (0,000 s)

```
Runs: 23/23      Errors: 0      Failures: 0

pSABbyCRC_UnitTestingSuite.BibliotecaTest [Runner: JUnit 4] (0,001 s)
  whenBuscoUsuarioPorNomeExistenteEntaoRetornoUsuarioValido (0,000 s)
  whenBuscoLivroPorTituloEAutorEntaoRetornoLivroCujoTituloEAutorConfere (0,000 s)
  whenBuscoLivroPorTituloOuAutorENaoEncontroEntaoRetornoLivroNulo (0,000 s)
  whenEmprestoLivroNaoNuloAUsuarioNuloEntaoLivroOuUsuarioNulosExceptionEhLancada (0,000 s)
  whenDevolvoLivroNuloParaEmprestimoEntaoDevolveLivroNuloParaEmprestimoExceptionEhLancada (0,000 s)
  whenEmprestoUmLivroAUsuarioEntaoLivroFicaIndisponivelParaEmprestimo (0,000 s)
  whenBuscoLivroPorTituloNaoNuloEAutorNuloEntaoTituloOuAutorVazioExceptionEhLancada (0,000 s)
  whenBuscoLivroPorTituloVazioEAutorVazioEntaoTituloOuAutorVazioExceptionEhLancada (0,000 s)
  whenBuscoLivroPorTituloNuloEAutorNuloEntaoTituloOuAutorVazioExceptionEhLancada (0,000 s)
  whenDevolvoUmLivroEmprestadoAUsuarioEntaoLivroFicaDisponivelParaEmprestimo (0,000 s)
  whenEmprestoLivroNuloAUsuarioNuloEntaoLivroOuUsuarioNulosExceptionEhLancada (0,000 s)
  whenEmprestoLivroNaoNuloAUsuarioNaoNuloEntaoLivroOuUsuarioNulosExceptionEhLancada (0,000 s)
  whenBuscoLivroPorTituloVazioEAutorNaoVazioEntaoTituloOuAutorVazioExceptionEhLancada (0,000 s)
  whenBuscoLivroPorTituloNuloEAutorNaoNuloEntaoTituloOuAutorVazioExceptionEhLancada (0,001 s)
  whenBuscoLivroPeloNrCatalogoInexistenteEntaoLivroEhNulo (0,000 s)
  whenBuscoUsuarioPorNomeVazioEntaoBuscaUsuarioComNomeVazioExceptionEhLancada (0,000 s)
  whenBuscoLivroPeloNrCatalogoEntaoRetornoLivroCujoNrCatalogoConfere (0,000 s)
  whenEmprestoTresLivrosAUmUnicosUsuarioSemLivrosEmprestadosEntaoListaLivrosdoUsuarioTemTresLivros (0,000 s)
  whenBuscoUsuarioPorNomeInexistenteEntaoRetornoUsuarioNulo (0,000 s)
  whenDevolvoLivroDisponivelParaEmprestimoEntaoDevolveLivroDisponivelParaEmprestimoExceptionEhLancada (0,000 s)
  whenBuscoUsuarioPorNomeNuloEntaoBuscaUsuarioComNomeNuloExceptionEhLancada (0,000 s)
  whenEmprestoLivroJahEmprestadoEntaoLivroIndisponivelParaEmprestimoExceptionEhLancada (0,000 s)
```

Versão 1 do Código eliminando negação no IF e tirar os ELSE, com os testes.

```
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!");
}

22         "--->Não pode adicionar livro inexistente!");
23     }
24
25     public void registraUsuario(String nome)
26         throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
27                UsuarioInexistenteException {
28         if (nome == null) throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29
30         if (!nome.isEmpty()) {
31             Usuario usuario = new Usuario(nome);
32             if (!_usuarios.contains(usuario)) {
33                 _usuarios.add(usuario);
34             } else
35                 throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
36                     + nome + "\"! Use outro nome!");
37         } else
38             throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
39     }
40 }
```

Versão 2 do Código eliminando negação no IF e tirar os ELSE, com os testes.

```
22         "---->N@o pode adicionar livro inexistente!");
23     }
24
25     public void registraUsuario(String nome)
26         throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
27             UsuarioInexistenteException {
28         if (nome == null) throw new UsuarioInexistenteException("---->N@o pode registrar usuario inexistente!");
29
30         if (!nome.isEmpty()) {
31             Usuario usuario = new Usuario(nome);
32             if (!_usuarios.contains(usuario)) {
33                 _usuarios.add(usuario);
34             } else
35                 throw new UsuarioJaRegistradoException("---->J@ existe usu@rio com o nome \""
36                     + nome + "\"! Use outro nome!");
37         } else
38             throw new UsuarioComNomeVazioException("---->N@o pode registrar usuario com nome vazio!");
39     }
40
```

```
22
23     public void registraUsuario(String nome)
24         throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
25         if (nome == null)
26             throw new UsuarioInexistenteException("---->N@o pode registrar usuario inexistente!");
27
28         if (nome.isEmpty())
29             throw new UsuarioComNomeVazioException("---->N@o pode registrar usuario com nome vazio!");
30         Usuario usuario = new Usuario(nome);
31         if (!_usuarios.contains(usuario)) {
32             _usuarios.add(usuario);
33         } else
34             throw new UsuarioJaRegistradoException(
35                 "---->J@ existe usu@rio com o nome \"" + nome + "\"! Use outro nome!");
36     }
37
38
```

Versão 3 do Código eliminando negação no IF e tirar os ELSE, com os testes.

```
22
23     public void registraUsuario(String nome)
24         throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
25         if (nome == null)
26             throw new UsuarioInexistenteException("---->N@o pode registrar usuario inexistente!");
27
28         if (nome.isEmpty())
29             throw new UsuarioComNomeVazioException("---->N@o pode registrar usuario com nome vazio!");
30         Usuario usuario = new Usuario(nome);
31         if (!_usuarios.contains(usuario)) {
32             _usuarios.add(usuario);
33         } else
34             throw new UsuarioJaRegistradoException(
35                 "---->J@ existe usu@rio com o nome \"" + nome + "\"! Use outro nome!");
36     }
37
38
```

```
22
23     public void registraUsuario(String nome)
24         throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
25         if (nome == null)
26             throw new UsuarioInexistenteException("---->N@o pode registrar usuario inexistente!");
27
28         if (nome.isEmpty())
29             throw new UsuarioComNomeVazioException("---->N@o pode registrar usuario com nome vazio!");
30
31         Usuario usuario = new Usuario(nome);
32         if (_usuarios.contains(usuario))
33             throw new UsuarioJaRegistradoException(
34                 "---->J@ existe usu@rio com o nome \"" + nome + "\"! Use outro nome!");
35         _usuarios.add(usuario);
36     }
37
38
```