

RELATÓRIO CICLO DE REFATORAÇÃO

Lista de maus cheiros

- Expressões booleanas negativas (Complicated Boolean Expressions)
 - Tornar as expressões booleanas positivas (3 Ciclos)
- Ifs aninhados
 - Replace Nested Conditional with Guard Clauses (3 Ciclos)

1º Ciclo (Expressões booleanas negativas)

Código antes da refatoração:

```
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!");
}
```

Tipo Mau Cheiro/Técnica de Refatoração

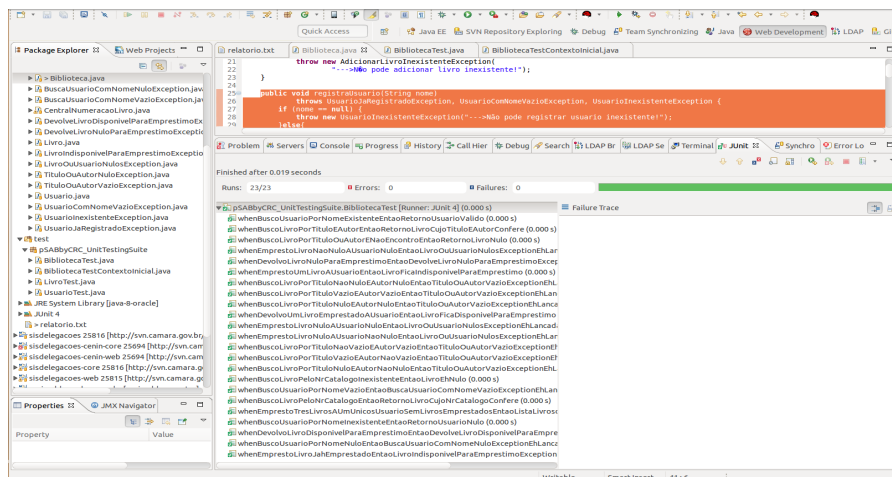
- Expressões booleanas negativas (Complicated Boolean Expressions)
 - Tornar as expressões booleanas positivas

Código Refatorado

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

    if (nome == null) {
        throw new UsuarioInexistenteException("---->Não pode registrar usuario inexistente!");
    } else {
        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!");
    }
}
```

Execução bem-sucedida



2º Ciclo (Expressões booleanas negativas)

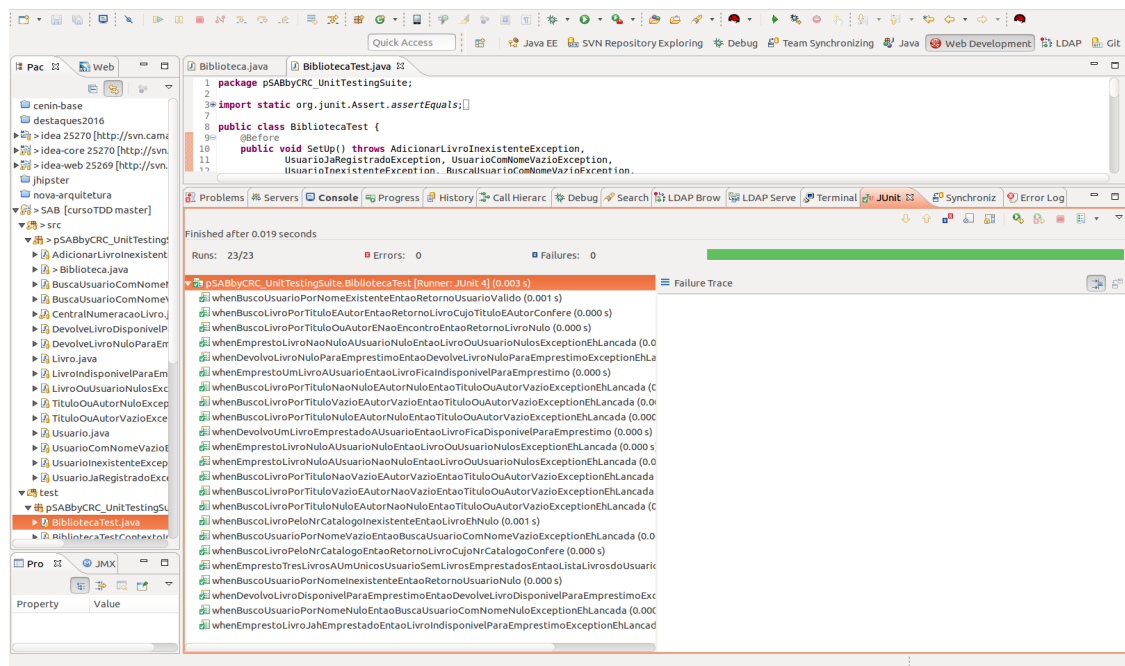
Código antes da refatoração:

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

Código Refatorado

