

# Plano de Teste

nome do sistema

versão x.x

## Histórico das alterações

Data	Versão	Descrição	Autor(a)
dd/mm/aaaa	x.x	Release Inicial	Cleanio

# 1 - Introdução

Descrever os requisitos a testar, os tipos de testes definidos para cada iteração, os recursos de hardware e software a serem empregados e o cronograma dos testes ao longo do projeto. As seções referentes aos requisitos, recursos e cronograma servem para permitir ao gerente do projeto acompanhar a evolução dos testes.

Com esse documento, devemos:

- Identificar informações do projeto existente e os componentes de software que devem ser testados.
- Listar os Requisitos a testar.
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.
- Também é possível apresentar aqui o programa que será testado.

## 2 - Requisitos a Testar

Esta seção deve conter os casos de uso e requisitos não funcionais identificados como objetos dos testes ao longo do desenvolvimento do projeto. Como, em geral, os requisitos a testar são obtidos diretamente dos requisitos do sistema, esta seção é concebida como opcional. Assim sendo, sempre que novos requisitos a testar, que não constem como requisitos do sistema, forem identificados ou, simplesmente, por questões de organização e clareza, esta seção deve ser preenchida. Dependendo das informações disponíveis, essa seção pode começar a ser preenchida já nas primeiras iterações do ciclo de vida a partir do documento de requisitos.

Caso seja necessário, liste aqui os requisitos a testar subdivididos em casos de uso e requisitos não-funcionais.

### Casos de uso:

Identificador do caso de uso	Nome do caso de uso
id UC1	nome UC1

### Requisitos não-funcionais:

Identificador do requisito	Nome do requisito
id req1	nome req1

### 3 - Tipos de teste

Esta seção deve conter os tipos de testes escolhidos para cada iteração do projeto. Pode-se definir inicialmente apenas os tipos de testes que serão usados na próxima iteração, mas é possível também já registrar eventuais tipos de teste que se espera utilizar nas demais iterações. Com base no guia de testes, indique os tipos de testes que melhor se adequam aos requisitos, tipo da aplicação e seus recursos disponíveis e, caso necessário, complemente ou forneça mais detalhes da técnica e dos critérios de completude sugeridos no guia para cada tipo de teste indicado.

- Teste de interface de usuário;
- Teste de performance;
- Teste de carga;
- Teste de stress;
- Teste de segurança e controle de acesso;
- Teste de instalação;

#### 3.1 - Métodos da Classe

Para teste de funcionalidade. Aqui deve-se verificar se cada classe retorna o esperado. Se possível usar teste automatizado.

Objetivo	Descrição			
Técnica	<div><div>( )Manual</div><div>( )Automática</div></div>			
Estágio do teste	<div><div>( )Integração</div><div>( )Sistema</div><div>( )Unidade</div><div>Aceitação()</div></div>			
Abordagem do teste	<div><div>( )Caixa branca</div><div>( )Caixa preta</div></div>			
Responsável(is)	Equipe responsável			

## 3.2 - Persistência de Dados

Para teste de integridade de dados e do banco de dados. Aqui deve-se verificar se os dados não se perdem ao desligar o programa. Se o programa consegue se recuperar em caso de falha ou fechamento repentino. Se possível usar teste automatizado.

Objetivo	Descrição				
Técnica	<table><tr><td>( )Manual</td><td>( )Automática</td></tr></table>	( )Manual	( )Automática		
( )Manual	( )Automática				
Estágio do teste	<table><tr><td>( )Integração</td><td>( )Sistema</td><td>( )Unidade</td><td>Aceitação()</td></tr></table>	( )Integração	( )Sistema	( )Unidade	Aceitação()
( )Integração	( )Sistema	( )Unidade	Aceitação()		
Abordagem do teste	<table><tr><td>( )Caixa branca</td><td>( )Caixa preta</td></tr></table>	( )Caixa branca	( )Caixa preta		
( )Caixa branca	( )Caixa preta				
Responsável(is)	Equipe responsável				

### 3.3 - Integração dos Componentes

Para teste de funcionalidade. Aqui deve-se verificar se as classes e métodos conseguem fazer a integração entre elas para uma sequência de ações do programa. Se possível usar teste automatizado.

<b>Objetivo</b>	<b>Descrição</b>							
<b>Técnica</b>	<table border="1"><tr><td>( )Manual</td><td colspan="3">( )Automática</td></tr></table>				( )Manual	( )Automática		
( )Manual	( )Automática							
<b>Estágio do teste</b>	<table border="1"><tr><td>( )Integração</td><td>( )Sistema</td><td>( )Unidade</td><td>Aceitação()</td></tr></table>				( )Integração	( )Sistema	( )Unidade	Aceitação()
( )Integração	( )Sistema	( )Unidade	Aceitação()					
<b>Abordagem do teste</b>	<table border="1"><tr><td>( )Caixa branca</td><td colspan="3">( )Caixa preta</td></tr></table>				( )Caixa branca	( )Caixa preta		
( )Caixa branca	( )Caixa preta							
<b>Responsável(is)</b>	Equipe responsável							

### 3.4 - Testes funcionais

Objetivo	Descrição			
Técnica	<div><div>( )Manual</div><div>( )Automática</div></div>			
Estágio do teste	<div><div>( )Integração</div><div>( )Sistema</div><div>( )Unidade</div><div>Aceitação()</div></div>			
Abordagem do teste	<div><div>( )Caixa branca</div><div>( )Caixa preta</div></div>			
Responsável(is)	Equipe responsável			

## 4 - Recursos

Esta seção deve descrever os recursos humanos, de ambiente de teste (hardware e software) e de ferramentas de automatização de testes necessários para execução dos testes que devem ser descritos nas subseções que seguem.

### 4.1 - Ambiente de teste - Software e Hardware

Descreva aqui o hardware e sua configuração, e o software. Por exemplo, o sistema operacional, browsers, servidor web, etc.

### 4.2 - Ferramenta de teste

Descreva aqui as ferramentas específicas de teste usadas no projeto.

## 5 - Cronograma

<b>Tipo de teste</b>	<b>Duração</b>	<b>Data de início</b>	<b>Data de término</b>
<b>Planejar teste</b>		dd/mm/aaaa	dd/mm/aaaa
<b>Projetar teste</b>		dd/mm/aaaa	dd/mm/aaaa
<b>Implementar teste</b>		dd/mm/aaaa	dd/mm/aaaa
<b>Executar teste</b>		dd/mm/aaaa	dd/mm/aaaa
<b>Avaliar teste</b>		dd/mm/aaaa	dd/mm/aaaa