**Gerador Burndown**

**Plano de Teste de Aplicação**

**Versão 1.3.1**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 09/05/2015 | 1.0 | Versão inicial | Thiago Schulz |
| 23/05/2015 | 1.1 | Revisão | Glauber Maximiano |
| 23/05/2015 | 1.2 | Edição | Sérgio Souza |
| 26/05/2015 | 1.3 | Edição | Sérgio Souza |
| 31/05/2015 | 1.3.1 | Correções | Sérgio Souza |

**Índice Analítico**

**1. Introdução**

1.1 Finalidade   
1.2 Escopo   
1.3 Público-alvo   
1.4 Terminologia e Acrônimos do Documento   
1.5 Referências   
1.6 Estrutura do Documento

**2. Missão de Avaliação e Motivação dos Testes**

2.1 Fundamentos   
2.2 Missão de Avaliação   
2.3 Motivadores dos Testes

**3. Itens de Teste-Alvo**

**4. Resumo dos Testes Planejados**

4.1 Resumo das Inclusões dos Testes

**5. Abordagem dos Testes**

5.1 Catálogos Iniciais de Ideias de Teste e Outras Fontes de Referência   
5.2 Tipos e Técnicas de Teste   
5.2.1 Teste de Integridade de Dados e de Banco de Dados   
5.2.2 Teste de Funcionamento   
5.2.3 Teste de Ciclos de Negócios   
5.2.4 Teste de Interface do Usuário   
5.2.5 Teste de Segurança e de Controle de Acesso   
5.3 Casos de Teste

**6. Critérios de Entrada e de Saída**

6.1 Plano de Teste – Criar sprint   
6.1.1 Critérios de Entrada de Plano de Teste   
6.1.2 Critérios de Saída de Plano de Teste   
6.1.3 Critérios de Suspensão e de Reinício   
6.2 Plano de Teste – Criar estória   
6.2.1 Critérios de Entrada de Plano de Teste   
6.1.2 Critérios de Saída de Plano de Teste   
6.2.3 Critérios de Suspensão e de Reinício   
6.3 Plano de Teste – Criar estória   
6.3.1 Critérios de Entrada de Plano de Teste   
6.3.2 Critérios de Saída de Plano de Teste   
6.3.3 Critérios de Suspensão e de Reinício

**7. Produtos Liberados**

7.1 Sumários de Avaliação de Testes   
7.2 Geração de Relatórios sobre Cobertura de Teste   
7.3 Registros de Incidentes e Solicitações de Mudança   
7.4 Conjunto de Testes de Regressão e Scripts de Teste de Suporte   
7.5 Produtos de Trabalho Adicionais   
7.5.1 Resultados Detalhados dos Testes   
7.5.2 Scripts de Teste Funcionais Automatizados Adicionais   
7.5.3 Guia de Teste   
7.5.4 Matrizes de Rastreabilidade

**8. Necessidades Ambientais**

8.1 Hardware Básico do Sistema   
8.2 Elementos de Software Básicos do Ambiente de Teste   
8.3 Ferramentas de Produtividade e de Suporte   
8.4 Configurações do Ambiente de Teste

**Plano de Teste da Aplicação Gerador Burndown**

**1.**     **Introdução**

**1.1**     **Finalidade**

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias para planejar e controlar o esforço de teste referente a uma iteração específica. Em nosso caso, a geração do gráfico de *Burndown*.

**1.2**     **Escopo**

Ele descreve a abordagem técnica e detalhada dada ao sistema gerada pelos analistas de TI para verificar o correto funcionamento do sistema, de acordo com os requisitos anteriormente levantados

Serão realizados um teste de "caixa branca" e "caixa preta" para gerar o gráfico de cada *sprint*.

Principais objetivos:

* Verificar o gráfico gerado pelo sistemas de acordo com o seguinte cenário: uma equipe de 4 pessoas trabalhando 8 horas por dia, desenvolvendo 2 *sprints* de 5 dias cada, totalizando 10 dias (4 pessoas por dia (32 horas por dia), trabalhando 10 dias);

Testes a serem realizados:

* Data inicial e final da *sprint*: quantidade de dias tem que resultar em 2 *sprints* de 5 dias cada;
* Quantidade de recursos: tem resultado esperado entre 1 e 4 pessoas;
* Quantidade de horas trabalhadas por dia: tem que gerar um valor no intervalo entre 1 e 32.

**1.3**     **Público-alvo**

Este documento destina-se a equipe aos interessados no desenvolvimento da aplicação de gerador *burndown*. Testadores, desenvolvedores e, neste caso o avaliador do sistema, são os principais interessados e o público alvo do documento.

