

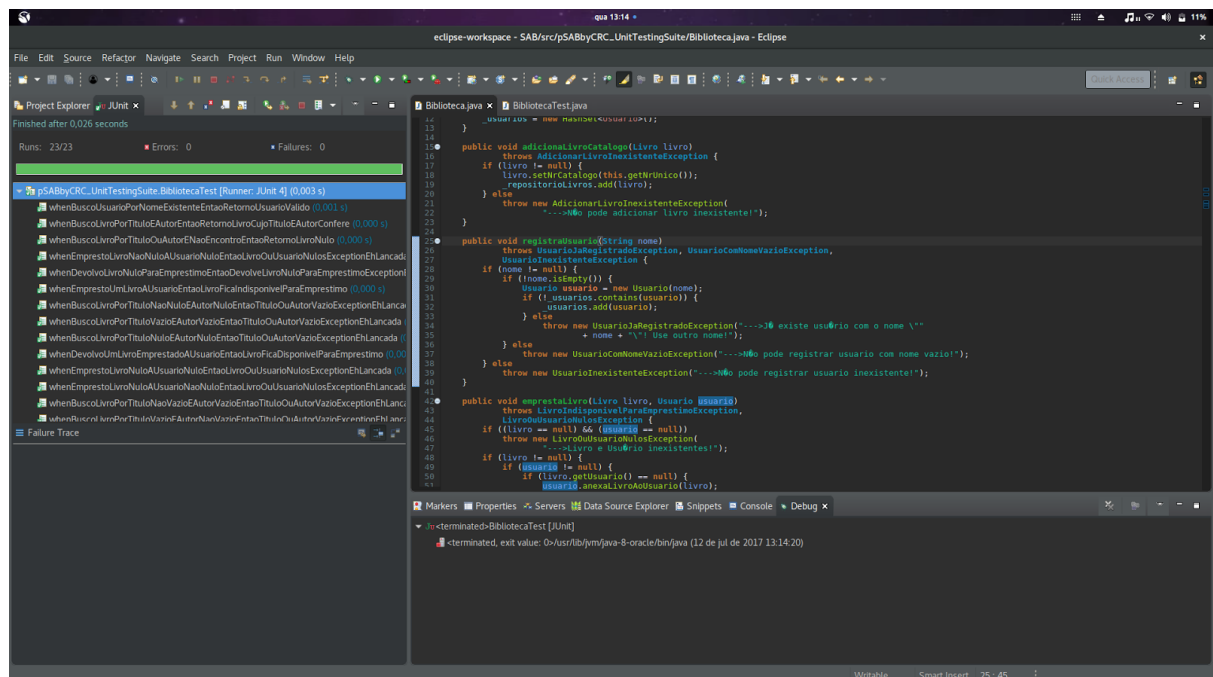
Relatório Atividade 2

Desenvolvimento de Software Guiado por Testes - ITA

Começamos com o código inicial planejado para atividade, o método registraUsuário(String):

```
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("--->J❖ existe
usu❖rio com o nome \"\"+ nome + \"\"! Use outro nome!");
        } else
            throw new UsuarioComNomeVazioException("--->N❖o pode registrar
usuario com nome vazio!");
        } else
            throw new UsuarioInexistenteException("--->N❖o pode registrar usuario
inexistente!");
    }
}
```

O código acima, como demonstrado pela imagem abaixo, passa todos os testes propostos pelo próprio modelo do SAB.



