

# **Testes Funcionais - Fase1**

Neste documento estão informados os casos de teste que serão realizados na primeira fase do projeto. As funcionalidades envolvidas nos testes mencionados abaixo são: Inserção de um item bibliográfico (BibTeX entry) para as categorias *BOOK* e *ARTICLE* e importação de itens bibliográficos na base corrente. Vale lembrar que os testes aqui feitos são relacionados à testes funcionais, uma vez que estamos verificando a correta funcionalidade das funções, de acordo com os requisitos estudados sobre o software JabRef.

## **1. Teste Funcionais**

Nos casos de teste funcionais realizados em 1.1 e 1.2 a seguir, utilizamos a Análise do Valor Limite. A utilizamos nestes testes porque, com ela, poderíamos analisar todos os valores em todas as partições possíveis, de uma forma mais rápida e eficiente, pois veríamos apenas um valor de cada partição nos testes. Se este valor desse algo diferente do esperado, saberíamos que todos os valores dessa partição gerariam erros. É importante informar que, nesta primeira fase, sabíamos que, basicamente, o software não faz nenhuma validação de dados inseridos. Por isso, o esperado é que nenhum erro ocorra.

### **1.1- Inserção de um item bibliográfico em Book**

1.1.1 - Inserção de nenhum campo. Saída esperada: Inserção do item bibliográfico, sem erros;

1.1.2 - Inserção de apenas um campo. Saída esperada: Inserção do item bibliográfico, sem erros;

1.1.3 - Inserção de todos os campos. Saída esperada: Inserção do item bibliográfico, sem erros.

### **1.2- Inserção de um item bibliográfico em Article**

1.2.1 - Inserção de nenhum campo. Saída esperada: Inserção do item bibliográfico, sem erros;

1.2.2 - Inserção de apenas um campo. Saída esperada: Inserção do item bibliográfico, sem erros;

1.2.3 - Inserção de todos os campos. Saída esperada: Inserção do item bibliográfico, sem erros;

### **1.3 - Importação de itens bibliográficos na base corrente**

Foi-se criados arquivos, um com extensão .bib e outro .ris, nos quais seriam efetuado os testes de importação. Basicamente, foram testados as inserções dos atributos definidos, não sendo testados os atributos opcionais, na importação do arquivo. Mais uma vez, o teste ocorreu sem problemas, como esperado, ocorrendo a importação do arquivo, juntamente com seus atributos, sem nenhuma verificação e nenhum erro gerado, inclusive nos atributos, que foram importados corretamente. Utilizamos este tipo de teste, pois queríamos ver apenas o funcionamento da importação com os atributos.

## **2. Notas**

A implementação dos casos de teste para esta fase foram inseridas em:

### **Teste 1.1:**

<https://github.com/rophos-rb/DC-UFSCar-ES2-201601--Grupo-BBIJR/blob/master/jabref-master/src/test/java/net/sf/jabref/importer/fileformat/CasodeTesteBook.java>

### **Teste 1.2:**

<https://github.com/rophos-rb/DC-UFSCar-ES2-201601--Grupo-BBIJR/blob/master/jabref-master/src/test/java/net/sf/jabref/importer/fileformat/CasodeTesteArticle.java>

### **Teste 1.3:**

<https://github.com/rophos-rb/DC-UFSCar-ES2-201601--Grupo-BBIJR/blob/master/jabref-master/src/test/java/net/sf/jabref/importer/fileformat/TesteEntradaBibTest.java>

## arquivo gerado:

<https://github.com/rophos-rb/DC-UFSCar-ES2-201601--Grupo-BBIJR/blob/master/jabref-master/src/test/resources/net/sf/jabref/importer/fileformat/TesteEntradaBib.bib>

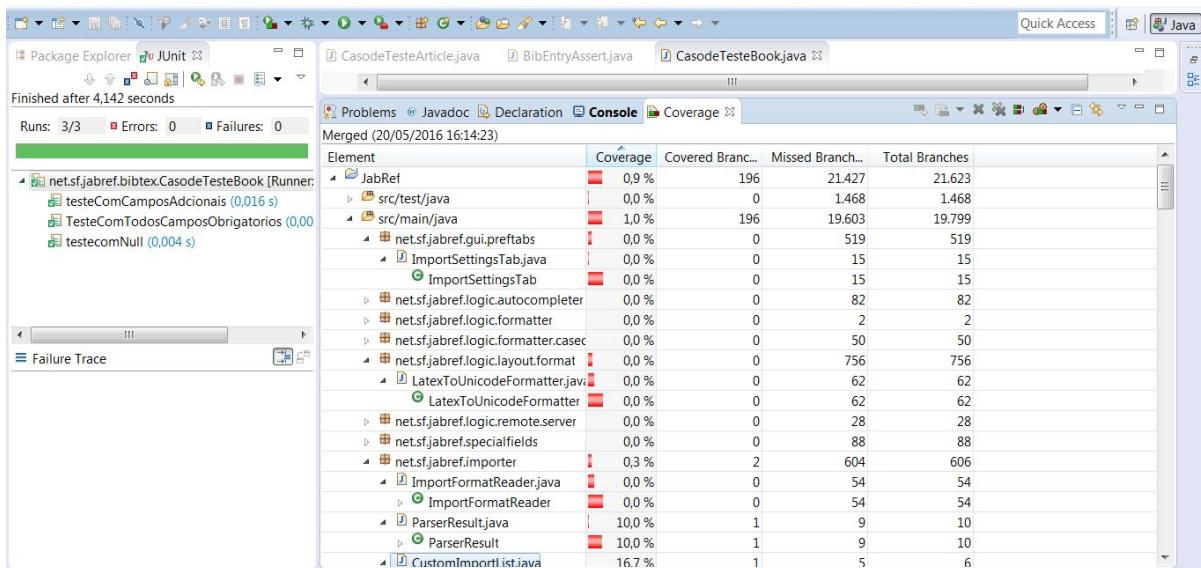
Clique ou copie e cole no seu navegador os links acima para ver os código implementados para cada teste.

## 3. Resultados

A seguir, são apresentados os resultados obtidos pelos testes feitos e declarados acima. Os resultados apresentados são as coberturas dos testes sobre os códigos, feita pelo software Eclemma no Eclipse.

### 3.1 Cobertura de *Branches* em Book

Abaixo, as imagens mostram a cobertura de *branches* de nossos casos de teste em book.



Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

net.sf.jabref.bibtex.CasodeTesteBook [Runner]

- testeComCamposAdcionais (0,016 s)
- TesteComTodosCamposObrigatorios (0,00 s)
- testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

Merged (20/05/2016 16:14:23)

Element	Coverage	Covered Branches	Missed Branches	Total Branches
CustomImportList	16,7 %	1	5	6
readPrefs()	50,0 %	1	1	2
CustomImportList(JabRe		0	0	0
net.sf.jabref.logic.cleanup	0,6 %	1	163	164
FieldFormatterCleanup.java	0,0 %	0	18	18
FieldFormatterCleanup	0,0 %	0	18	18
FieldFormatterCleanup(		0	0	0
getFormat()		0	0	0
getFormatter()		0	0	0
CleanupPreset.java	4,2 %	1	23	24
CleanupPreset	4,2 %	1	23	24
isRenamePDF()	25,0 %	1	3	4
CleanupStep		0	0	0
CleanupPreset(Set<Clea		0	0	0
getFormatterCleanups()		0	0	0
isActive(CleanupStep)		0	0	0
isCleanUpDOI()		0	0	0
isCleanUpSuperscripts()		0	0	0
isCleanUpUpgradeExter		0	0	0
isConvertToBiblate()		0	0	0

Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

net.sf.jabref.bibtex.CasodeTesteBook [Runner]

- testeComCamposAdcionais (0,016 s)
- TesteComTodosCamposObrigatorios (0,00 s)
- testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

Merged (20/05/2016 16:14:23)

Element	Coverage	Covered Branches	Missed Branches	Total Branches
isConvertToBiblate()		0	0	0
isConvertUnicodeToLate		0	0	0
isFixFileLinks()		0	0	0
isMakePathsRelative()		0	0	0
isMovePDF()		0	0	0
isRenamePdfOnlyRelativ		0	0	0
net.sf.jabref.exporter	1,1 %	8	693	701
CustomExportListJava	0,0 %	0	14	14
CustomExportList	0,0 %	0	14	14
FieldFormatterCleanups.java	18,2 %	8	36	44
FieldFormatterCleanups	18,2 %	8	36	44
convertToString()	50,0 %	1	1	2
getMetaDataTable(List)	87,5 %	7	1	8
FieldFormatterCleanups		0	0	0
ExportComparator.java		0	0	0
net.sf.jabref.gui.desktop	1,4 %	1	71	72
net.sf.jabref.sql	2,1 %	3	139	142
DatabaseType.java	75,0 %	3	1	4
DatabaseType	75,0 %	3	1	4
build(String)	75,0 %	3	1	4

Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

net.sf.jabref.bibtex.CasodeTesteBook [Runner]

- testeComCamposAdcionais (0,016 s)
- TesteComTodosCamposObrigatorios (0,00 s)
- testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

Merged (20/05/2016 16:14:23)