```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {  
    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);  
            } else  
                throw new UsuarioJaRegistradoException(  
                    "---->Já existe usuário com o nome \"\" + nome + "\"! Use outro nome!");  
        }  
    }  
}
```

Execução bem-sucedida



3º Ciclo (Expressões booleanas negativas)

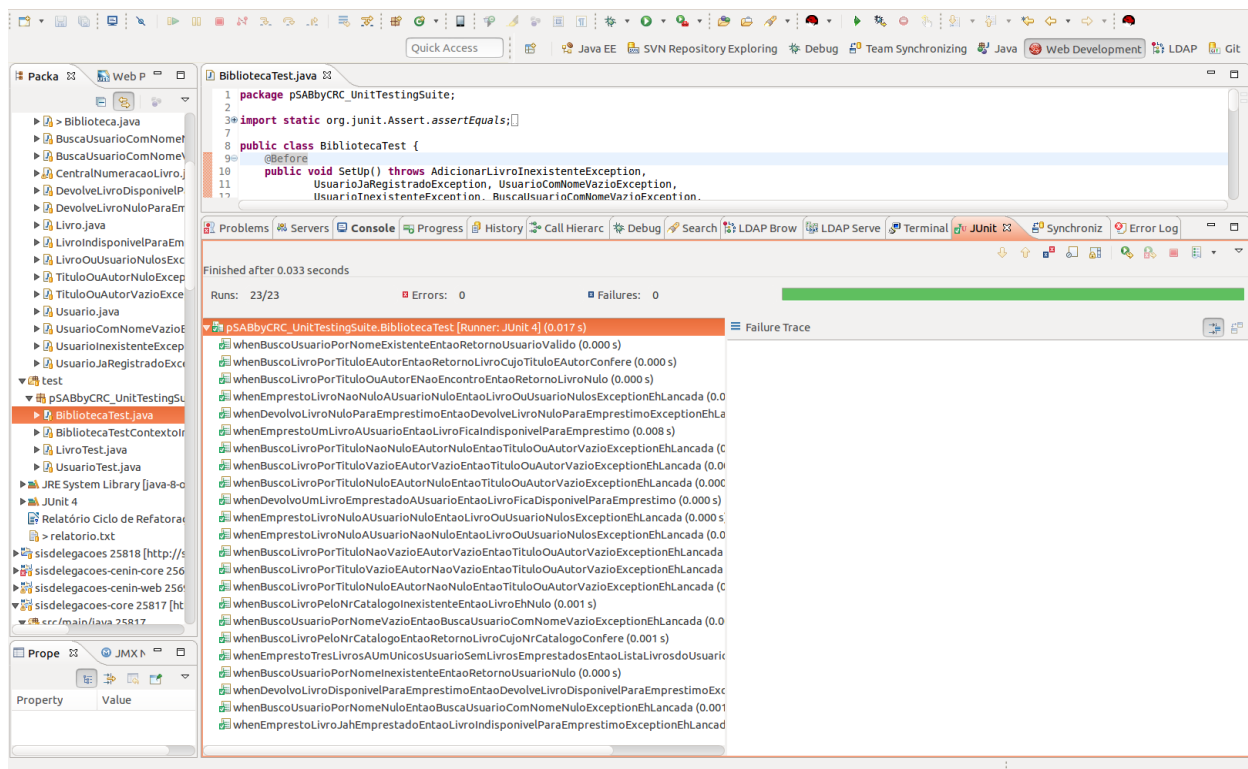
Código antes da refatoração:

```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
    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);
            } else
                throw new UsuarioJaRegistradoException(
                    "---->Já existe usuário com o nome \"" + nome + "\"! Use outro nome!");
        }
    }
}
```

Código Refatorado