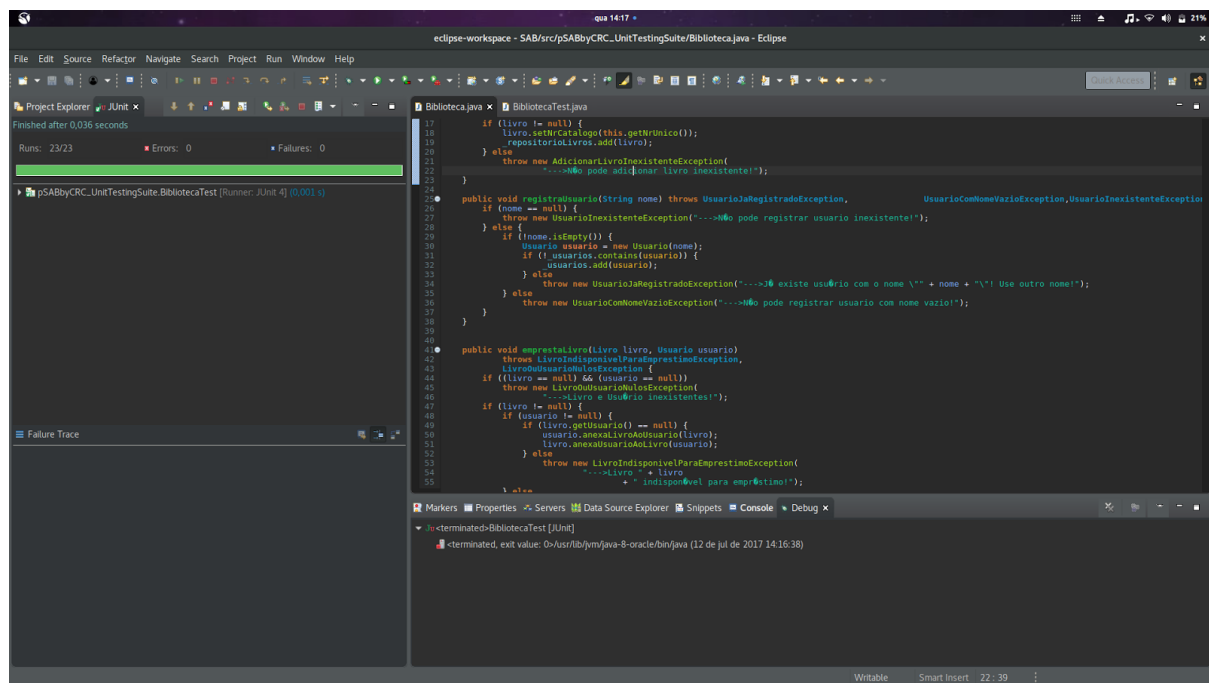
Na tabela abaixo, listamos os maus cheiros encontrados à partir da inspeção do código inicial.

Tipo de Mau Cheiro	Trecho(s) do código
Condicionais negativos	<pre>if (nome != null) if (!nome.isEmpty) if (!_usuarios.contains(usuario))</pre>
Condicionais aninhados	<pre>if (nome != null) { ... if (!nome.isEmpty) { ... if (!_usuarios.contains(usuario)) {</pre>

Começamos, então, pelo primeiro mau cheiro listado: os condicionais negativos. Podemos reverter esse mau cheiro simplesmente invertendo os comandos condicionais, associando a afirmação positiva ao campo *then* e a afirmação negativa ao campo *else*.