Element	Coverage	Covered Branches	Missed Branches	Total Branches
getFormattedMessage()		0	0	0
DBStrings.java		0	0	0
DBStrings		0	0	0
DBStrings()		0	0	0
isConfigValid(boolean)		0	0	0
isInitialized(boolean)		0	0	0
setPassword(String)		0	0	0
DBStringsPreferences.java		0	0	0
DBStringsPreferences		0	0	0
DBStringsPreferencesSt		0	0	0
net.sf.jabref.model.entry	2,3 %	13	552	565
BibEntry.java	14,4 %	13	77	90
BibEntry	14,8 %	13	75	88
clearField(String)	50,0 %	1	1	2
equals(Object)	50,0 %	5	5	10
getAuthorTitleYear(int)	50,0 %	2	2	4
setType(String)	50,0 %	2	2	4
setField(String, String)	75,0 %	3	1	4
BibEntry()		0	0	0
RibEntry(String)		0	0	0

Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

- net.sf.jabref.bibtex.CasodeTesteBook [Runner]
  - testeComCamposAdcionais (0,016 s)
  - TesteComTodosCamposObrigatorios (0,00 s)
  - testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

Merged (20/05/2016 16:14:23)

Element	Coverage	Covered Branches	Missed Branches	Total Branches
BibEntry(String, String)	0	0	0	0
firePropertyChangedEvent	0	0	0	0
getciteKey()	0	0	0	0
getFieldOptional(String)	0	0	0	0
getId()	0	0	0	0
hasField(String)	0	0	0	0
setciteKey(String)	0	0	0	0
setParsedSerialization(Serialization)	0	0	0	0
toLowerCase(String)	0	0	0	0
IdGenerator.java	0	0	0	0
IdGenerator	0	0	0	0
next()	0	0	0	0
net.sf.jabref.logic.util	3,2 %	2	60	62
net.sf.jabref.importer.fileformat	3,8 %	83	2.101	2.184
BibtexParser.java	32,5 %	82	170	252
BibtexParser	32,5 %	82	170	252
consume(char, char)	25,0 %	1	3	4
purgeEOFCharacters(String)	25,0 %	1	3	4
skipSpace()	25,0 %	1	3	4
consumeUncritically(char, char)	30,0 %	3	7	10

Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

- net.sf.jabref.bibtex.CasodeTesteBook [Runner]
  - testeComCamposAdcionais (0,016 s)
  - TesteComTodosCamposObrigatorios (0,00 s)
  - testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

Merged (20/05/2016 16:14:23)

Element	Coverage	Covered Branches	Missed Branches	Total Branches
parseField(BibEntry)	30,0 %	3	7	10
dumpTextReadsofarTo	33,3 %	4	8	12
parseFieldContent(String)	44,4 %	8	10	18
parseKey()	44,4 %	8	10	18
consume(char)	50,0 %	1	1	2
parse()	50,0 %	1	1	2
skipOneNewline()	50,0 %	2	2	4
parseBracketedTextExact	56,2 %	9	7	16
parseFileContent()	60,0 %	6	4	10
parseEntry(String)	64,3 %	9	5	14
parseAndAddEntry(String)	66,7 %	4	2	6
parseTextToken()	66,7 %	4	2	6
unread(int)	75,0 %	3	1	4
getPureTextFromFile()	100,0 %	2	0	2
isEOFCharacter(int)	100,0 %	4	0	4
read()	100,0 %	4	0	4
skipWhitespace()	100,0 %	4	0	4
fromString(String)	0	0	0	0
BibtexParser(Reader)	0	0	0	0
initializeParserResult()	0	0	0	0

Package Explorer JUnit

Finished after 4,142 seconds

Runs: 3/3 Errors: 0 Failures: 0

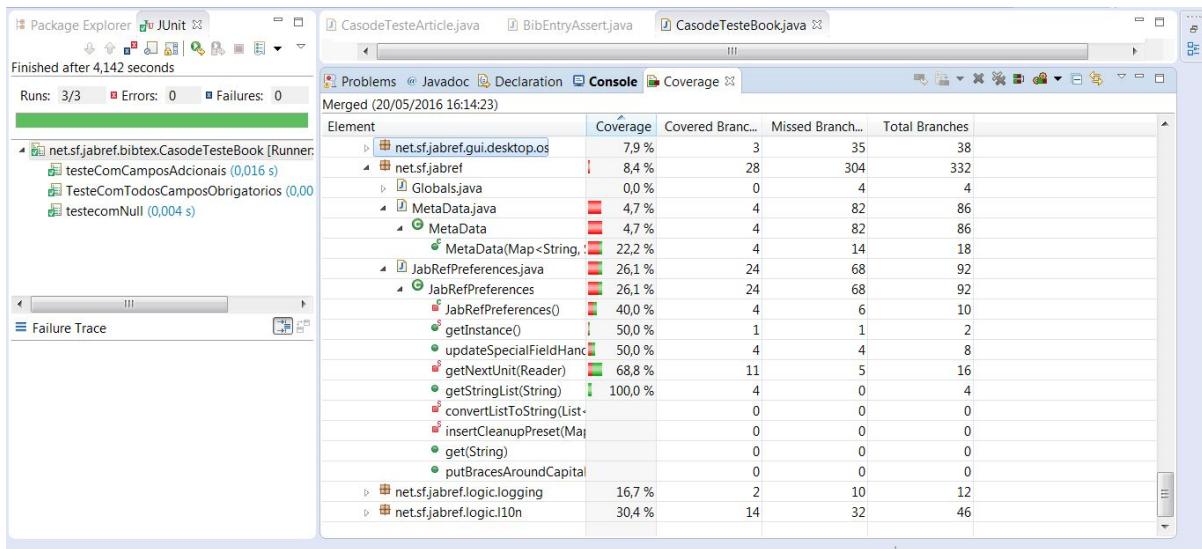
- net.sf.jabref.bibtex.CasodeTesteBook [Runner]
  - testeComCamposAdcionais (0,016 s)
  - TesteComTodosCamposObrigatorios (0,00 s)
  - testecomNull (0,004 s)

Failure Trace

Problems Javadoc Declaration Console Coverage

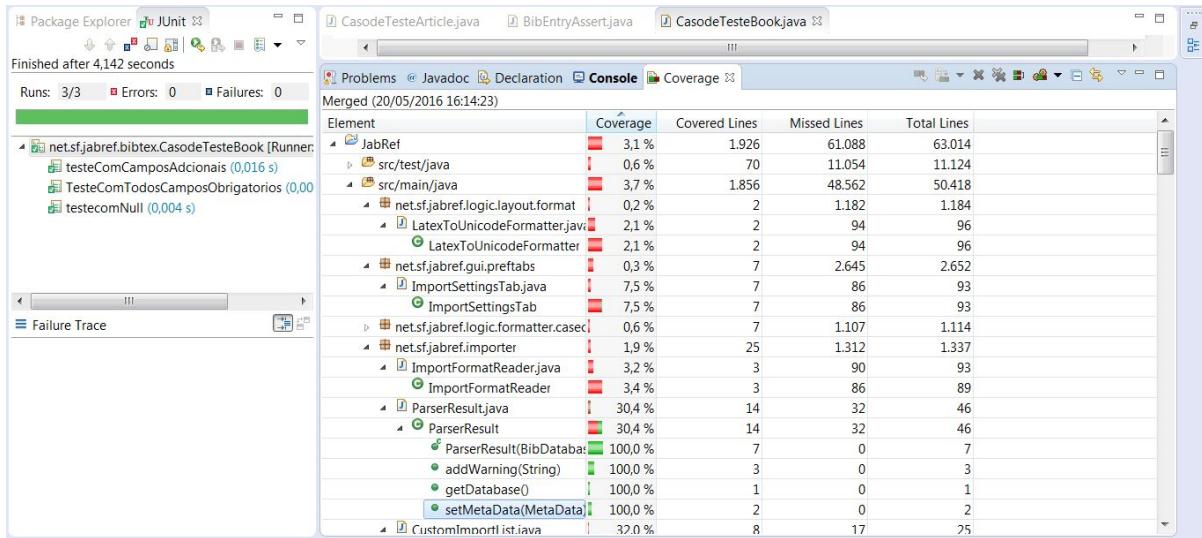
Merged (20/05/2016 16:14:23)

