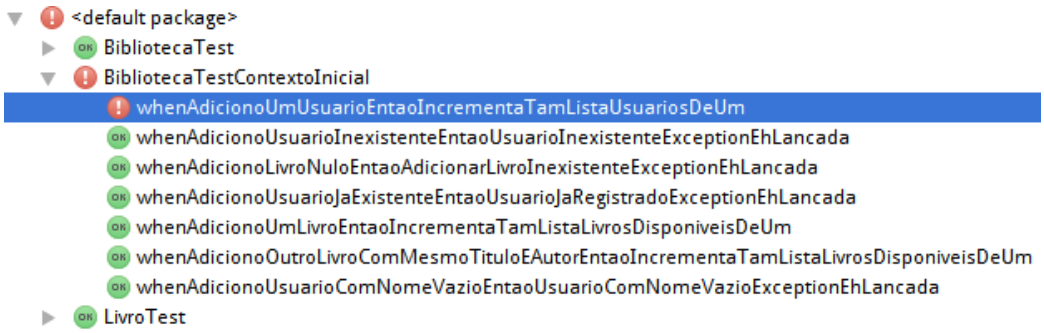
Nome: Regis Hideki Hattori

Refatoração do SAB

1 - Corrigindo o teste da implementação original

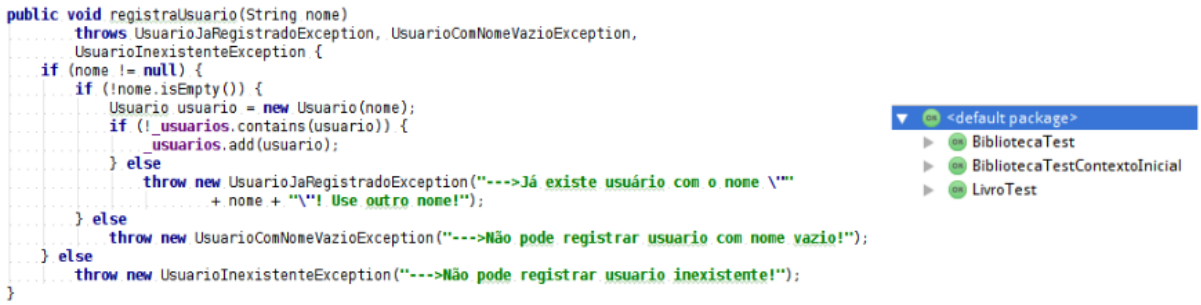
Como relatado no fórum (https://www.coursera.org/learn/tdd-desenvolvimento-de-software-guiado-por-testes/discussions/weeks/2/threads/USbm3yf6Eea3ThLvjcf5Xw?sort=createdAtAsc&page=1), o código original não passa nos testes, como pode ser visto na figura abaixo:



Ao analisar o código e os testes, foi constatado que o erro ocorria pelo fato do teste usar um *BeforeClass* para instanciar o objeto sendo testado. Isso fazia com que um único objeto fosse compartilhado entre todos os testes, o que é uma má prática. Para resolver o problema, foi necessário mudar de *BeforeClass* para *Before*. Por consequência, foi necessário retirar o *static,* já que o *Before* é uma *annotation* para um método de instância, e não de classe, como ocorre com a *annotation BeforeClass*. A alteração descrita pode ser vista na imagem abaixo:

2 - Código funcionando

Com a correção realizada na seção anterior, todos os testes passaram, conforme pode ser visto na figura abaixo:



3 - Ciclos de refatoração

3.1 - Eliminando expressões booleanas negativas:

Foram identificados 3 expressões booleanas negativas:

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

Para corrigí-las, foram realizados 3 passos, um para cada expressão, retirando a negação de cada expressão booleana e, por consequência, os códigos presentes nos *if/else*'s correspondentes.

3.1.1 - Eliminando expressão booleana negativa: *nome != null*

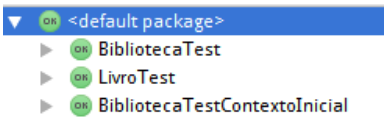
Antes:

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

Depois:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
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      }
```

Testes passando:



3.1.2 - Eliminando expressão negativa: *!nome.isEmpty()*

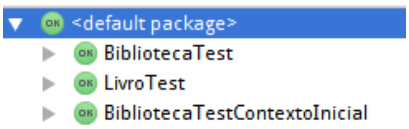
Antes:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
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      }
```