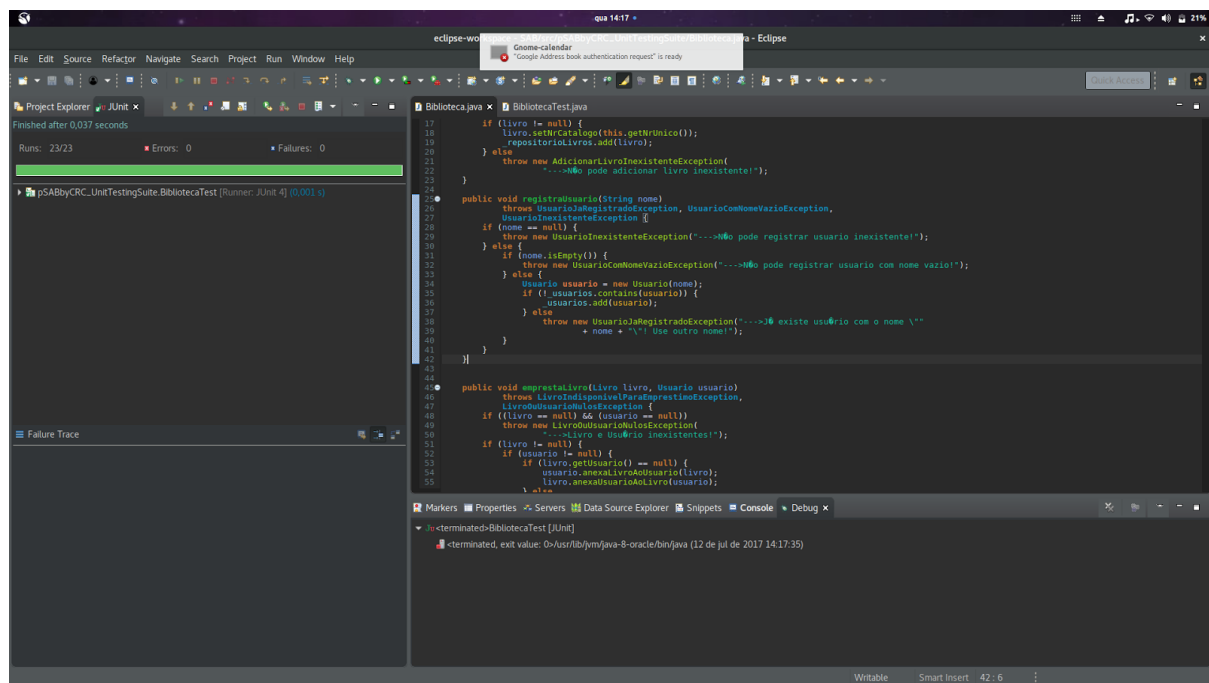
Nosso primeiro ciclo de refatoração aplica a técnica no if mais externo, resultando no código abaixo. Os testes são rodados novamente, nos certificando de que tudo continua funcionando.

```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException,
UsuarioComNomeVazioException, UsuarioInexistenteException {
    if (nome == null) {
        throw new UsuarioInexistenteException("--->N o pode registrar usuario
inexistente!");
    } else {
        if (!nome.isEmpty()) {
            Usuario usuario = new Usuario(nome);
            if (!_usuarios.contains(usuario)) {
                _usuarios.add(usuario);
            } else
                throw new UsuarioJaRegistradoException("--->J  existe
usu rio com o nome \"" + nome + "\"! Use outro nome!");
        } else
            throw new UsuarioComNomeVazioException("--->N o pode registrar
usuario com nome vazio!");
    }
}
```



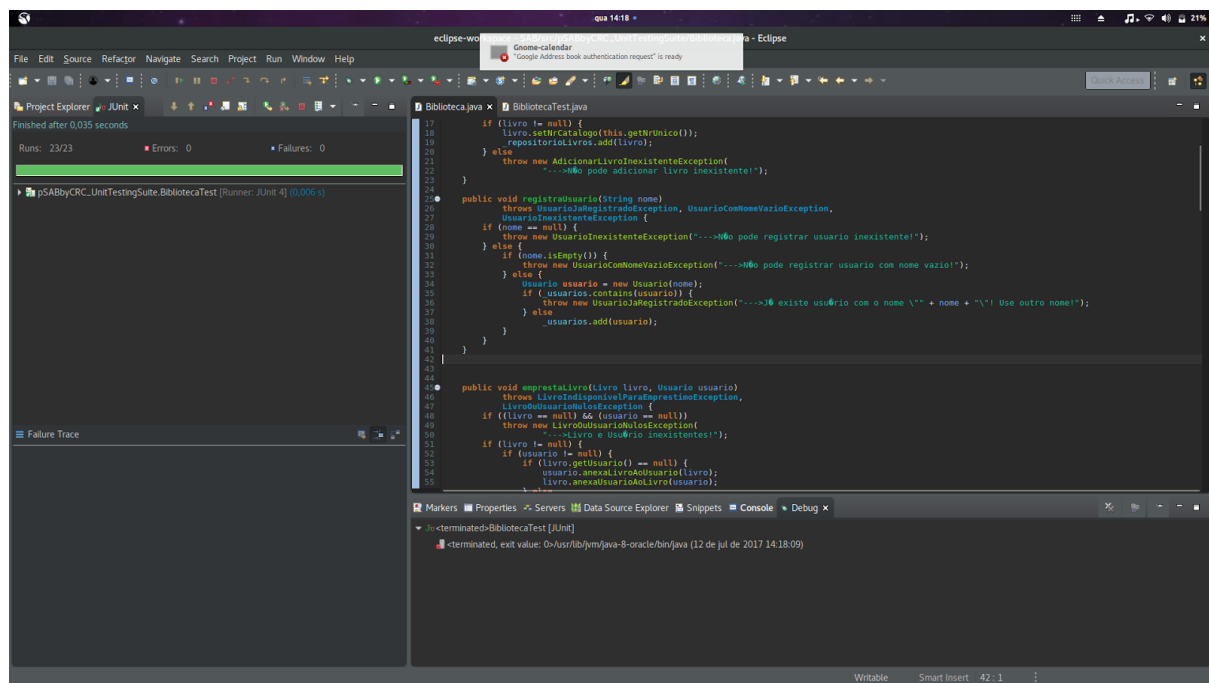
O segundo ciclo de refatoração aplica a mesma técnica no segundo if, mais uma vez conferindo que todos os testes passam:

```
public void registraUsuario(String nome)
    throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
    UsuarioInexistenteException {
    if (nome == null) {
        throw new UsuarioInexistenteException("--->N o pode registrar usuario
inexistente!");
    } else {
        if (nome.isEmpty()) {
            throw new UsuarioComNomeVazioException("--->N o pode registrar
usuario com nome vazio!");
        } else {
            Usuario usuario = new Usuario(nome);
            if (!_usuarios.contains(usuario)) {
                _usuarios.add(usuario);
            } else
                throw new UsuarioJaRegistradoException("--->J  existe
usu rio com o nome \"\"
                    + nome + "\"! Use outro nome!");
        }
    }
}
```