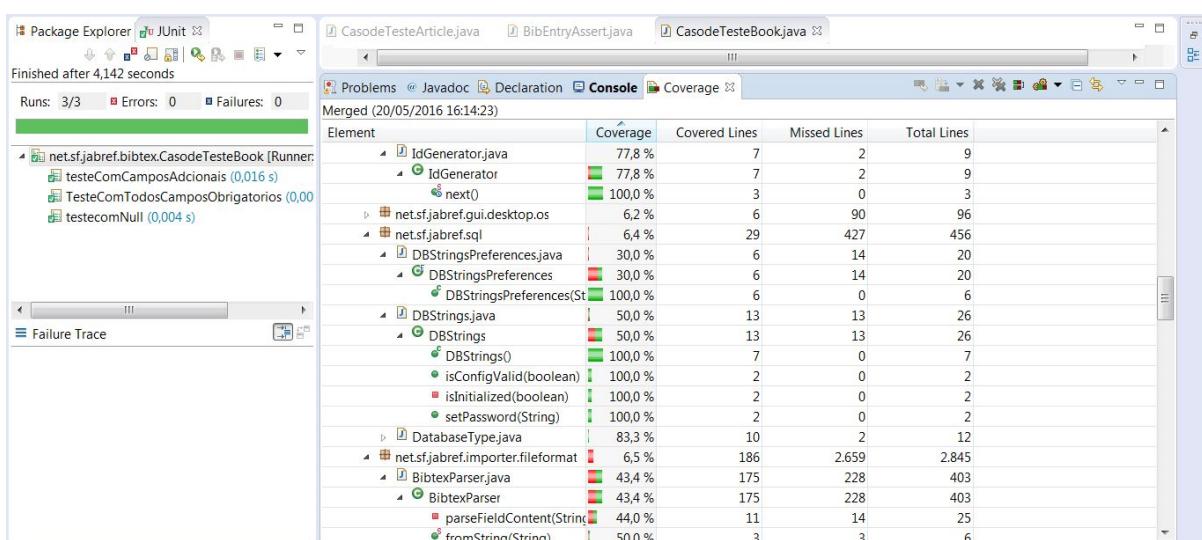
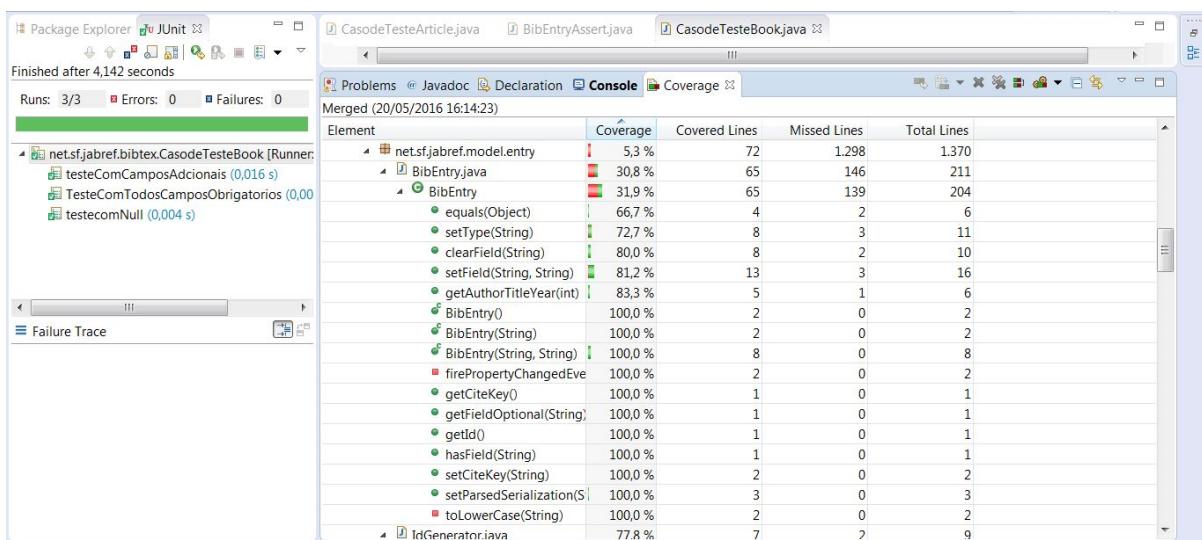
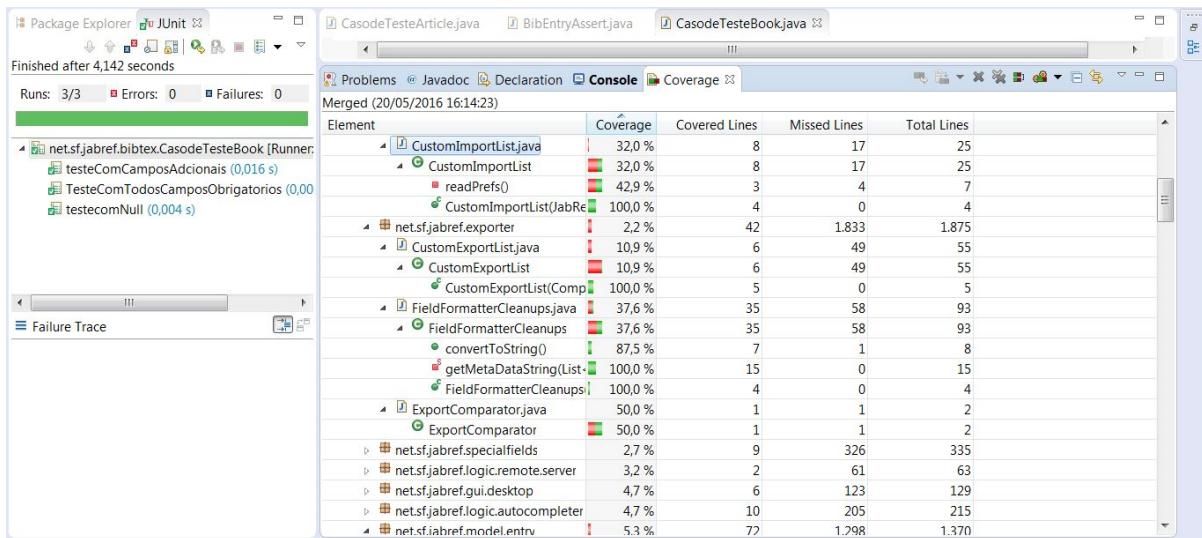
Element	Coverage	Covered Branches	Missed Branches	Total Branches
parseRemainingContent	0	0	0	0
peek()	0	0	0	0
FieldContentParser.java	50,0 %	1	1	2
FieldContentParser	50,0 %	1	1	2
format(String, String)	50,0 %	1	1	2
FieldContentParser()	0	0	0	0
format(StringBuilder, String)	0	0	0	0
net.sf.jabref.logic.formatter.bibte	4,8 %	4	80	84
net.sf.jabref.model.database	5,9 %	8	128	136
net.sf.jabref.logic.util.strings	7,6 %	24	290	314
StringUtil.java	2,6 %	7	263	270
StringUtil	2,6 %	7	263	270
quote(String, String, char)	58,3 %	7	5	12
HTMLUnicodeConversionMap	85,0 %	17	3	20
HTMLUnicodeConversionM	85,0 %	17	3	20
cleanLaTeX(String)	0	0	0	0
UnicodeToReadableCharMap	0	0	0	0
UnicodeToReadableCharM	0	0	0	0
UnicodeToReadableCha	0	0	0	0
net.sf.jabref.nui.desktop.os	7,9 %	3	35	38

