

**Responsável pelo documento**

Isaias Silva

**Revisado por**

Victor Jatobá

**Validado por**

Victor Jatobá

**Resumo**

O Plano de Especificação de Teste foi desenvolvido no âmbito de oferecer ao projeto PBL uma ótima qualidade em seu funcionamento e tem como propósito especificar os casos de teste a respeito da aplicação

**Índice**

(Criar após o término do documento e da revisão, para a validação final já deve estar pronto.)

**1 Introdução**

* 1. **Objetivo**

O plano de especificação de casos de Teste do Projeto PBL, tem como objetivo de definir a metodologia de teste a ser seguida durante todo o desenvolvimento do projeto

* 1. **Âmbito**

Este plano de teste é realizado no âmbito do PROJETO PBL, realizado pelos alunos do 8° Semestre da disciplina de Tópicos Avançados em Informática II, do Curso de Sistemas de Informação pela Faculdade Unime – Lauro de Freitas.

* 1. **Definições Acrônimos e Abreviaturas**

*Framework –* Um framework é uma estrutura de trabalho que permite que outro projeto de software possa ser organizado e desenvolvido.

*Application Programming Interface –* Ou Interface de Programação de Aplicativos. Trata-se de um conjunto de rotinas e padrões disponibilizado por determinado software para a utilização de suas funcionalidades por outros programas.

1. **Especificações do Teste**

Os testes a seguir permitem averiguar a qualidade, estabilidade, disponibilidade, interface e funcionamento do Projeto PBL, entre outras funcionalidades.

* 1. **Procedimento**

Apesar de hoje em dia existirem várias maneiras de testar o software, para este teste em especifico será utilizado o teste de Caixa Preta, que consiste em testar o funcionamento geral do sistema, da base de dados, assim como o resultado esperado deste funcionamento.

Os testes Unitário ou teste de Caixa Preta são testes que podem incidir sobre um método, uma função interna, um programa, um componente e as diversas funcionalidades do Projeto PBL.

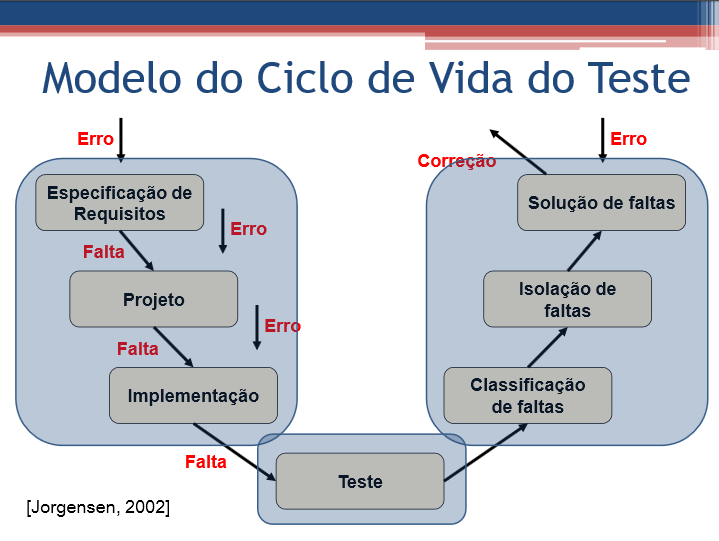
Estes testes desconhecem o funcionamento interno do sistema, ou seja não levam em consideração a maneira e a forma de codificação e sim o resultado esperado do código e do funcionamento do sistema, partem do princípio que a partir da entrada de dados (*input*) irão ser obtidos determinados resultados, caso o objeto de teste funcione conforme os requisitos especificam. Este tipo de teste estará presente em todas as etapas do processo de teste

* 1. **Critérios de Aceitação e Rejeição de Testes**

Uma das formas de buscar a qualidade nos testes é através de uma análise de cobertura de código atingido pelo teste unitário, a cobertura de código é um relatório que diz ao analista de teste quais os trechos de código foram executados por um conjunto de testes.

Os critérios de aceitação e rejeição dos testes será pela quantidade de cobertura do teste, sendo aceito em um valor corresponde entre 91% e 100%. Rejeitando portanto qualquer modulo ou funcionalidade na qual a cobertura e aprovação do teste seja inferior a 90%.

Os critérios de aceitação e rejeição de testes deverá ser analisado para cada interação do ciclo de vida do software, na qual seguirá o padrão descrito na imagem abaixo.



* 1. **Requisitos do Ambiente de Teste**

**2.3.1 Hardware e Software**

Com hardware e software, o teste tem como necessidades uma máquina com os requisitos recomendados para que possa rodar nas ultimas versões instáveis dos *browser* (Internet Explorer - Mozila Firefox, Google Crome)

**2.3.2 Ferramentas de Teste**

Existem diversos frameworks que geram a cobertura de código para testes, no Projeto PBL, será usado um plug-in do próprio Grails, chamado *coverage,* além das ferramentas de desenvolvimento Netbeans IDE, eclipse IDE e SpringTools Groovy Grails Suite.

1. **Casos de Teste**

***2.1 Tela de Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Pré-condições** | O sistema estar disponível para acesso via url |  |  |
| **N.º** | **Pré-condições** | **Ação** | **Resultado Esperado** |
| **01** | Ter apresentado a tela de Login após digitar a url na barra de endereços do navegador. | Clicar no botão “Logar.”. | O sistema deve abrir a tela “inicial do sistema”. |
| **02** | Ter apresentado a tela de Login após digitar a url na barra de endereços do navegador. | Clicar no botão “Esqueci Minha Senha”. | O sistema deve abrir a tela para que o usuário possa recuperar sua senha. |

***2.2 Tela Inicial***

|  |  |  |  |
| --- | --- | --- | --- |
| **Pré-condições** | O usuário ter efetuado o Login no sistema |  |  |
| **N.º** | **Pré-condições** | **Ação** | **Resultado Esperado** |
| **01** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Diversas ações (rodar o sistema e ver as possíveis ações.) | O sistema deve abrir a tela “inicial do sistema”. |
| **02** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Clicar no item de menu “Exemplo: Categoria” | O sistema deve abrir a tela de categoria, conforme o exemplo |
| **03** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Clicar no item de menu “Exemplo: Categoria” | O sistema deve abrir a tela de categoria, conforme o exemplo |
| **04** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Clicar no item de menu “Exemplo: Categoria” | O sistema deve abrir a tela de categoria, conforme o exemplo |
| **05** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Clicar no item de menu “Exemplo: Categoria” | O sistema deve abrir a tela de categoria, conforme o exemplo |
| **06** | Ter apresentado a tela inicial, (home) após o usuário ter efetuado o Login | Clicar no item de menu “Exemplo: Categoria” | O sistema deve abrir a tela de categoria, conforme o exemplo |