Depois:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
30              if (nome.isEmpty()) {
31                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
32              } else {
33                  Usuario usuario = new Usuario(nome);
34                  if (!_usuarios.contains(usuario)) {
35                      _usuarios.add(usuario);
36                  } else
37                      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
38                              + nome + "\"! Use outro nome!");
39              }
40          }
41      }
```

Testes passando:



3.1.3 - Eliminando expressão negativa: *!_usuarios.contains(usuario)*

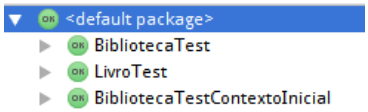
Antes:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
30              if (nome.isEmpty()) {
31                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
32              } else {
33                  Usuario usuario = new Usuario(nome);
34                  if (!_usuarios.contains(usuario)) {
35                      _usuarios.add(usuario);
36                  } else
37                      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
38                              + nome + "\"! Use outro nome!");
39              }
40          }
41      }
```

Depois:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
30              if (nome.isEmpty()) {
31                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
32              } else {
33                  Usuario usuario = new Usuario(nome);
34                  if (_usuarios.contains(usuario)) {
35                      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
36                              + nome + "\"! Use outro nome!");
37                  } else {
38                      _usuarios.add(usuario);
39                  }
40              }
41          }
42      }
```

Testes passando:



3.2 - Eliminando *if/else's* aninhados

Como poder visto na imagem abaixo, após a remoção das expressões booleanas negativas, ainda sobraram muitos if/else's aninhados:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
30              if (nome.isEmpty()) {
31                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
32              } else {
33                  Usuario usuario = new Usuario(nome);
34                  if (_usuarios.contains(usuario)) {
35                      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
36                              + nome + "\"! Use outro nome!");
37                  } else {
38                      _usuarios.add(usuario);
39                  }
40              }
41          }
42      }
```

3.2.1 - Eliminando um nível.

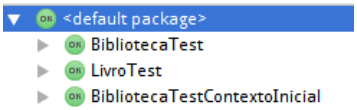
Como o primeiro *if* lança uma exceção, não é necessário usar o *else* destacado abaixo:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          } else {
30              if (nome.isEmpty()) {
31                  throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
32              } else {
33                  Usuario usuario = new Usuario(nome);
34                  if (_usuarios.contains(usuario)) {
35                      throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
36                              + nome + "\"! Use outro nome!");
37                  } else {
38                      _usuarios.add(usuario);
39                  }
40              }
41          }
42      }
```

Segue o código após a eliminação do *else*, com o segundo *if* no mesmo nível do primeiro:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          }
30
31          if (nome.isEmpty()) {
32              throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
33          } else {
34              Usuario usuario = new Usuario(nome);
35              if (_usuarios.contains(usuario)) {
36                  throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
37                          + nome + "\"! Use outro nome!");
38              } else {
39                  _usuarios.add(usuario);
40              }
41          }
42      }
```

Os testes continuam passando:



3.2.2 - Eliminando mais um nível.

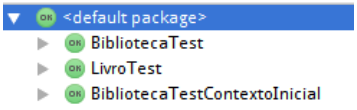
Pelo mesmo motivo relatado no ciclo anterior, o *else* abaixo também pode ser retirado:

```
24      public void registraUsuario(String nome)
25              throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
26              UsuarioInexistenteException {
27          if (nome == null) {
28              throw new UsuarioInexistenteException("--->Não pode registrar usuario inexistente!");
29          }
30
31          if (nome.isEmpty()) {
32              throw new UsuarioComNomeVazioException("--->Não pode registrar usuario com nome vazio!");
33          } else {
34              Usuario usuario = new Usuario(nome);
35              if (_usuarios.contains(usuario)) {
36                  throw new UsuarioJaRegistradoException("--->Já existe usuário com o nome \""
37                          + nome + "\"! Use outro nome!");
38              } else {
39                  _usuarios.add(usuario);
40              }
41          }
42      }
```

Código após a eliminação do *else*:

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

O código em verde, que antes estava dentro do *else*, passa a ficar no meso nível dos dois primeiros *if*'s. E os testes continuam passando?



3.2.3 - Eliminação de mais um nível.

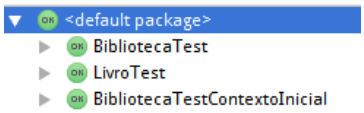
Pelo mesmo motivo relatado nos dois ciclos anteriores, o *else* abaixo pode ser removido:

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

Código após a remoção do *else*:

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

Como ocorreu nas duas etapas anteriores, o código que estava dentro do *else* (destacado em verde na imagem acima), passou a ficar no mesmo nível dois *if*'s anteriores. E o código continua passando nos testes:



Não foram encontrados mais maus cheiros entre os relatados neste módulo do curso (nomes inapropriados, comentários e código duplicado). Portanto, a versão final ficou com o código abaixo:

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