**1.4**     **Terminologia e Acrônimos do Documento**

Termos utilizados nestes documento serão de acordo com metodologia *SCRUM*.

**1.5**     **Referências**

[1] Documento de especificação do Gerador de Burndown.

**1.6**     **Estrutura do Documento**

N/A.

**2.**     **Missão de Avaliação e Motivação dos Testes**

Os testes descritos neste documento têm por objetivo garantir o desenvolvimento de uma aplicação que atenda aos resultados esperados e confiáveis. Valores corretos são exigências da aplicação, pois os gráficos gerados pela aplicação guiaram equipes no desempenho de suas funções.

**2.1**     **Fundamentos**

Considerando que o gerador de *Burndown* desenvolvido deve obedecer alguns padrões de qualidade, desenvolvimento e confiabilidade serão aplicados testes para validar as funcionalidades conforme descritas no documento de especificação do Gerador de *Burndown*.

**2.2**     **Missão de Avaliação**

Este trabalho tem por missão realizar uma bateria de testes no código com o intuito de garantir a entrega de uma aplicação confiável por meio da geração de resultados corretos. Uma aplicação confiável significa a garantia da satisfação do cliente final.

**2.3**     **Motivadores dos Testes**

O desenvolvimento deste plano de testes tem por principal objetivo a detecção de falhas de funcionalidades existente na aplicação. Criando um plano de testes que consiga tratar falhas de grande impacto no sistema aumenta aspectos de qualidade do sistema, confiabilidade dos dados e melhor satisfação por parte do cliente.

**3.**     **Itens de Teste-Alvo**

A listagem abaixo identifica os itens de software, que serão sujeitos aos testes abordados por este plano.

* Métodos existentes no código;
* Funções existentes no código;
* Classes.

**4.**     **Resumo dos Testes Planejados**

Os testes realizados devem abranger os seguintes itens:

* Criação de *sprint*;
* Criação de estória; e
* Criação de item histórico.

**4.1**     **Resumo das Inclusões dos Testes**

**Criação de *Sprints***

|  |  |
| --- | --- |
| Objetivo do Tipo de Teste: | Verificar se a criação da *sprint* suporta o recebimento de data inicial e final e gera o total de horas adequadamente |
| Técnica: | Ao inserir as duas datas (inicial e final) o sistema deverá gerar o tempo total na *Sprint* |
| Critérios de Êxito: | O total de horas da *sprint* deve ser positivo e em horas. |
| Considerações Especiais: | A geração de resultado correto depende da inserção de data inicial e data final corretamente e do preenchimento correto da lista de estórias. |

**Criação de Estória**

|  |  |
| --- | --- |
| Objetivo do Tipo de Teste: | Verificar se a estória possui dados válidos quanto tempo estimado, código e quantidade de pontos |
| Técnica: | Deverá ser criada uma lista de estória e definir o total de horas para aquela *sprint*. Esta informação é essencial no item *Sprint*. |
| Critérios de Êxito: | Todos as horas e pontos de cada estória são somados e gerados valores totais que definem o tempo gasto daquela *sprint*. |
| Considerações Especiais: | Uma estória deve ter um valor positivo e a quantidade de pontos deve levar em consideração o tempo total da estória. Estória com tempo estimado alto deve ter pontos maiores. |

**Criação de Item Histórico**

|  |  |
| --- | --- |
| Objetivo do Tipo de Teste: | Verificar se a geração do gráfico de burndown está seguindo o padrão correto |
| Técnica: | Os dados da estória e tempo gasto deve ser inserido no sistema e o gráfico deve ser gerado em formato de histórico. Cada dia deve possui um novo resultado e o gráfico deve acompanhar estas informações inseridas. |
| Critérios de Êxito: | Cada dia deve possuir um resultado positivo e o gráfico deve ser preenchido com parte da linha que representa aquela informação inserida. |
| Considerações Especiais: | O valor máximo de item histórico é número total de dias para *sprint* seguido pelo projeto. Um projeto com *sprint* a cada 7 dias deve possuir sete itens no gráfico. |

**5.**     **Abordagem dos Testes**

Os testes serão feitos de forma automática utilizando a framework de testes automatizados do C# (NUnit). Cada unidade do código irá passar por uma análise inicial para identificação de suas entradas e saídas.

**5.1**     **Catálogos Iniciais de Ideias de Teste e Outras Fontes de Referência**

A internet será fonte de busca de ideias em casos de testes de alta complexidade. Outras referências poderão ser usadas para auxiliar no processo de desenvolvimento de planos de testes, como o livro Testes Funcionais de Software Molinari, Leonardo.

