
Corp

PBTest

Manual do Sistema

Versão 1.0

Informações do Documento

Título do documento		Documento de Requisitos do BPTest	
Autores		Matheus Ale da Silva (ma)	
Comentários			
Nome do arquivo			
HISTÓRICO DE REVISÕES			
Revisão	Data	Descrição	Autor
01	01/10/17	Elaboração da primeira versão do documento.	ma
02	15/10/17	adição do tópico 8.2 Erros conhecidos.	ma

Índice

Introdução	5
Objetivos	5
Escopo	5
Referências	5
Visão Geral do Documento	5
Padrões e Convenções	5
Identificação dos Requisitos	5
Prioridade dos Requisitos	6
Descrição Geral do Sistema	6
Requisitos Funcionais	8
Usuario	8
[RF001](Importante) Cadastrar usuário	8
[RF002](Importante) Cargos de Usuário	8
Projetos	8
[RF003](Essencial) Atributos de um Projeto	8
Casos de Uso	8
[RF004](Essencial) Atributos dos casos de uso	8
[RF005](Essencial) Casos de teste e casos de uso	8
Artefatos de Teste	9
[RF006](Essencial) Artefatos de teste	9
Casos de Teste	9
[RF007](Essencial) Atributos dos casos de teste	9
[RF008](Essencial) Caso de teste e artefato de teste	9
[RF009](Essencial) Especificação dos casos de teste	9
[RF010](Essencial) Organização dos casos de teste	9
Roteiros de teste	9
[RF011](Essencial) Atributos dos roteiro de teste	9
[RF012](Essencial) Roteiros e casos de uso	9
[RF013](Essencial) Roteiros e casos de teste	9
Execução de Teste	10
[RF014](Essencial) A execução de teste	10
[RF015](Essencial) Execução de Casos de teste	10
[RF016](Essencial) Execução de roteiros de teste	10
Matriz de Rastreabilidade	10
[RF017](Essencial) A matriz de rastreabilidade	10
[RF018](Essencial) Matriz e Roteiro de teste	10

Corp

1 Introdução

Este documento tem como objetivo apresentar uma descrição detalhada dos requisitos do software PBTest (Pandora Box Test). Por meio deste, os desenvolvedores e qualquer pessoa interessada no projeto terão uma noção de como implementar essas funcionalidades deste sistema.

1.1 Objetivos

Dentre os objetivos deste documento estão o de listar da forma mais clara possível, os requisitos funcionais e não funcionais do sistema em questão.

1.2 Escopo

O presente documento visa ser escrito de forma abrangente e clara para que possa ser utilizado por todos os desenvolvedores, analistas e arquitetos envolvidos e ainda ser compreendido pelo cliente.

1.3 Referências

1.4 Visão Geral do Documento

A seguir são apresentadas as divisões deste documento e uma rápida descrição de cada seção:

- Seção 2 – Descrição Geral do Sistema: descreve o escopo do sistema e seus usuários de maneira geral.
- Seção 3 – Requisitos Funcionais: especifica os requisitos funcionais planejados para o sistema.
- Seção 4 – Requisitos Não-Funcionais: especifica todos os requisitos não-funcionais da primeira iteração do sistema.
- Seção 5 – Análise dos Requisitos: especifica as prioridades e dependências dos requisitos.
- Seção 6 – Diagrama de casos de uso: resume o relacionamento entre os casos de uso que executam os requisitos elicitados, apresentando casos de uso com fluxos de eventos, entradas e saídas.
- Seção 7 - Diagrama de classes.

1.5 Padrões e Convenções

1.5.1 Identificação dos Requisitos

Para a especificação dos requisitos utilizaremos a seguinte representação:

[TIPODOREQUISITONúmero](PRIORIDADE) Nome

O campo TIPODOREQUISITO poderá ser especificado pelos códigos RF (Requisitos Funcionais) ou RNF (Requisitos Não-Funcionais). Já o campo Número será preenchido com um número correspondente à ordem em que os requisitos aparecem no documento.

O campo PRIORIDADE refere-se a prioridade atribuída a ao requisito em questão de acordo com a sessão seguinte.

1.5.2 Prioridade dos Requisitos

A cada requisito será atribuída uma prioridade. A descrição de cada uma segue abaixo:

- Essencial é um requisito imprescindível. Sem ele, o sistema não funcionará.
- Importante é um requisito que deve ser implementado, mas, se não for, o sistema funcionará do mesmo jeito, mas de maneira insatisfatória.
- Desejável é um requisito que trará um diferencial adicional ao sistema. Por isso, pode ser deixado para ser implementado por último ou em próximas iterações.

