MINERVA SOLUTIONS

Plano de Testes e Inspeções: Iteração 4

Data: 06/11/2018	**Objetivo Estratégico:** Gerenciamento dos dados administrativos do Instituto Ada Lovelace.
ID: 1009	**Nome do Projeto:** sigAda - Sistema de Gerenciamento do Ada
CC: Paulo Pires	**Cliente:** Instituto Ada Lovelace
Patrocinador: UFRJ	**Gerente de Projeto:** Rafael Cardoso

1. Histórico de Versão

Data	Versão	Descrição	Autor(es)
06/11/2018	1.0.0	Abertura do Documento	Eduardo, Rafael e Tainá
18/11/2018	1.1.0	Adição dos itens 2, 4, 5, 6 e 7	Tainá
19/11/2018	1.2.0	Adição dos itens 3 e 8 Atualização das referências	Tainá
09/12/2018	1.2.1	Correção ortográfica	Rafael
09/12/2018	2.0.0	Reestruturação do documento	Tainá, Rafael e Eduardo
10/12/2018	2.0.1	Atualização dos links	Tainá

2. Introdução

2.1 Finalidade

Este documento tem como finalidade apresentar:

- Os alvos do teste, ou seja, quais serão os artefatos a serem testados dentro do escopo da iteração 4.
- As metas e objetivos a serem alcançados com a realização de tal teste.
- A abordagem que será utilizada neste teste.
- Os recursos necessários para realização do teste.
- Os produtos liberados ao fim do teste, ou seja, os relatórios de resultado do teste.

2.2 Escopo

Este documento diz respeito ao planejamento de teste elaborado para ser aplicado à iteração 4 do projeto sigAda, em específico sobre a parte de implementação, descrita no Documento de Plano de Iteração, cujo link está disponível no tópico de "Referências".

2.3 Visão Geral

Este documento apresenta uma estrutura inicial parecida com os outros documentos já gerados, ou seja, inicia-se com o histórico de versões e em seguida com uma breve introdução sobre o objetivo da construção de tal documento, bem como qual é seu escopo e referências para outros documentos, estes posteriormente mencionados ao longo do documento.

Após a introdução, segue de fato o planejamento decidido para a realização do teste e das inspeções para a iteração 4, abordando os motivos pelos quais tal teste/inspeção está sendo realizado, bem como quais são os itens-alvo do mesmo. Além disso, é apresentado quais testes/inspeções em específicos que serão feitos, assim como as estratégias utilizadas para a execução destes.

Por fim, é descrito os recursos a serem utilizados neste teste e na inspeção, e quais são os produtos liberados pelos mesmos, abordando como serão feitos os relatórios de resultado e as documentações de solicitação de mudança do código.

2.4 Referências

- Documento de Plano de Iteração (ID - 1003)
- Documento de Relatório de Resultados de Teste e Inspeção (Template)

3. Missão de Avaliação e Motivação dos Testes e das Inspeções

3.1 Missão de Avaliação

Este teste tem como missão encontrar o maior número de erros possíveis, principalmente aqueles que seriam críticos ao funcionamento do sistema, como a autenticação do login. Além disso, deve ser analisado e avaliado os riscos de qualidade e custo oferecidos por tais erros. Também é objetivo deste teste verificar a veracidade da representação dos casos de uso implementados. Por

fim, ao combinar todas as informações mencionadas previamente, será gerado um relatório de resultados, melhor descrito nos tópicos posteriores.
Já a inspeção também tem como objetivo encontrar um bom número de erros, porém é feito de maneira estática. Utiliza-se de ferramentas como o Findbugs para encontrar possíveis erros, que são comuns de acontecer, de maneira que não exija a entrada de dados. Tais erros poderiam ser erros gramaticais, problemas de sintaxe e anomalias de fluxo de dados.

3.2 Motivadores de Teste e Inspeção

Um esforço deve ser feito para que este planejamento seja realizado, uma vez que, como os itens-alvo deste teste/inspeção fazem parte da "base" da construção do sistema, é necessário que estes estejam corretos de acordo com as especificações geradas pelo projeto, pois podem colocar em risco a qualidade do projeto, bem como o orçamento planejado para o mesmo.

4. Itens Alvo dos Testes e Inspeções

Os itens listados a seguir são os alvos do teste e da inspeção descrito por este documento:

Pacote "com.ms.sigada.model"

- Login.java
- Aluno.java
- Pessoa.java
- AlunoDAO.java
- LoginDAO.java

Pacote "com.ms.sigada.controller"

- LoginController.java
- AlunoController.java

Pacote "com.ms.sigada.view"

- LoginForm.java
- ManterAlunoForm.java
- MenuRHForm.java

5. Resumo dos Testes e das Inspeções Planejadas

5.1 Verificações dinâmicas

Abaixo segue a lista de testes que serão feitos nos itens de teste-alvo mencionados no tópico anterior:

Teste de Unidade

- Verificar se está sendo aceito apenas CPF válido.
- Verificar se as solicitações ao banco de dados estão sendo feitas corretamente.

Teste de Integração

- Verificar se o nome de usuário digitado no form de login será mostrado no form de menu.
- Verificar se ao clicar para fazer uma ação referente à um aluno no form de menu, a janela referente a essa ação será exibida.
- Verificar se ao clicar para fazer uma ação referente à um funcionário no form de menu, a janela referente a essa ação será exibida.
- Verificar se ação referente aos botões salvar, limpar, ou deletar estão funcionando de acordo.
- Verificar se ao logar-se como funcionário do R.H., o menu direcionado para o mesmo será exibido.