```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
    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!");
            }else{
                _usuarios.add(usuario);
            }
        }
    }
}
```

Execução bem-sucedida



Lista de maus cheiros

- ~~Expressões booleanas negativas (Complicated Boolean Expressions)~~
 - ~~Tornar as expressões booleanas positivas (3 Ciclos)~~
- Ifs aninhados
 - Replace Nested Conditional with Guard Clauses (3 Ciclos)

1º Ciclo (Ifs aninhados)

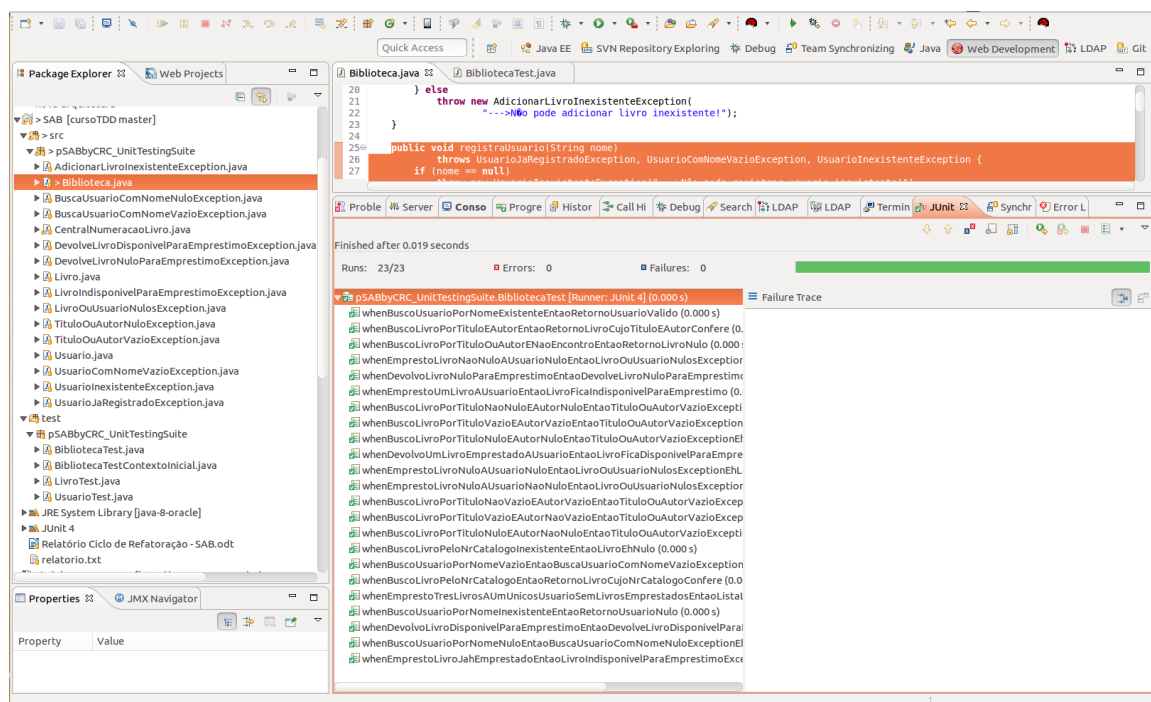
Código antes da refatoração:

```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {  
    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!");  
            } else {  
                _usuarios.add(usuario);  
            }  
        }  
    }  
}
```

Código Refatorado

```
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!");  
    } 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);  
        }  
    }  
}
```

Execução bem-sucedida



2º Ciclo (Ifs aninhados)

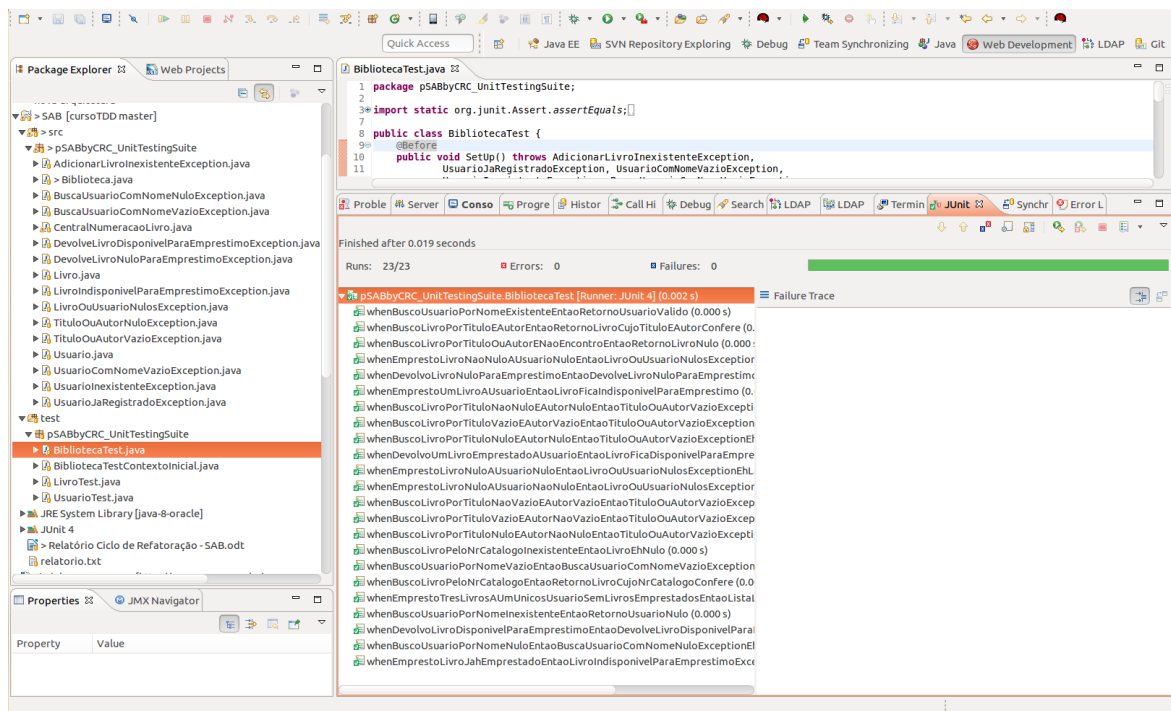
Código antes da refatoração:

