TCE-QA : TESTES

PLANO DE TESTES

**<<nome\_do\_projeto>>**

*[Observações:*

*- O texto em azul exibido entre colchetes e em itálico foi incluído para orientar o autor e deve ser* ***excluído*** *antes da publicação do documento.*

*- Os textos em azul entre sinais de <> representam substituições a serem feitas, por exemplo, <nome do coordenador de projetos.>. E devem ter a cor alterada para preto.*

*- O nome do arquivo deve seguir a seguinte sintaxe: PlanoDeTestes\_NomeProduto : Fonte: PRODEV-TCE/RN*

## HISTÓRICO DE REVISÃO

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 16/12/2019 | 1.0 | Criação do template | Rebecca Betwel |
| 01/10/2020 | 2.0 | Inclusão de índice; Adição de novos tipos de teste; Melhor descrição do documento | Ana Karoliny Queiroz |
| 23/10/2020 | 2.1 | Atualização do template | Jeckson Victor |

## ÍNDICE

1. Objetivos
2. Escopo
3. Requisitos do Teste
4. Estratégia de Teste
   1. Tipos de Teste
      1. Teste do Sistema (Teste Funcional)
      2. Teste da Interface
      3. Teste de Desempenho
      4. Teste de Carga
      5. Teste de Estresse
   2. Ferramentas
5. Recursos
6. Riscos e Restrições
7. Produtos Gerados
8. Tarefas do projeto

## OBJETIVOS

Este Documento descreve o plano para testar o sistema <nome\_do\_sistema>. Este documento de Plano de Teste suporta os seguintes objetivos:

* Identificar informações existentes do projeto e os componentes de software que devem ser testados
* Listar os requisitos de teste recomendados (nível alto).
* Recomendar e descrever as estratégias de teste a serem empregados.
* Identificar os recursos requeridos e fornecer uma estimativa dos esforços de teste.
* Listar os elementos de produto de trabalho das tarefas de teste.

## ESCOPO

Este plano de teste é aplicado aos testes de integração e de sistema que serão conduzidos durante a fase de Integração do <nome\_do\_sistema>.

Os objetos em teste deste produto de software englobam:

* API: [http://gitlab.tce.govrn/<nome\_do\_sistema>/api](http://gitlab.tce.govrn/%3Cnome_do_sistema%3E/api)
* Client: [http://gitlab.tce.govrn/<nome\_do\_sistema>/client](http://gitlab.tce.govrn/%3Cnome_do_sistema%3E/client)
* UCs
* Telas do Projeto
* Padrões usados na diretoria

## REQUISITOS DO TESTE

A lista a seguir identifica os itens (casos de uso, requisitos funcionais, requisitos não funcionais) que foram identificados como alvos do teste. Essa lista representa o que será testado. Detalhes sobre cada teste serão determinados posteriormente à medida que os casos de teste forem identificados e os Scripts de teste forem desenvolvidos.

* Teste do Sistema (Teste Funcional):
  + Verificar Caso de Uso Login
  + Verificar Caso de Uso
* Teste de Interface:
  + Verificar a facilidade de navegação utilizando um conjunto de amostras de telas
  + Verificar se as telas de amostra estão em conformidade com os padrões da GUI
* Teste de Desempenho:
  + Verificar o tempo de resposta para acesso ao sistema
  + Verificar o tempo de resposta para login remoto
  + Verificar o tempo de resposta para cadastro de <nome\_da\_funcionalidade(s)>
* Teste de Carga:
  + Verificar a resposta do sistema quando estiver X Jurisdicionado com login efetuado
  + Verificar a resposta do sistema quando existir 50 acessos simultâneos à <nome\_da\_funcionalidade(s)>
* Teste de Estresse:
  + Verificar a resposta do sistema durante o uso no horário de pico.
  + Verificar a resposta do sistema durante a quantidade máxima (definida pelo teste de carga) de logins de Jurisdicionados.

## ESTRATÉGIA DE TESTE

A estratégia de Teste apresenta a abordagem recomendada para o teste do Sistema. A seção anterior dos Requisitos de Testes descrevia *o que* será testado; esta descreve *como* será testado.

As principais considerações para a estratégia de teste são as técnicas a serem utilizadas e o critério para saber quando o teste está concluído.

Além das considerações fornecidas para cada teste a seguir, o teste deve ser executado apenas utilizando bancos de dados conhecidos e controlados, em ambientes protegidos.