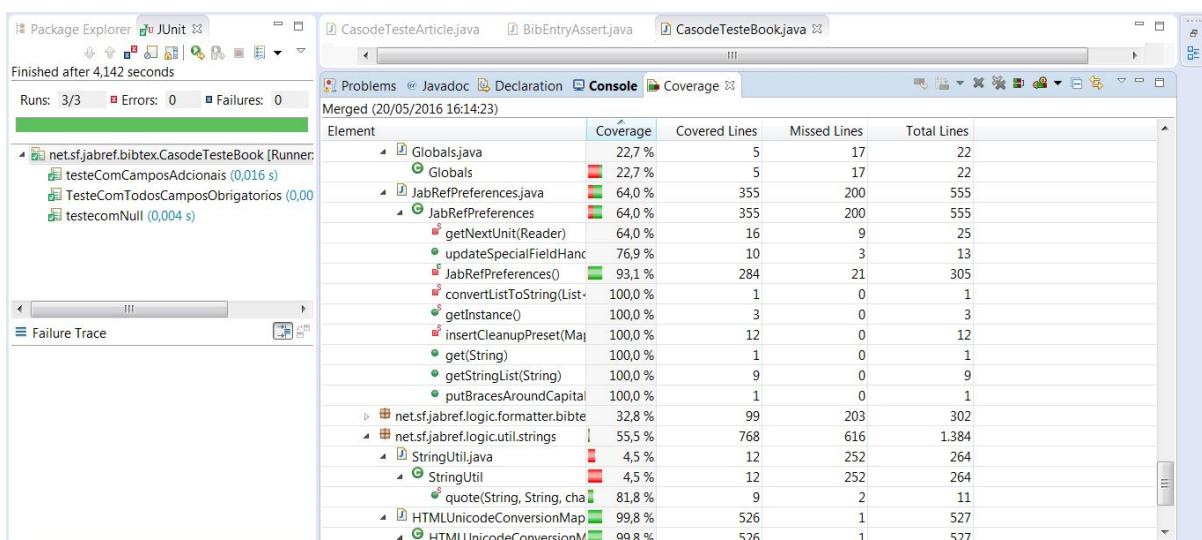
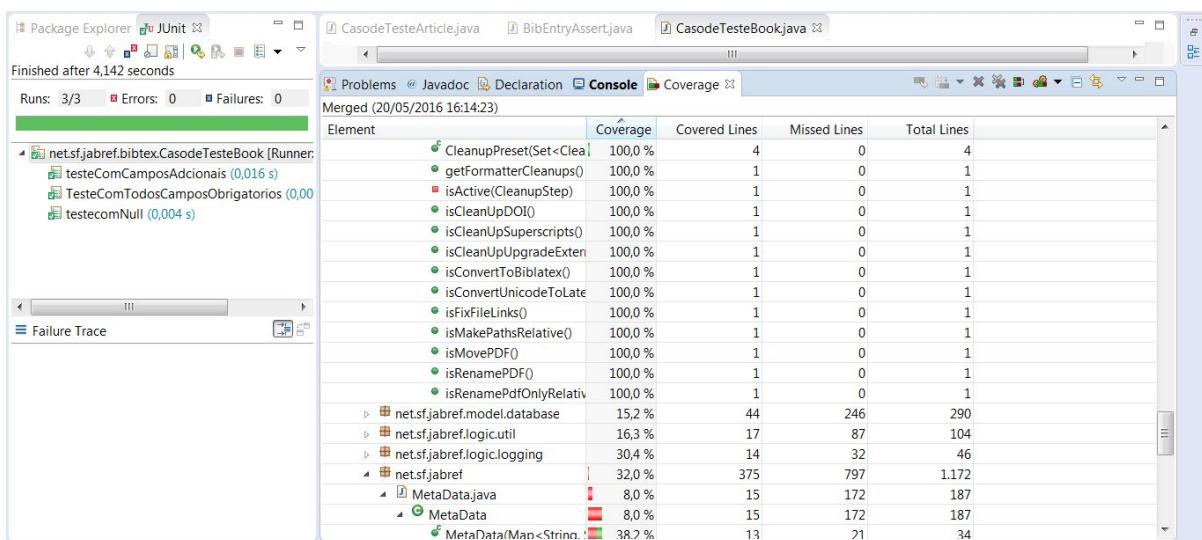
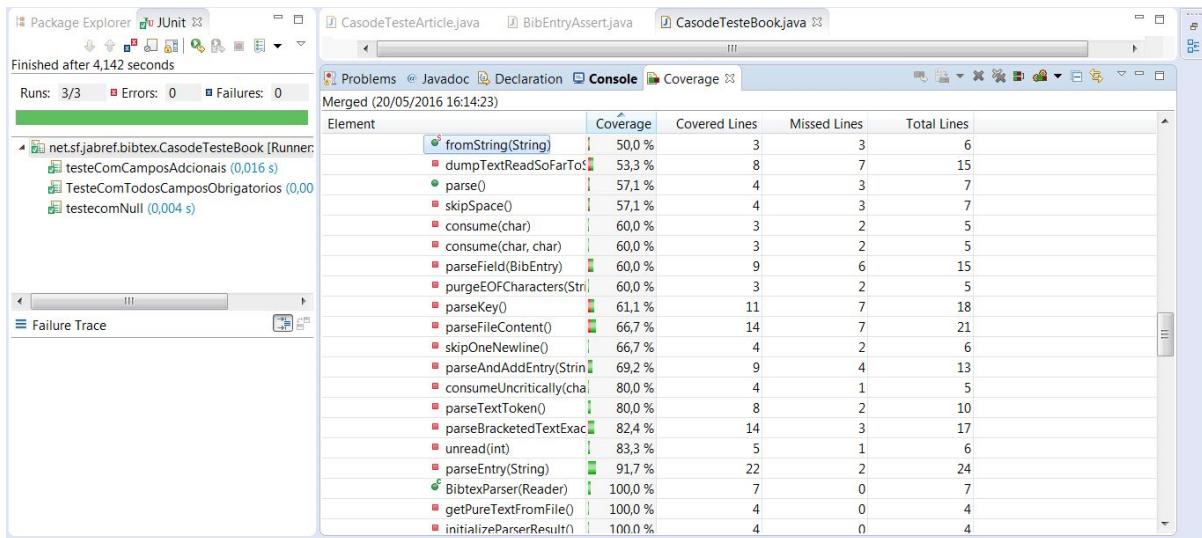


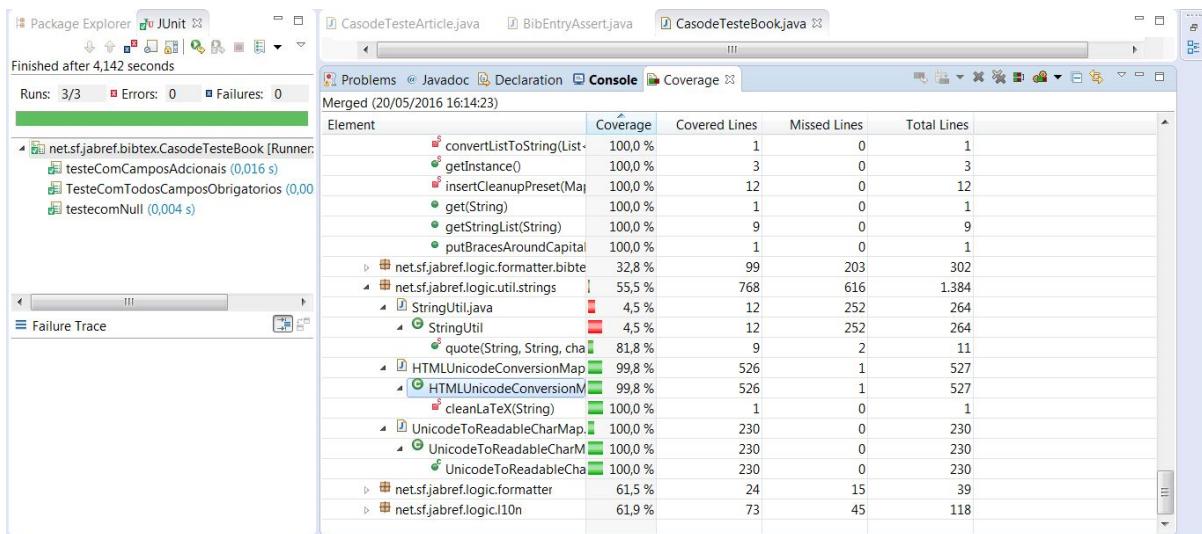
### 3.2 Cobertura de linhas sobre o Book

A seguir, as imagens mostram a cobertura de *lines* de nossos casos de teste em *book*.