**5.2**     **Tipos e Técnicas de Teste**

*5.2.1*     *Teste de Integridade de Dados e de Banco de Dados*

Não se aplica a este projeto.

*5.2.2*     *Teste de Funcionamento*

|  |  |
| --- | --- |
| **Objetivo da Técnica:** | Este item tem por objetivo garantir o desenvolvimento de testes apropriados a verificação de funcionalidades do sistema, como entrada de dados, processamento e resultado (gráfico de burndown). |
| **Técnica:** | Executar testes automatizados de cada unidade do sistema por meio do framework de testes da ferramenta, usando dados válidos e inválidos, para verificar o seguinte:   * Os resultados esperados ocorrem quando dados válidos são usados; * Os resultados não esperados são exibidos quando dados inválidos são usados; * Mensagens são geradas corretamente quando o sistema recebe inputs incorretos. |
| **Estratégias:** | Uma série de asserts serão usados para verificar a saída de dados de cada unidade. Por exemplo, para verificar se o resultado esperado em uma unidade está correta o assertEquals será usado para testar a saída do dado e compara-lo a um resultado esperado. |
| **Ferramentas Necessárias:** | Para execução dos testes será usado a framework de testes automatizados da linguagem usada na aplicação. Além disso, conexão com a internet será imprescindível para o desenvolvimento de testes eficientes. |
| **Critérios de Êxito:** | * Todos os testes planejados foram executados; * Todos os testes planejados e executados que apresentaram defeito tiveram tratamento adequado e correção da falha. |
| **Considerações Especiais:** | N/A |

*5.2.3*     *Teste de Ciclos de Negócios*

Não se aplica a este projeto.

*5.2.4*     *Teste de Interface do Usuário*

Este projeto não irá ter interface de usuário, porém será realizado um teste manual para verificar se os resultados condizem com o esperado.

|  |  |
| --- | --- |
| **Objetivo da Técnica:** | Este teste tem por objetivo verificar se o gráfico gerado pela aplicação condiz com um modelo criado manualmente. Serão avaliados eixos, linhas e informações gráficas geradas pela aplicação. |
| **Técnica:** | Primeiramente serão definidos dados fictícios de um gráfico de burndown. Será criado o gráfico deste teste manualmente.  Em um segundo momento, os mesmos dados serão inseridos na aplicação e gerado o gráfico.  Por fim, será feito uma comparação entre os dois modelos e checado a sua semelhança. |
| **Estratégias:** | Os mesmos dados podem ser inseridos no modelo manual e automático e os resultados comparados. Isso garantiria que o Sistema trabalha corretamente.  Valores divergentes entre o teste automático e manual também poderiam ser utilizados para verificar testes falhos. |
| **Ferramentas Necessárias:** | * Papel; * Lápis; * Régua; e * Framework de testes automatizados da linguagem usada na aplicação. |
| **Critérios de Êxito:** | * O gráfico gerado manualmente possui mesma forma com o gráfico gerado pela aplicação; * Possíveis erros identificados foram corrigidos. |
| **Considerações Especiais:** | N/A |

*5.2.5*    *Teste de Segurança e de Controle de Acesso*

Não se aplica a este projeto.

**5.3**     **Casos de Teste**

Abaixo a lista de testes caixa preta que serão executados:

Caso de teste: Criar Sprint

Caso de teste: Criar estória

Caso de teste: Gerar gráfico

**6.**     **Critérios de Entrada e de Saída**

**6.1**     **Plano de Teste – Criar *sprint***

*6.1.1*     *Critérios de Entrada de Plano de Teste*

Somente valores inteiros positivos devem ser inseridos;

O tempo da *sprint* deve estar dentro dos limites: >= 1 && <= 30

*6.1.2*     *Critérios de Saída de Plano de Teste*

|  |  |
| --- | --- |
| **Entrada** | **Saída** |
| **1** | Exibição de somente um dia no gráfico |
| **2** | Exibição de duas datas no gráfico |
| **0** | *Throw Exception* |
| **30** | Exibição de trinta dias no gráfico |
| **29** | Exibição de vinte e nove dias no gráfico |
| **31** | *Throw Exception* |
| **Ab** | *Throw Exception* |
| **ab2** | *Throw Exception* |
| **2er** | *Throw Exception* |
| **\*2** | *Throw Exception* |
| **Sem inserção de valor** | *Throw Exception* |

*6.1.3*     *Critérios de Suspensão e de Reinício*

Caso seja encontrado algum defeito no código;

O reinício dos testes se dá com a correção do defeito.

**6.2**     **Plano de Teste – Criar estória**

*6.2.1*     *Critérios de Entrada de Plano de Teste*

Dois valores devem ser inseridos.