```
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!");
    } 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);
        }
    }
}
```

Código Refatorado

```
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!");
    } else {
        _usuarios.add(usuario);
    }
}
```

Execução bem-sucedida



3º Ciclo (Ifs aninhados)

Código antes da refatoração:

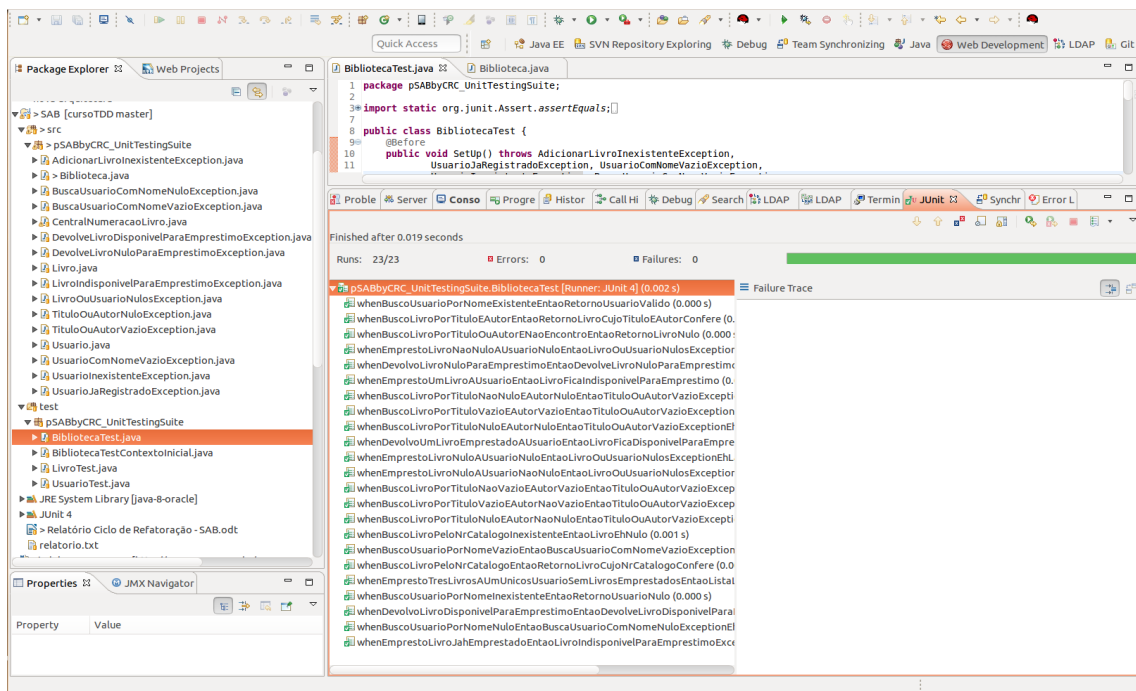
```
public void registraUsuario(String nome) throws UsuarioJaRegistradoException, UsuarioComNomeVazioException, UsuarioInexistenteException {
    if (nome == null) throw new UsuarioInexistenteException("---->Não pode registrar usuario inexistente!");
    if (nome.isEmpty()) throw new UsuarioComNomeVazioException("---->Não 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!");
    } else {
        _usuarios.add(usuario);
    }
}
```

Código Refatorado

```
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);
}
```

Execução bem-sucedida



Lista de maus cheiros

- ~~Expressões booleanas negativas (Complicated Boolean Expressions)~~
 - ~~Tornar as expressões booleanas positivas (3 Ciclos)~~
- Ifs aninhados
 - ~~Replace Nested Conditional with Guard Clauses (3 Ciclos)~~