E no terceiro if:

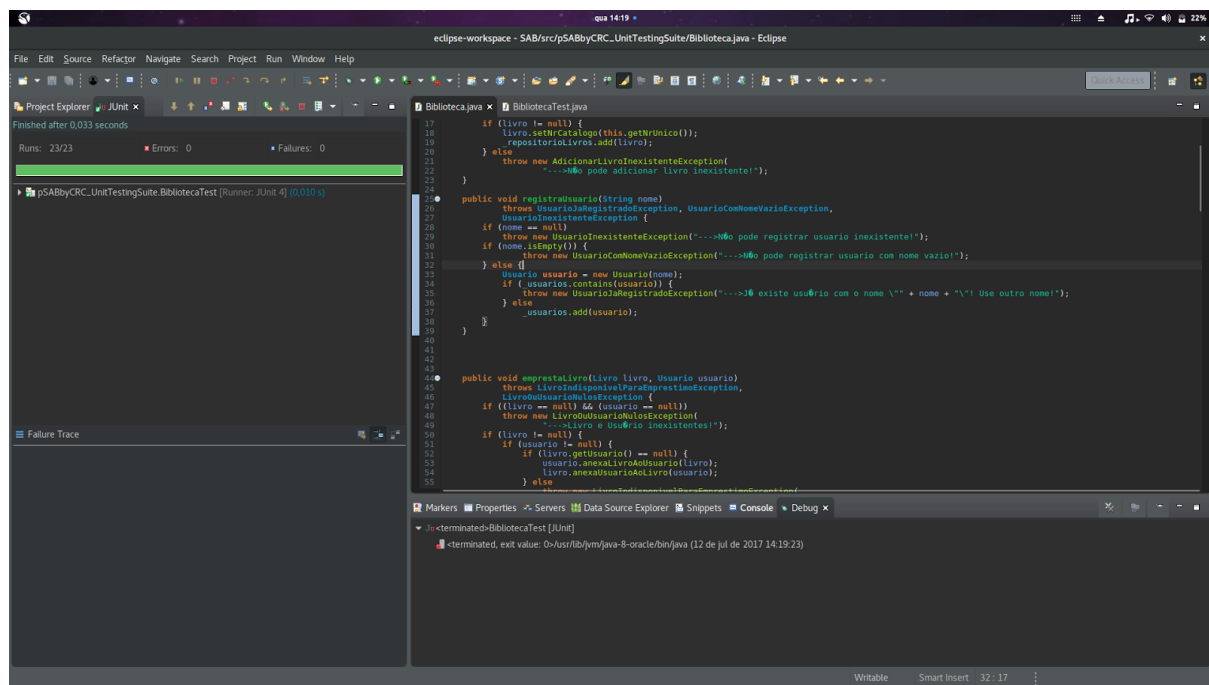
```
public void registraUsuario(String nome)
    throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
    UsuarioInexistenteException {
    if (nome == null) {
        throw new UsuarioInexistenteException("--->N o pode registrar usuario
inexistente!");
    } else {
        if (nome.isEmpty()) {
            throw new UsuarioComNomeVazioException("--->N o pode registrar
usuario com nome vazio!");
        } else {
            Usuario usuario = new Usuario(nome);
            if (_usuarios.contains(usuario)) {
                throw new UsuarioJaRegistradoException("--->J  existe
usu rio com o nome \"" + nome + "\"! Use outro nome!");
            } else
                _usuarios.add(usuario);
        }
    }
}
```