Somente valores inteiros positivos devem ser inseridos;

O primeiro referente ao tempo estimado da estória e o segundo o número de pontos que aquela estória possui.

Tempo estimado da estoria (min): >= 1 && <= (9600 \* número de desenvolvedores na equipe)

9600 representa: 8hs de trabalho por dia \* 20 dias de trabaho por mês \* 60 minutos

Pontos: >= 1

*6.1.2*     *Critérios de Saída de Plano de Teste*

|  |  |
| --- | --- |
| **Tempo estimado da estória** | |
| **Entrada** | **Saída** |
| **1** | Criação de uma estoria com um minuto de duração |
| **2** | Criação de uma estoria com dois minutos de duração |
| **0** | Throw Exception |
| **ab** | Throw Exception |
| **ab2** | Throw Exception |
| **2er** | Throw Exception |
| **\*2** | Throw Exception |
| **Sem inserção de valor** | Throw Exception |

|  |  |
| --- | --- |
| **Pontos por estória** | |
| **Entrada** | **Saída** |
| **1** | Criação de uma estória com 1 ponto |
| **2** | Criação de uma estória com 2 pontos |
| **0** | *Throw Exception* |
| **ab** | *Throw Exception* |
| **ab2** | *Throw Exception* |
| **2er** | *Throw Exception* |
| **\*2** | *Throw Exception* |
| **Sem inserção de valor** | *Throw Exception* |

*6.2.3*     *Critérios de Suspensão e de Reinício*

Caso seja encontrado algum defeito no código;

O reinício dos testes se dá com a correção do defeito.

**6.3**     **Plano de Teste – Criar estória**

*6.3.1*     *Critérios de Entrada de Plano de Teste*

Este teste tem como entrada conforme nos dois casos de testes expressos acima.

*6.3.2*     *Critérios de Saída de Plano de Teste*

Os mesmos resultados expressos nos casos de testes acima. Caso todos os valores inseridos estavam corretos será gerado o gráfico de burndown.

*6.3.3*     *Critérios de Suspensão e de Reinício*

Caso seja encontrado algum defeito no código;

Reinício se dá com a correção do defeito.

**7.**     **Produtos Liberados**

* Código de teste pronto e funcional; e
* Prints de resultados de teste atestando o sucesso da tarefa.

**7.1**     **Sumários de Avaliação de Testes**

A cada conclusão com sucesso do teste de unidade do código o documento “Sumário de Avaliação de Testes” será atualizado. Informações como tipo de teste realizado, unidade testada, resultado do teste, etc serão incluídos neste documento.

Um artefato que comprove o resultado dos testes também será incluído. Para este trabalho o artefato utilizado será o print de tela do resultado.

**7.2**     **Geração de Relatórios sobre Cobertura de Teste**

Esta informação será incluída no documento “Sumário de Avaliação de Testes” conforme regras descritas acima.

**7.3**    **Registros de Incidentes e Solicitações de Mudança**

No Sumário de avaliação de testes também receberá as unidades que não passaram no teste e que necessitam de correção.

**7.4**     **Conjunto de Testes de Regressão e Scripts de Teste de Suporte**

N/A

**7.5**     **Produtos de Trabalho Adicionais**

A framework de teste existente no Visual Studio será usada para geração dos testes automatizados.

*7.5.1*     *Resultados Detalhados dos Testes*

Não se aplica. Resultados serão inseridos no Sumário de Avaliação de Testes.

*7.5.2*     *Scripts de Teste Funcionais Automatizados Adicionais*

Não se aplica.

*7.5.3*     *Guia de Teste*

*7.5.4*     *Matrizes de Rastreabilidade*

**9.**     **Necessidades Ambientais**

**9.1**     **Hardware Básico do Sistema**

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste Plano de Teste.

|  |  |  |
| --- | --- | --- |
| **Recurso** | **Quantidade** | **Nome e Tipo** |
| PCs de Desenvolvimento de Teste | 1 | Processador i3 ou superior.  Memória de 4GB ou acima. |

**9.2**     **Elementos de Software Básicos do Ambiente de Teste**

São necessários os seguintes elementos de softwares básicos no ambiente de teste deste Plano de Teste.

|  |  |  |
| --- | --- | --- |
| **Nome do Elemento de Software** | **Versão** | **Tipo e Outras Observações** |
| Windows | 7 ou superior | Sistema Operacional |
| Internet Explorer |  | Navegador da Internet |
| Visual Studio | 2012 | Suite de desenvolvimento |
| NUnit |  | Framework de testes automatizados. |

**9.3**     **Ferramentas de Produtividade e de Suporte**

N/A

**9.4**     **Configurações do Ambiente de Teste**

N/A