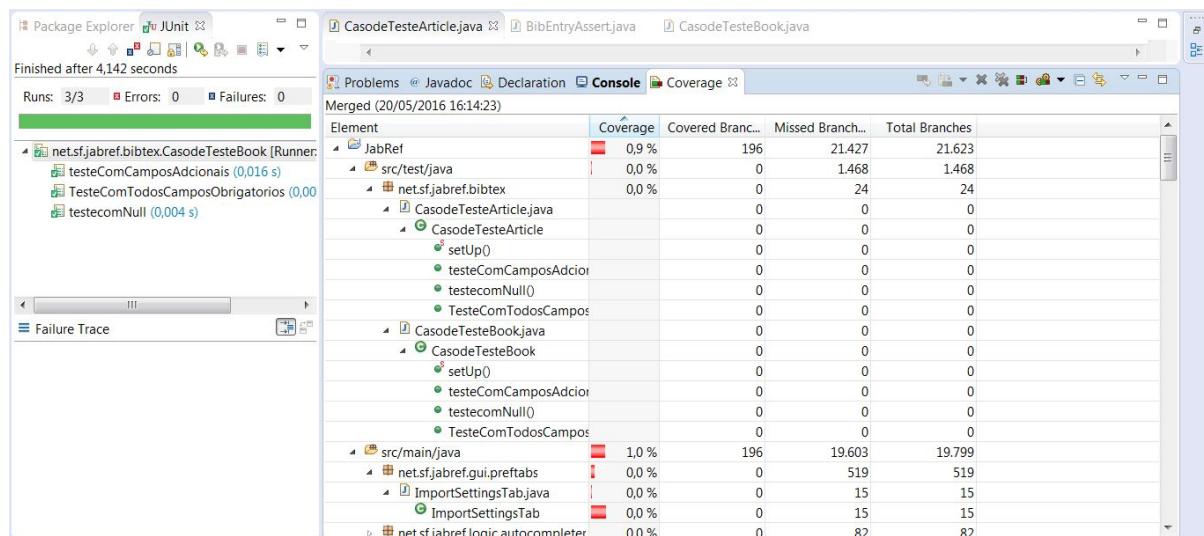


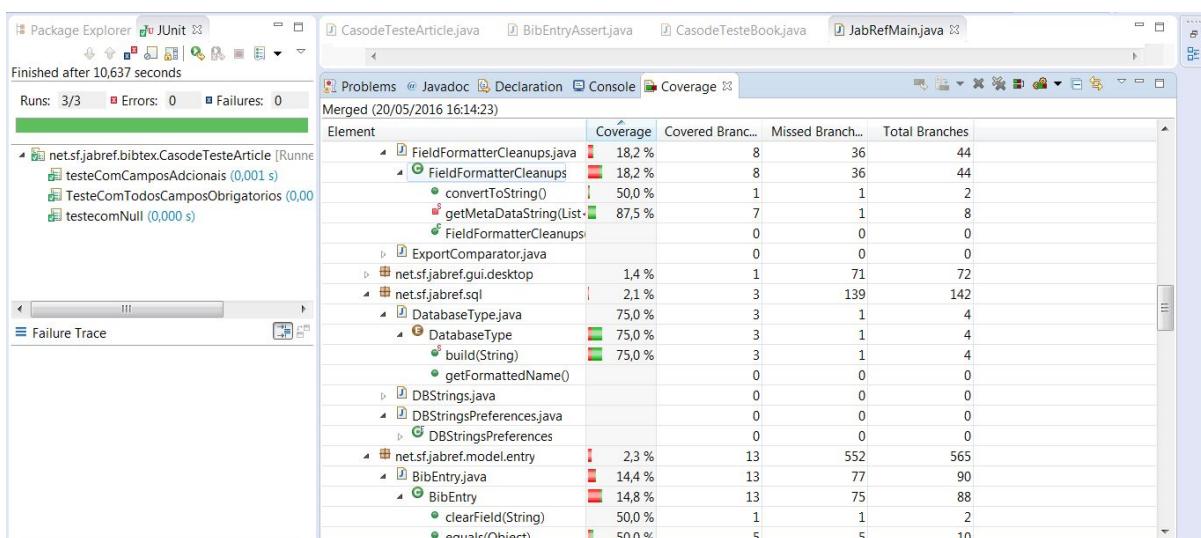
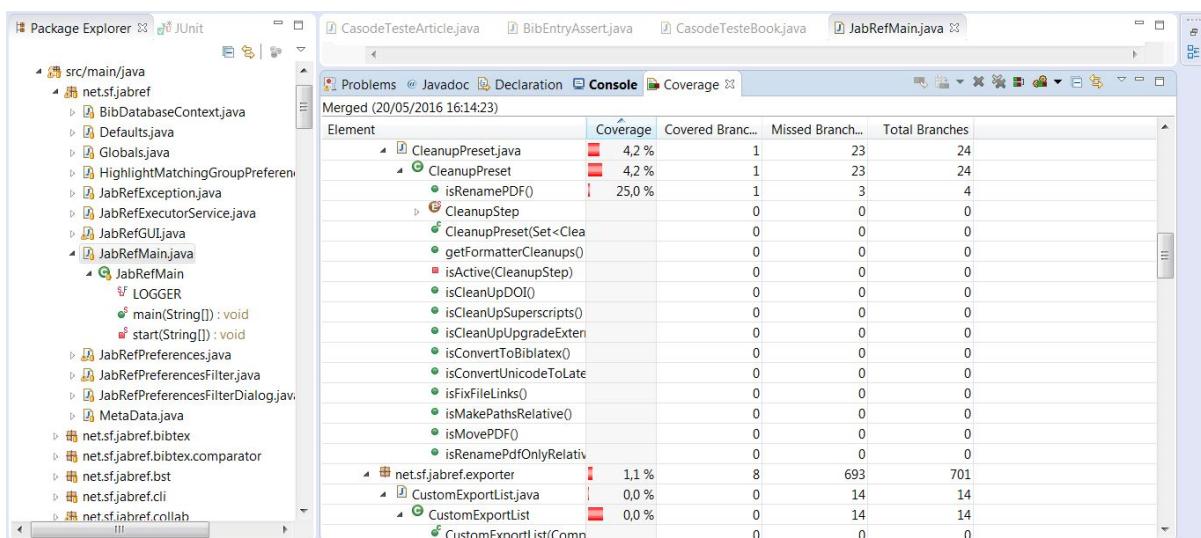
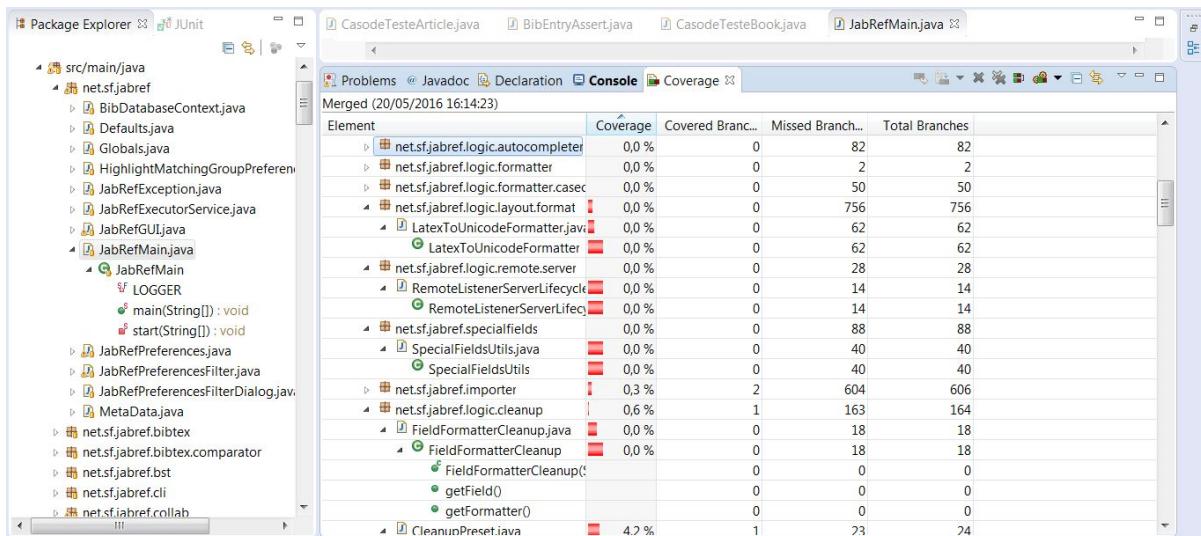


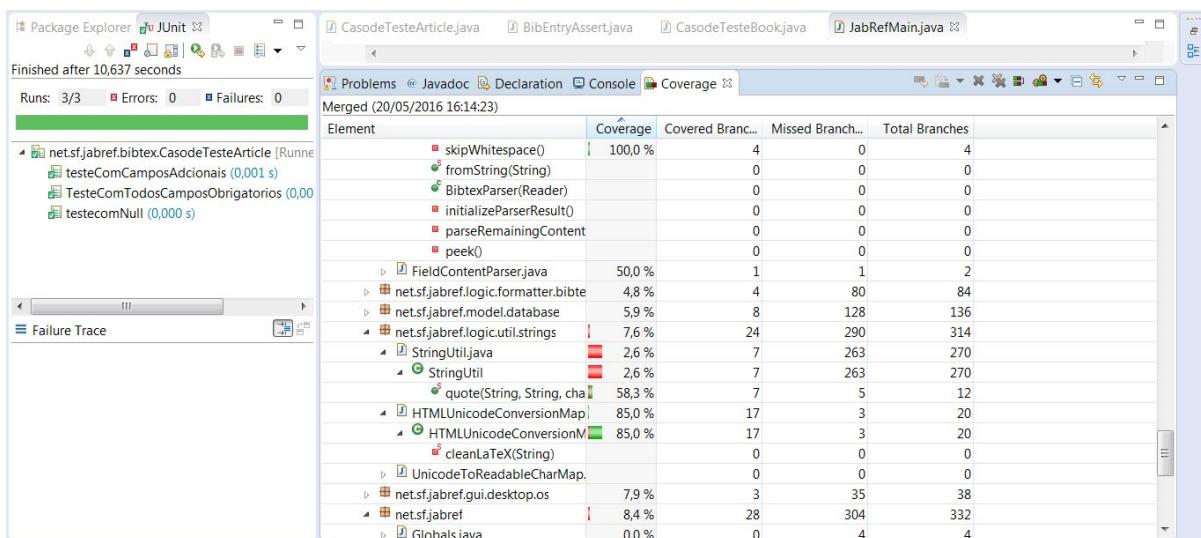
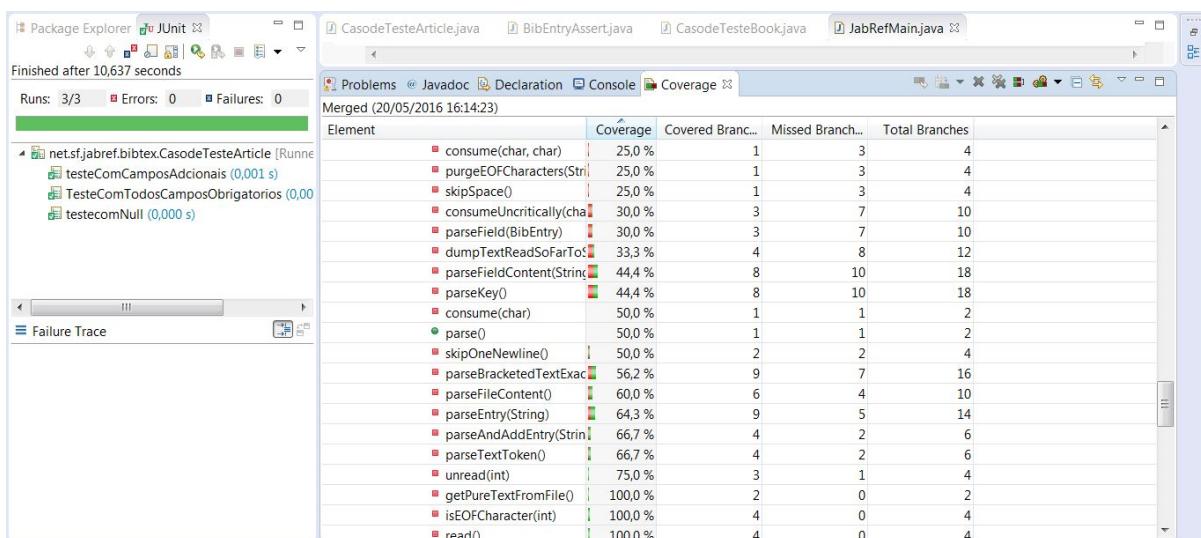
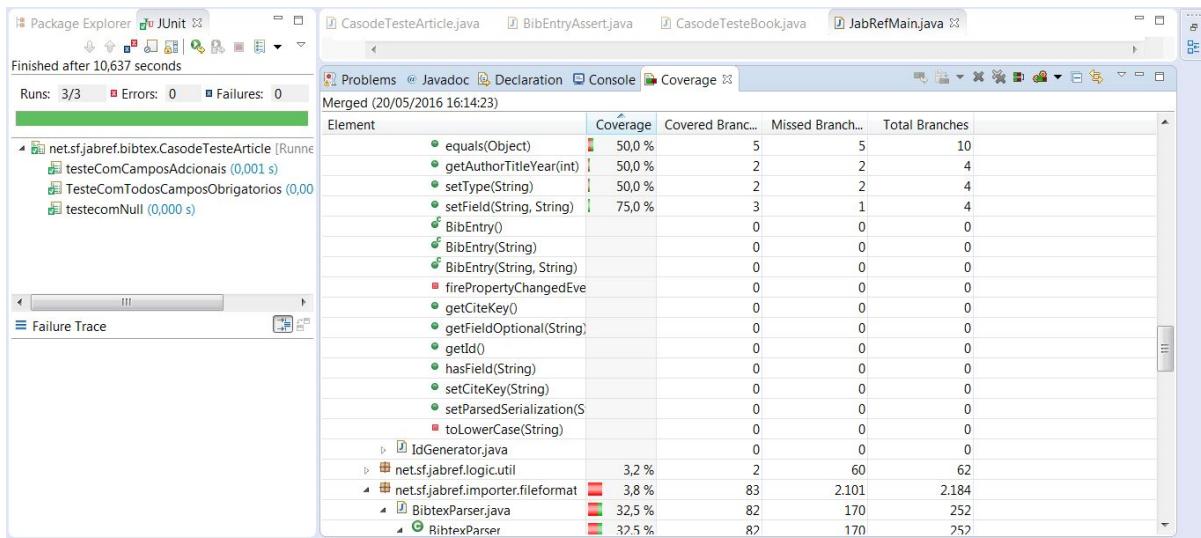