Agora que o primeiro mau cheiro foi eliminado e todos os nossos testes continuam passando, passamos para o próximo mau cheiro: os comandos condicionais aninhados. À partir da inspeção do nosso código atual, vemos que nossos blocos if/else podem ser facilmente substituídos por cláusulas de guarda.

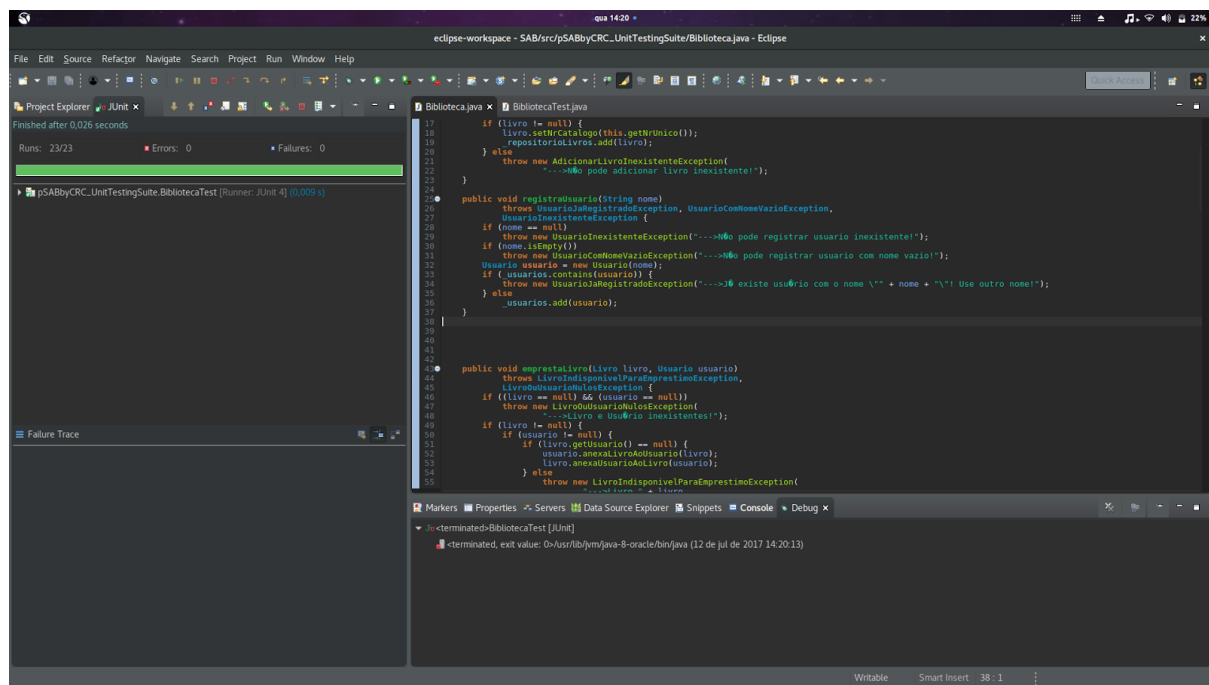
Aplicamos a técnica descrita acima no if mais externo:

```
public void registraUsuario(String nome)
    throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
    UsuarioInexistenteException {
    if (nome == null)
        throw new UsuarioInexistenteException("--->N o pode registrar usuario
inexistente!");
    if (nome.isEmpty()) {
        throw new UsuarioComNomeVazioException("--->N o pode registrar
usuario com nome vazio!");
    } else {
        Usuario usuario = new Usuario(nome);
        if (_usuarios.contains(usuario)) {
            throw new UsuarioJaRegistradoException("--->J  existe usu rio com
o nome \"" + nome + "\"! Use outro nome!");
        } else
            _usuarios.add(usuario);
    }
}
```