2 Descrição Geral do Sistema

O sistema tem como objetivo ser uma ferramenta que auxilie na documentação e visualização e execução de testes de software utilizando o framework de teste de unidade Junit.

Para atingir esse objetivo, o sistema deve permitir que o usuário especifique dentro do software um projeto. E a cada projeto deve ser possível especificar casos de uso.

Com um projeto e com pelo menos um caso de uso cadastro do a ele o usuário poderá vincular a cada artefato de seu projeto um ou mais casos de teste.

Com os artefato e casos de testes especificados, o usuário deve criar roteiros de testes. Esses roteiros de testes devem ser compostos de pelo menos um artefato caso de teste.

Cada roteiro de teste deve ser executado pelo sistema quando solicitado pelo usuário. Esta execução deve gerar uma matriz de rastreabilidade. que permita ao usuário visualizar os resultados da execução dos testes.

2.1 Escopo Negativo

Devido a dimensão que o projeto pode ter, faz-se relevante definir o escopo não apenas dizendo as coisas que serão feitas, mas também deixando claro o que não fará parte do nosso escopo.

Algumas funcionalidades foram tidas pela equipe como funcionalidades que precisam de uma maior atenção. Sendo assim esse projeto se compromete a desenvolver apenas as funcionalidades de prioridade importante. Não fazem parte do escopo desse

proposta serviços tais como:

- ☆ Desenvolvimento de funcionalidades extras: só serão implementadas as funcionalidades explicitamente descritas.
- ☆ Importar dados em qualquer formato: o requisito de importar dados só poderá importar dados no formato especificado pelo programa, não garantindo resultados corretos se o formato não for respeitado.
- ☆ Controles de segurança sofisticados: a segurança do acesso dos usuários será com senha cadastrada.

2.2 Descrição dos Atores

O sistema apresenta apenas um ator, o usuário. Este ator pode realizar as seguintes operações dentro do sistema.

- Cadastrar um projeto dentro do sistema.
- Cadastrar caso de uso dentro do sistema.
- Vincular a artefatos do projeto casos de teste.
- Especificar sequências de testes.
- Executar as sequências de testes.

3 Requisitos Funcionais

Esta seção apresenta em detalhes os requisitos funcionais do sistema.

3.1 [RF001](Desejável) Cadastrar usuário

Um usuário cadastrado deve ter nome, cargo, E-mail, senha, foto de perfil, telefone e descrição. Sendo que os campos de foto de perfil, telefone e descrição tem seu preenchimento opcional.

3.2 [RF002](Desejável) Cargos de Usuário

Usuário podem ter os cargos de: Gerente de projeto, Líder de teste Testador.

3.3 [RF003](Essencial) Atributos de um Projeto

Para um projeto ser criado é necessário que o usuário preencha as seguintes informações: nome do projeto, descrição do projeto, prefixo para os casos de teste, prefixo para os roteiros de teste, prefixo para casos de uso, e caminho do sistema para o diretório raiz dos artefatos do projeto e o caminho para os casos de teste.

3.4 [RF004](Essencial) Atributos dos casos de uso

Um caso de uso é composto de uma descrição, objetivo, atores e um título que são cadeias de caracteres. Todos esses atributos são obrigatórios.

3.5 [RF005](Essencial) Casos de teste e casos de uso

A um caso de uso, deve ser possível que sejam vinculados casos de teste.

3.6 [RF006](Essencial) Artefatos de teste

O sistema deve considerar como artefato de testes qualquer arquivo com extensão ou .class que existir na pasta raiz ou sub pastas do projeto. A cada artefato, o usuário deverá poder vincular 1 ou mais casos de teste.

3.7 [RF007](Essencial) Atributos dos casos de teste

Um caso de teste é composto por: nome, nome do usuário que o especificou uma descrição textual que explique em linguagem natural o que está sendo testado.

3.8 [RF008](Essencial) Atributos dos roteiros de teste

Um roteiro de teste é composto por: nome, descrição, nome do usuário que o criou, caso de uso, casos de teste.

3.9 [RF009](Essencial) Roteiros e casos de teste

A um roteiro de teste deve ser possível anexar casos de teste. Pelo menos um caso de teste deve ser anexado para que a criação do roteiro seja completada.

3.10 [RF011](Essencial) Execução de roteiros de teste

O sistema deve permitir que o usuário execute roteiros de teste. A execução dos casos de teste deve ser feita na ordem em que estão especificados no roteiro de teste.