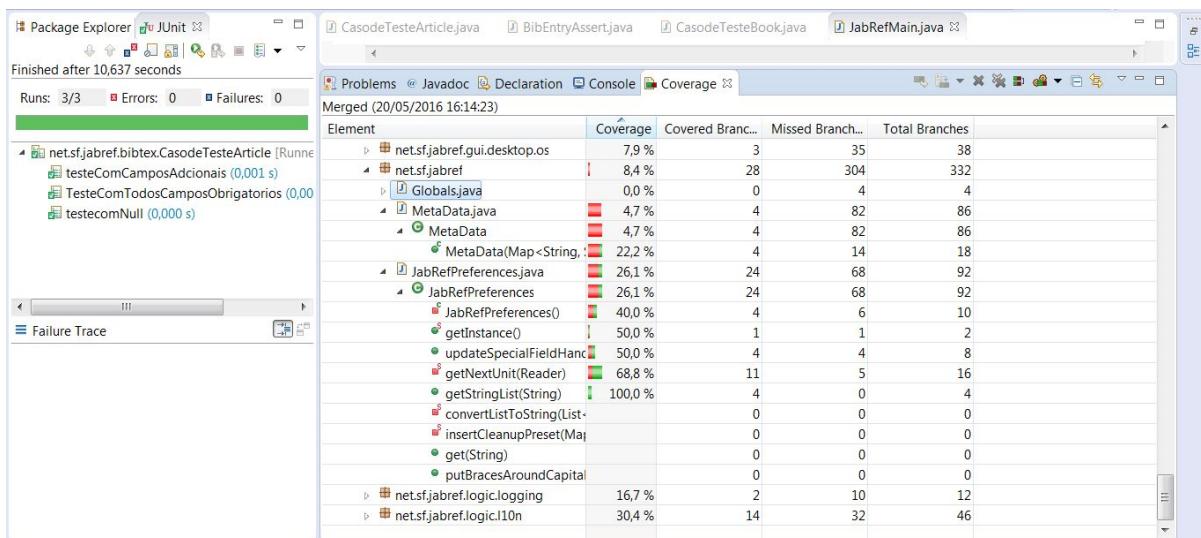
### 3.3 Cobertura de *Branches* em Article

A seguir, as imagens mostram a cobertura de *branches* de nossos casos de teste em *article*.