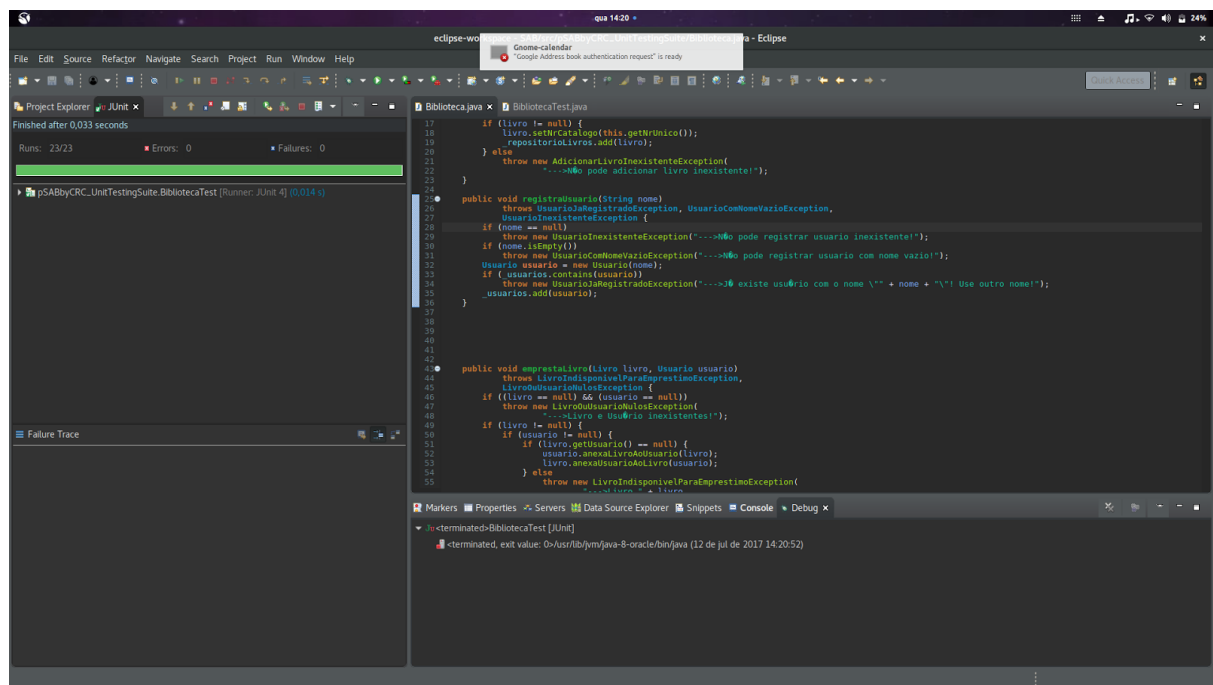
Rodamos os testes novamente, e todos passam. Seguimos para o segundo if:

```
public void registraUsuario(String nome)
    throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
    UsuarioInexistenteException {
    if (nome == null)
        throw new UsuarioInexistenteException("--->N o pode registrar usuario
inexistente!");
    if (nome.isEmpty())
        throw new UsuarioComNomeVazioException("--->N o pode registrar usuario
com nome vazio!");
    Usuario usuario = new Usuario(nome);
    if (_usuarios.contains(usuario)) {
        throw new UsuarioJaRegistradoException("--->J  existe usu rio com o nome
\"" + nome + "\"! Use outro nome!");
    } else
        _usuarios.add(usuario);
}
```



Rodamos os testes de novo e passamos para o terceiro if:

```
public void registraUsuario(String nome)
    throws UsuarioJaRegistradoException, UsuarioComNomeVazioException,
    UsuarioInexistenteException {
    if (nome == null)
        throw new UsuarioInexistenteException("--->Nao pode registrar usuario
inexistente!");
    if (nome.isEmpty())
        throw new UsuarioComNomeVazioException("--->Nao pode registrar usuario
com nome vazio!");
    Usuario usuario = new Usuario(nome);
    if (_usuarios.contains(usuario))
        throw new UsuarioJaRegistradoException("--->] existe usu*rio com o nome
\"\" + nome + \"\"! Use outro nome!");
    _usuarios.add(usuario);
}
```



Pode-se notar que, após a refatoração de todos os maus-cheiros encontrados, o código acima encontra-se mais limpo e fácil de ler, ainda mantendo todas as suas funcionalidades.