3.11 [RF012](Essencial) A matriz de rastreabilidade

Uma matriz de rastreabilidade deve conter em suas colunas os casos de teste. As linhas devem ser os artefatos testados. A célula de intersecção deve conter o resultado do teste.

3.12 [RF013](Importante) Configuração da execução de testes

Ao salvar um projeto, o sistema deve criar um arquivo de propriedades no diretório

de casos de testes informado pelo usuário. Este arquivo de propriedades deve permitir ao usuário informar um “.jar” do framework JUnit diferente do usado pelo sistema para a execução dos casos de teste. Deve permitir também que seja informado classpath adicionais para a jvm, bem como, novos parâmetros para a mesma. Além de permitir que o usuário troque o comando de execução de testes junit para um de sua escolha.

4 Requisitos Não-Funcionais

4.1 [RNF001]Códigos

Todos os códigos identificadores devem ser gerados automaticamente pelo sistema da seguinte forma, PREFIXO<Contador>. Onde PREFIXO é a cadeia de caracteres definida pelo usuário para cada artefato do projeto, e <contador> um número inteiro.

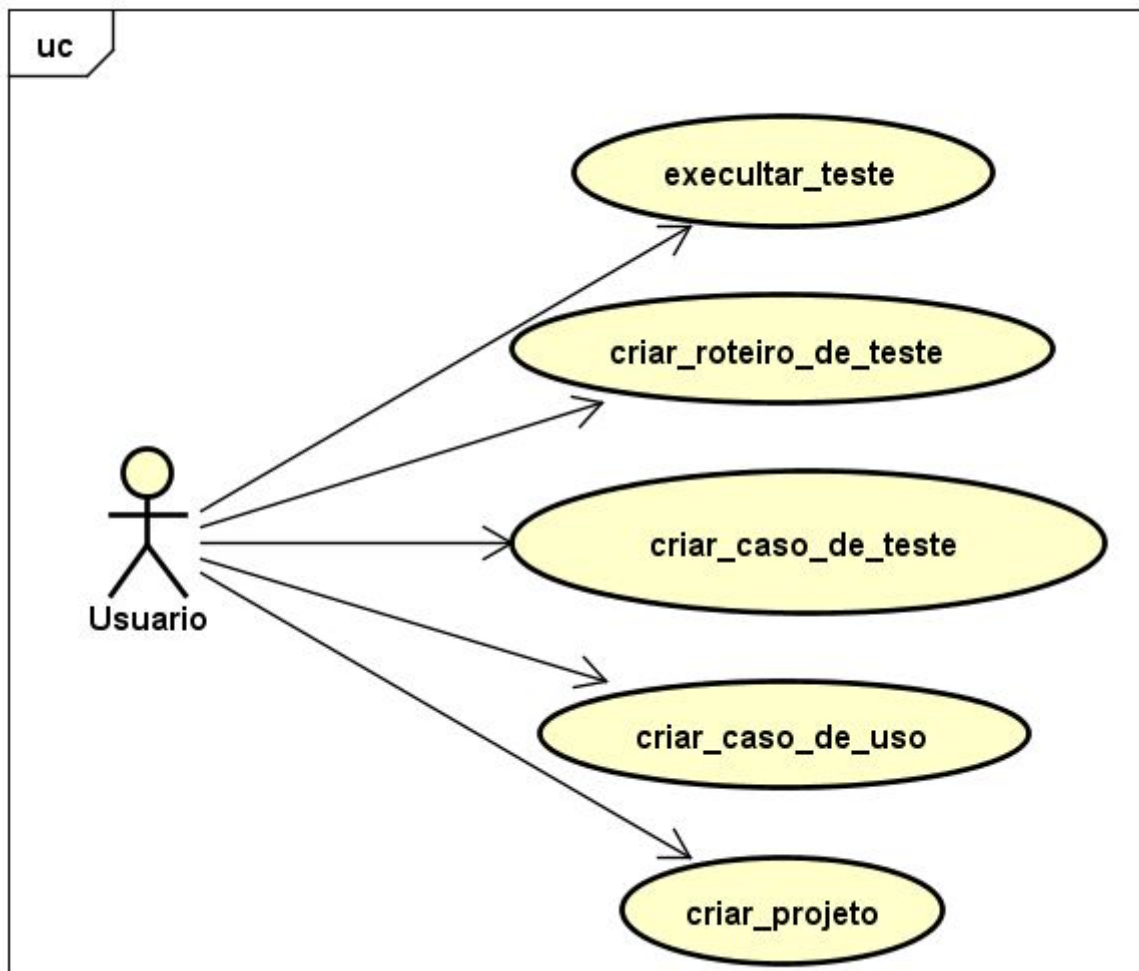
4.2 [RNF002] Hardware

A aplicação necessita de um monitor com no mínimo uma resolução de 1200x720, 2g de memória ram.