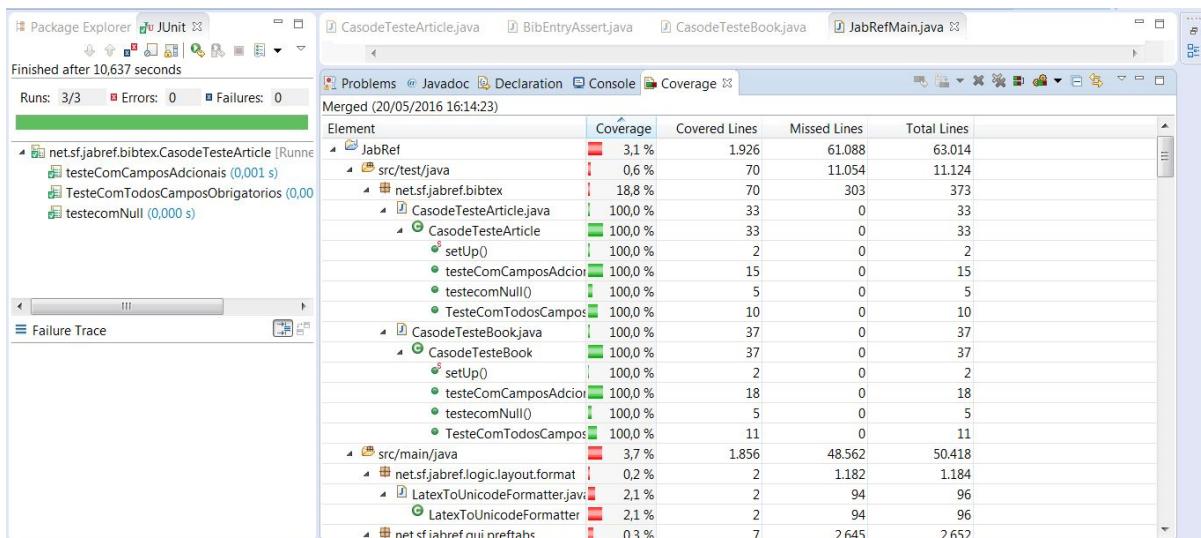


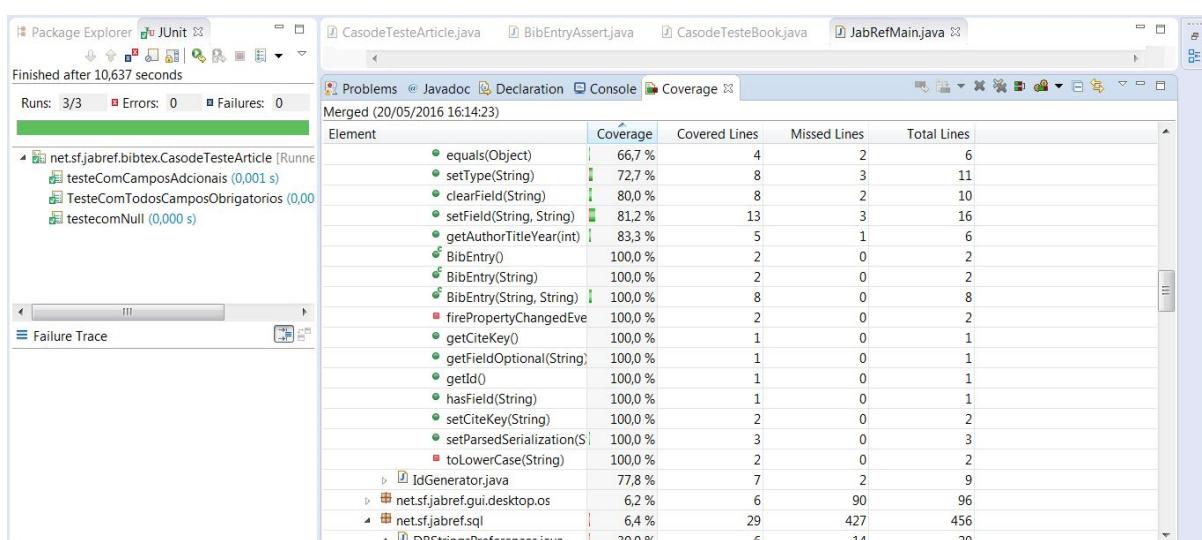
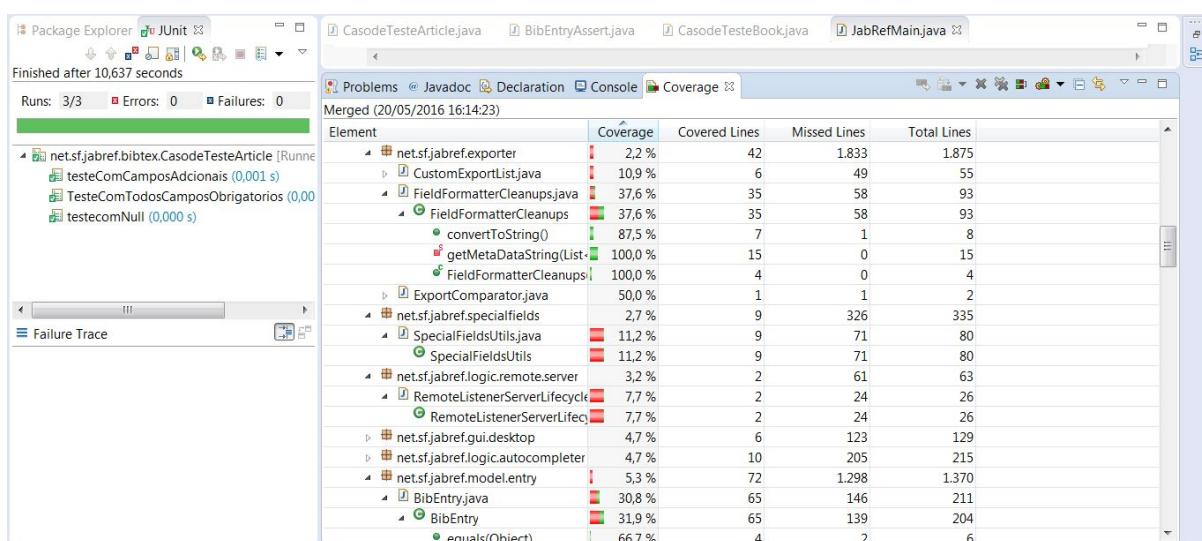
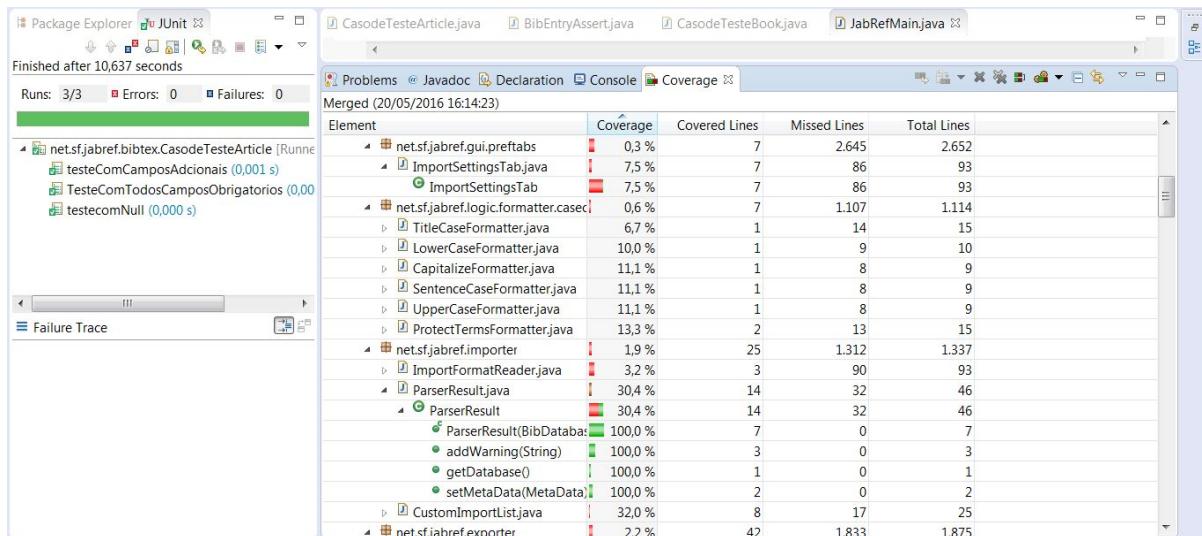


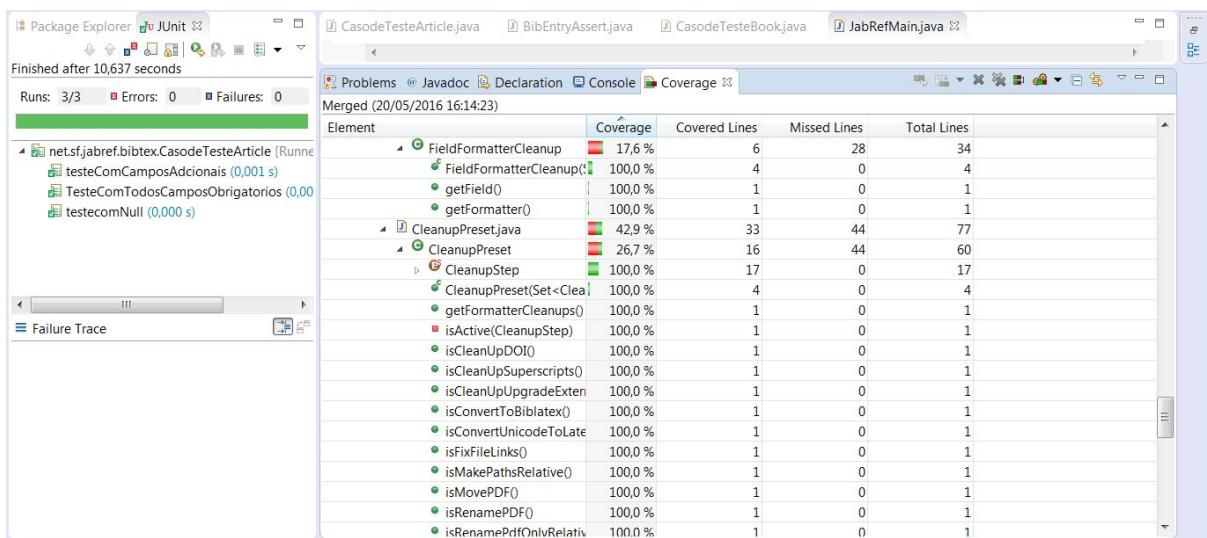
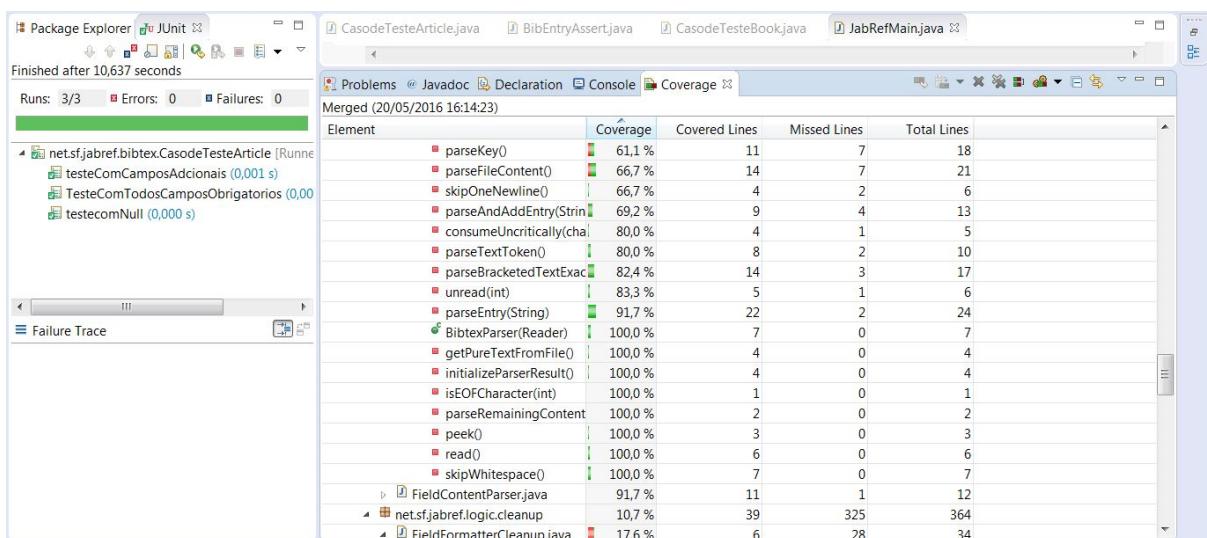
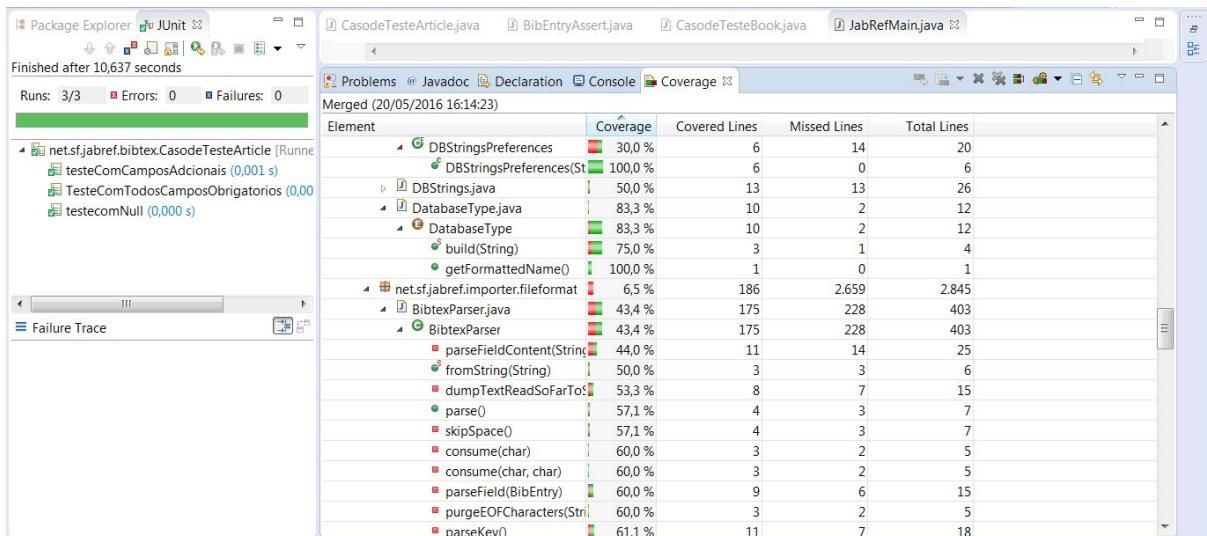