Teste de Sistema

- Verificar se a autenticação do login está ocorrendo de maneira correta, ou seja, quem está cadastrado, é logado e quem não está, não é.

- Verifica se o processo de manter aluno, ou seja, cadastrar um novo aluno, alterar suas informações e, em seguida, buscá-lo e deletá-lo está funcionando da forma como esperada.

Teste de Release

- Verifica se as funcionalidades desenvolvidas e testadas nas interações anteriores (teste de unidade, teste de integração e teste de sistema) estão funcionando como o esperado, assim como o desempenho do software, conforme a especificação.

5.2 Verificações estáticas

Análise Estática Automatizada

- Verificação de má prática no código → verifica se há violação de boas práticas de codificação, como por exemplo usar operadores "==" e "!=" para comprar Strings, métodos que só se diferenciem por letras maiúsculas.
- Verificação de corretude → verifica se o código apresenta alguma uma falha de codificação, como algum type cast que não deveria ser feito.
- Verificação de performance → verifica a performance do código, como por exemplo se há campos não são lidos ou não utilizados.
- Verificação de código evasivo → verifica se o código está confuso, anômalo ou escrito de forma que leva a erros, como por exemplo switch sem default.
- Verificação de corretude multithreaded → verifica se a corretude da sincronização, como por exemplo quando se faz a chamada do método wait, mas não se verifica se a condição para aguardar está sendo satisfeita.

6. Estratégia dos Testes e das Inspeções

Nome do teste	Teste de Unidade
Objetivo do teste	Analisar se o comportamento de um componente específico do programa está funcionando de forma adequada.
Técnica	Será feito vários testes com o auxílio da ferramenta Junit, de modo a automatizar o testes para futuros testes de regressão.
Ferramentas necessárias	JUnit
Critério de conclusão	• Para cada entrada dada a saída deve ser a esperada. • Funcionamento de acordo com o esperado.
Considerações especiais	Nenhuma

Nome do teste	Teste de Integração
Objetivo do teste	Analisar se o comportamento da integração das classes criadas estão corretas, de acordo com a especificação do projeto
Técnica	Será feito vários testes manuais pelos testadores referentes às funcionalidades mencionadas, utilizando dados de entrada, nesse caso a combinação usuário/senha, diferentes e os campos presente no formulário.
Ferramentas necessárias	Nenhuma
Critério de conclusão	● Todas as transações de um form para o outro tenha sido correta ● Todas as entradas disponíveis forem testadas ● Cada defeito tenha sido devidamente documentado
Considerações especiais	Nenhuma

Nome do teste	Teste de Conformidade
Objetivo do teste	Analisar se o que foi feito está de acordo com o que foi planejado nas especificações do projeto
Técnica	Será dado como teste manual com um conjunto de entradas válidas e outro de entradas inválidas e será verificado os seguintes tópicos: ● Quando for dado como entrada um dado válido, o resultado que é esperado deverá ocorrer, caso contrário, uma mensagem de erro deve ser exibida ● Cada regra de negócio será adequadamente aplicada.
Ferramentas necessárias	

Critério de conclusão	● Todas as entradas disponíveis terem sido testadas ● Cada defeito tenha sido devidamente documentado
Considerações especiais	Nenhuma

Inspeção	Análise Estática Automatizada
Objetivo do teste	A partir de padrões definidos, busca-se por possíveis falhas ou brechas para falhas presentes no código
Técnica	O código será processado pela ferramenta FindBugs, em que as categorias de bugs de má prática de código, corretude, performance, evasivo e corretude multithreaded serão buscadas utilizando o esforço máximo para aumentar a precisão da busca dessas categorias ao longo do código.
Ferramentas necessárias	Findbugs
Critério de conclusão	● Todo o código ser percorrido pelo Findbugs. ● Cada bug encontrado será documentado.
Considerações especiais	Nenhuma

7. Recursos

A seguir serão apresentados os recursos a serem utilizados para a execução do teste planejado por este documento:

Recursos Humanos

Função	Qtd. mínima recomendada	Responsabilidade(s)
Gerente de teste	1	Gerencia o planejamento e a

		execução dos testes
Designer de teste	1	Identificar, organizar e implementar os casos de teste
Testador	2	Executar os testes, registrar os resultados, documentar os erros e as solicitações de mudança

Recursos tecnológicos

Os testes e as inspeções se darão no próprio ambiente de desenvolvimento, Eclipse ou Netbeans, utilizando os sistemas operacionais Windows 10 ou Linux Ubuntu, utilizando as ferramentas JUnit para os testes e o Findbug para as inspeções. Além disso, será utilizado a ferramenta Google Docs para a documentação dos resultados, bem como as solicitações de mudança de código.

8. Produtos Liberados

8.1 Modelos de Teste e Inspeção

Para o teste e a inspeção descritos por este planejamento, será gerado um relatório de resultados, cujo link para um "template" está no tópico "Referências" no início do documento. Tal relatório contém uma breve introdução sobre o documento e informações sobre resultados do teste e da inspeção. Essas informações incluem um pequeno resumo dos resultados, uma descrição de como se deu a cobertura de teste/inspeção baseada nos requisitos e no código, uma listagem e análise dos defeitos encontrados, sugestões para ações a serem tomadas para solucionar tais problemas e, por fim, diagramas caso seja necessário tornar algum outro tópico mais claro ou acrescentar alguma informação extra.

8.2 Log de Testes e Inspeção

Será utilizado a ferramenta Google Docs para registrar e reportar, além de versionar, os relatórios dos resultados do teste.