4.3 [RNF003] Edição das informações de um projeto

Na edição das informações de um projeto, o usuário poderá apenas informar novos valores para: nome, src para as classes de produção e de testes, e uma nova descrição de projeto.

5 Diagrama de Casos de Uso



5.1 Descrição de Casos de Uso

UC001	Criar um projeto
Breve Descritivo: Este caso de Uso descreve o processo de criação de um projeto dentro do software	
Pré-condições: O usuário logado é do tipo Gerente de Projeto	
Atores: gerente de projeto	
Cenários Principais:	
1 – Informar o nome do projeto.	
2 –Informar os prefixos de: caso de uso, caso de teste e roteiro de teste.	
3 – Informar o diretório raiz do projeto;	
4 – Informa a descrição do projeto.	
5 – Finaliza a criação do projeto;	
Cenários Alternativos:	

Analista de Negócios: -
Entrevistado: -
Data: 01/10/2017
Versão: 1.0

UC002	Criar um caso de teste
Breve Descritivo: Este caso de Uso descreve o processo de criação de um caso de teste dentro do software.	
Pré-condições: Ter um projeto ativo; Estar na tela de casos de teste;	
Atores: Usuário	
Cenários Principais:	
1 - Selecionar um artefato de teste.	
2 - Clicar no botão “ <i>novo caso de teste</i> ”	
3 - Informar o nome e descrição do caso de teste.	
4 - Informar vincula o caso de teste a um caso de uso existente.	
5 - Informar a classe de teste.	
7 - Clicar no botão “ <i>salvar</i> ”.	
Cenários Alternativos:	
3, 4, 5, 6 - Se o usuário deixar de informar qualquer uma destas informações o sistema o alertará dizendo que a informação é requerida.	
Analista de Negócios: -	
Entrevistado: -	
Data: 10/11/2017	
Versão: 1.0	

6 Diagramas de Atividade

Diagrama de atividades do caso de uso UC002. [link para o gitHub](#).

7 Diagrama de Classes

Diagrama de classes disponível em: [link para o GitHub](#).

8 Considerações Finais

8.1 Conclusões

Após dois meses de desenvolvimento, o trabalho foi concluído com a presença de todos os requisitos funcionais importantes e as restrições dos requisitos não funcionais implementadas.

Durante o desenvolvimento do projeto, com certeza a atividade mais trabalhosa foi a

codificação da interface gráfica. digo isto pois até o início do desenvolvimento não possuía experiência com a API gráfica Swing do java, ou semelhantes.

Utilização do padrão arquitetural MVC foi de grande auxílio nos momentos em que partes do código tiveram de ser fatoradas, ou refeitas. Como no caso de quando toda a View foi reescrita para tornar a interface mais flexível.

8.2 Trabalhos Futuros

8.2.1 Requisitos futuros

8.2.1.1 Relatorios

Implementar a extração da documentação de um projeto para o formato pdf ou odg.

8.2.1.2 Estatísticas

Painel que permita ao usuário visualizar estatísticas sobre um projeto. Porcentagem de falhas e acertos de e casos de teste, erros em um caso de uso, etc.

8.2.1.3 Leitura de arquivos de propriedades

Ler o documento de propriedades de um projeto gerado pela IDE.

8.2.1.4 Compilar classes

Fazer com que o sistema compile as classes. Sem bagunçar os arquivos do projeto.

8.2.1.5 Relacionamento entre artefato e documentação

Mostrar em um grafo ou diagrama a relação entre os casos de uso, artefatos casos de teste e sequências.

8.2.1.6 Execução modular

Criar Módulos de execução de teste. Para que a execução de teste seja independente do sistema. Para permitir a execução de testes em projetos implementados em outras linhagens.

8.2.2 Atores ideais

O sistema apresenta três atores diferenciados pelas funcionalidades a que tem acesso. A cada nível de hierarquia as funcionalidades apenas aumentam de forma que cada nível é um tipo especial do anterior.

Testador: Qualquer usuário cadastrado no sistema. a eles é dado o direito de especificar e executar casos de teste vem como os roteiros de teste.

Gerente de Teste: A esse usuário é compelido o poder de especificar casos de uso e roteiros de teste. E cadastrar usuário do tipo Testador.

Gerente de Projeto: Apenas ele pode realizar operações como, deletar usuário

do sistema, deletar projetos, cadastrar usuários do tipo Gerente de Teste. Apenas ele pode Criar Projetos.