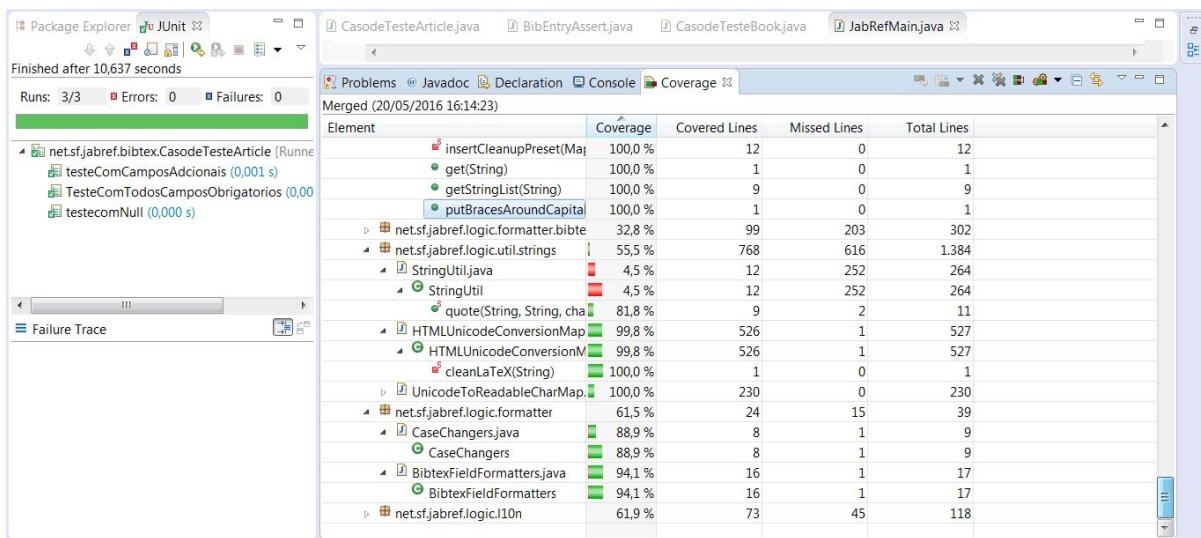
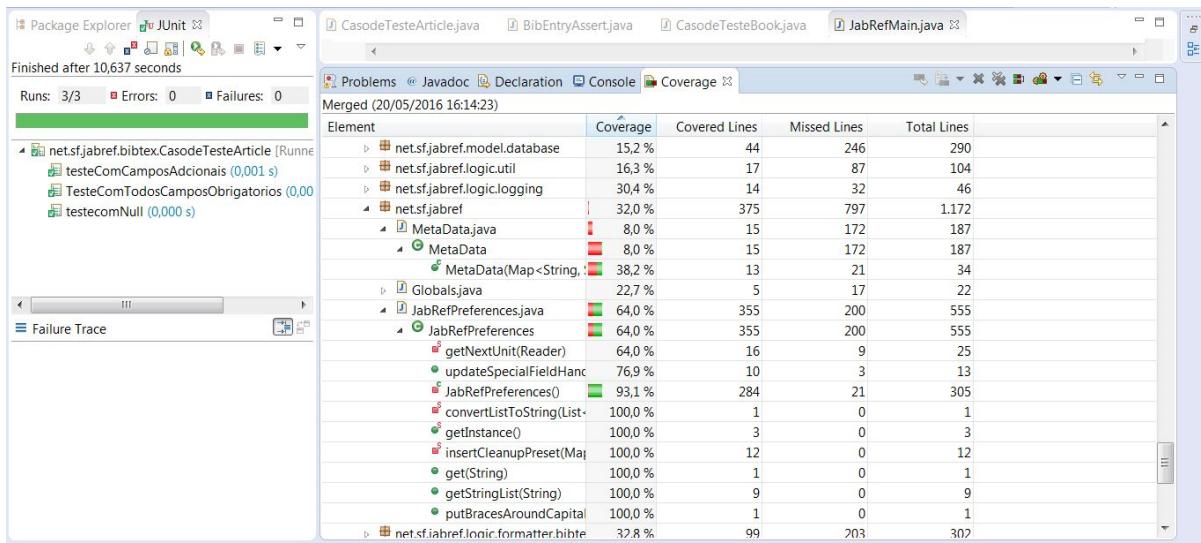
### 3.4 Cobertura de linhas em Article

A seguir, as imagens mostram a cobertura de *lines* de nossos casos de teste em *article*.



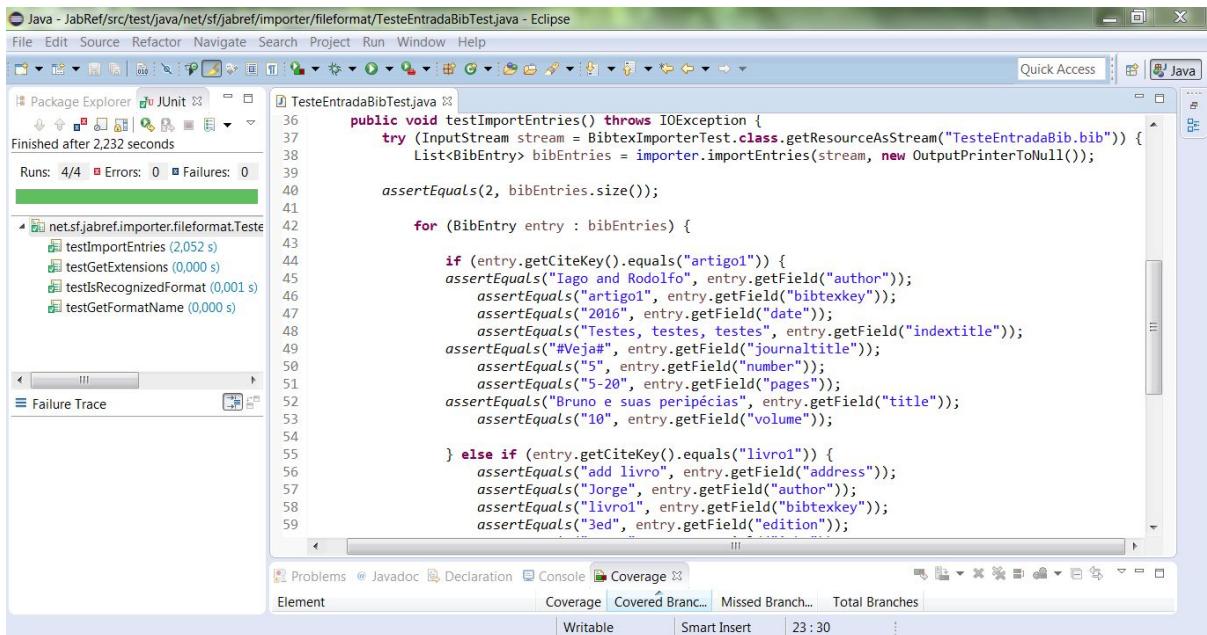






Observe que, como fizemos apenas casos de teste para *Book* e *Article*, muito do programa não foi testado. Mas, em contrapartida, as funções necessárias para criar um novo *Book* ou *Article* foram bem testadas e não houve erros, apenas de haver situações inesperadas, principalmente relacionado à não verificação de dados inseridos.

### 3.5 Teste em EntradaBib



The screenshot shows the Eclipse IDE interface with the following details:

- Java - JabRef/src/test/java/net/sf/jabref/importer/fileformat/TesteEntradaBibTest.java - Eclipse**: The active Java file.
- File Edit Source Refactor Navigate Search Project Run Window Help**: The menu bar.
- Quick Access**: A search bar at the top right.
- Package Explorer**: Shows the project structure with a green checkmark indicating successful build.
- JUnit**: Shows the test results:
  - Finished after 2,232 seconds
  - Runs: 4/4 Errors: 0 Failures: 0
- TesteEntradaBibTest.java**: The code for the test class, showing the implementation of the `testImportEntries` method.
- Failure Trace**: An empty section.
- Coverage**: Coverage statistics:
  - Element: Coverage
  - Total Branches: 23 : 30
  - Writable: Smart Insert

Os resultados dos testes de importação de arquivo mostram a correta importação e correta cópia dos dados para o arquivo ao importá-lo. Não há erros.