**PLANO DE TESTE**

**RESERVAS DE SALAS**

versão 1.0

**## Histórico das alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor(a) |
| dd/mm/aaaa | x.x | Release inicial | Danielle Farias |

**## 1 - Introdução**

Este documento descreve 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, você deve:

- Identificar informações de projeto existentes 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.

**Requisitos Funcionais**

RF1: Cadastro de salas

O sistema deve permitir o cadastro das salas disponíveis para reserva, com informações como nome, capacidade, localização e recursos disponíveis.

RF2: Visualização de salas disponíveis

O sistema deve exibir uma lista das salas disponíveis para reserva, mostrando informações relevantes, como capacidade, disponibilidade de recursos e horários livres.

RF3: Reserva de salas

Os usuários devem poder selecionar uma sala disponível e fazer uma reserva para uma data e horário específicos

.

RF4: Confirmação de reserva

O sistema deve fornecer uma confirmação imediata após a reserva ser feita, exibindo os detalhes da reserva, incluindo sala, data, horário e informações adicionais.

RF5: Visualização das reservas

Os usuários devem poder visualizar suas reservas atuais, incluindo detalhes como sala, data, horário e observações.

RF6: Cancelamento de reservas

Os usuários devem poder cancelar suas reservas existentes, desde que a data e o horário da reserva ainda não tenham passado.

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

**### Casos de uso:**



**### Requisitos não-funcionais:**

**Requisitos Não Funcionais**

RNF 1: Acesso por login e senha.

RNF2: Deverá possuir diferentes perfis de usuário.

RNF3: Compatibilidade com diferentes plataformas e dispositivos.

RNF4: Interface simples e objetiva.

RNF5 : Deverá estar disponível 24 horas por dia, 7 dias por semana.

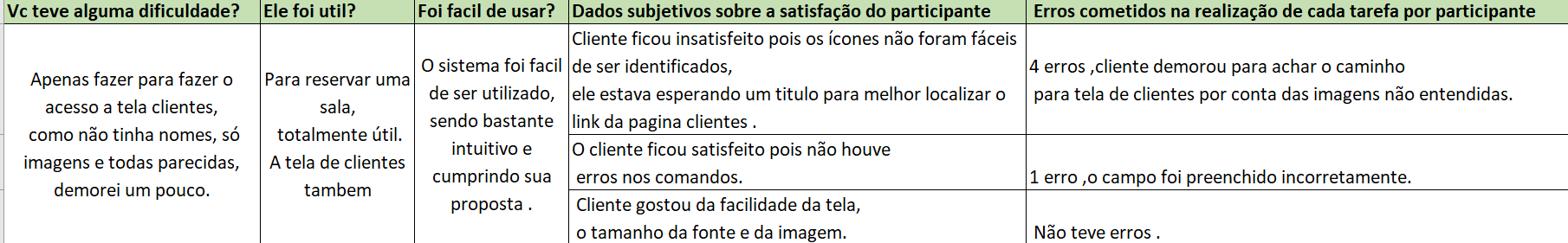
**# 3 - Tipos de teste**

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 usadas 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 adéquam 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 performance e Teste de interface de usuário:

  
  
- Teste de carga;

- Teste de stress;

- Teste de instalação;  
  
- Entre outros.  
  
**3.1 - Métodos da Classe**

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

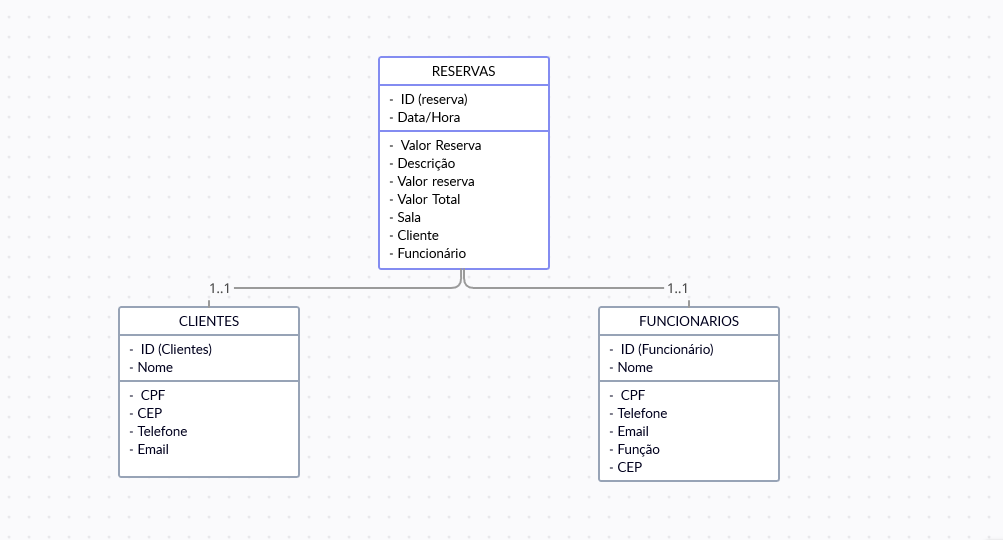
|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | Descreva aqui o objetivo | | |
| Técnica | Manual (x) | Automática (x) | |
| Estagio do teste | Integração () Sistema () | Unidade ( ) | Aceitação( ) |
| Abordagem do teste | Caixa branca () | Caixa preta () | |
| Responsável(is) | Programador(es) ou equipe de testes | | |

**Clientes:**

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | CRUD Clientes | | |
| técnica | Manual (x) | Automática ( ) | |
| Estagio do teste | Integração ( )  Sistema ( x ) | Unidade ( ) | Aceitação( ) |
| Abordagem do teste | Caixa branca () | Caixa preta ( x ) | |
| Responsável(is) | Lorhan , Miguel e Yasmin | | |

**Reservas:**

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | CRUD Reservas | | |
| Técnica | Manual (x) | Automática ( ) | |
| Estagio do teste | Integração ( )  Sistema ( x ) | Unidade ( ) | Aceitação( ) |
| Abordagem do teste | Caixa branca () | Caixa preta ( x ) | |
| Responsável(is) | Lorhan , Miguel e Yasmin | | |

**3.2 – Persistência de dados.**  
  


|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | Verificar se os dados são mantidos após um desligamento súbito do sistema | | |
| técnica | Manual (x) | Automática ( ) | |
| Estagio do teste | Integração ( ) Sistema ( x ) | Unidade ( ) | Aceitação( ) |
| Abordagem do teste | Caixa branca ( ) | Caixa preta ( x ) | |
| Responsável(is) | Lorhan , Miguel e Yasmin | | |

**3.3 – Teste de Integração dos componentes.**

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | Verificação de todos os métodos e classes fazem integração entre si para uma sequência de ações | | |
| Técnica | Manual (x) | Automática (x ) | |
| Estagio do teste | Integração (x ) Sistema ( ) | Unidade ( ) | Aceitação( ) |
| Abordagem do teste | Caixa branca (x ) | Caixa preta ( x ) | |
| Responsável(is) | Lorhan , Miguel e Yasmin | | |

**RECURSOS  
  
4.1 – Ambientes de teste – Software e Hardware.**  
  
Processador: Intel(R) Core(TM) i7-10700T CPU @ 2.00GHz 1.99 GHz  
Memória RAM: 32,0 GB  
Sistema Operacional: Windows 11 PRO  
Servidor WEB: Node.js  
Banco de dados: MongoDB   
Navegador: Google Chrome  
  
**4.2 – Ferramentas de teste.**  
  
Ferramentas usadas: Jest  
  
CRONOGRAMA  
  