A estratégia de teste a seguir é genérica por natureza e foi desenvolvida para ser aplicada aos requisitos listados na seção 4 deste documento.

## Tipos de Teste

## Teste do Sistema (Teste Funcional)

Os testes do aplicativo devem ter foco em quaisquer requisitos de destino que possam ser rastreados diretamente para casos de uso (ou funções de negócios) e regras de negócios. A meta desse teste é verificar a adequada aceitação, o processamento e a recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa ou preta, ou seja, verificar o sistema (e seus processos internos) e analisar a saída (resultados).

Objetivo do Teste: Assegurar a navegação correta do aplicativo, além da entrada, processamento e recuperação de dados.

Técnica: Executar cada caso de uso, fluxo de caso de uso ou função, utilizando dados válidos e inválidos, para verificar o seguinte:

* Os resultados esperados serão obtidos quando forem utilizados dados válidos.
* As mensagens de erro / aviso apropriados sejam exibidas quando dados inválidos forem utilizados.
* Cada regra de negócio será adequadamente aplicada.

Critérios de conclusão: Todos os testes planejados foram executados. Todos os defeitos identificados foram tratados.

## Teste da Interface

Os testes da interface verificam a interação de um usuário com o sistema. A meta do Teste de UI é assegurar que a interface com o usuário forneça ao usuário o acesso e a navegação adequados por meio das funções dos aplicativos. Além disso, o teste de UI assegura que os objetos contidos na UI funcionem conforme esperado e estejam em conformidade com os padrões corporativos ou do segmento de mercado.

Objetivo do Teste:

* A navegação pelo sistema reflete os requisitos e funções de negócios, incluindo a navegação janela a janela, campo a campo e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras).
* Objetos e características da janela, tais como menus, tamanho, posição, estado e foco estão em conformidade com os padrões.

Técnica: Criar/Modificar testes para cada janela a fim de verificar a navegação adequada e os estados de objeto para cada janela e objeto do sistema.

Critérios de Conclusão: Verificação com êxito de cada janela permanecer consistente ou dentro do padrão aceitável.

## Teste de Desempenho

O teste de desempenho mede tempos de resposta, taxas de transição e outros requisitos sensíveis ao tempo. A meta do teste de desempenho é verificar e validar se os requisitos de desempenho foram alcançados. O teste de desempenho normalmente é executado várias vezes, cada uma utilizando uma “carga de segundo plano” diferente no sistema, O teste Inicial deve ser executado com uma carga “nominal”, semelhante a carga normal observada (ou prevista) no sistema de destino. Um segundo teste de desempenho é executado utilizando uma carga de pico.

Além disso, os testes de desempenho podem ser utilizados para traçar o perfil e ajustar o desempenho de um sistema com uma função de condições, como a carga de trabalho ou configurações do hardware.

Objetivos do Teste: Validar o tempo de resposta do sistema para funções de negócios ou transações designadas sob as duas condições a seguir:

* Volume normal previsto
* Volume de pior caso previsto.

Técnica:

* Utilizar scripts de teste desenvolvidos para os Teste de sistema
* Modificar arquivos de dados (a fim de aumentar o número de transações) ou modificar scripts a fim de aumentar o número de iterações ocorrido em cada transação.
* Os scripts devem ser executados em uma máquina (o melhor é avaliar o desempenho de um único usuário, uma única transação) e repetidos com vários clientes (virtuais ou reais)

Critérios de Conclusão:

* Transação Única/ Usuário Único: conclusão com êxito dos scripts de teste sem nenhum defeito e na alocação de tempo esperada / requerida (por transação)
* Várias Transações/ Vários usuários: Conclusão com êxito dos scripts de teste sem nenhum defeito e dentro de alocação de tempo aceitável.

## Teste de Carga

As medidas do teste de carga sujeitam o sistema em teste a cargas de trabalho variáveis para avaliar a capacidade do sistema em continuar a funcionar corretamente sob essas diferentes cargas de trabalho. A meta desse teste de carga é determinar e assegurar que o sistema funcione adequadamente com uma carga de trabalho superior à carga máxima esperada. Além disso, o teste de carga avalia as características de desempenho (tempos de resposta, taxas de transação e outros aspectos sensíveis ao tempo).

Objetivo do teste: Verificar o tempo de resposta do sistema para casos de negócios ou transações designadas sob condições de carga de trabalho variáveis.

Técnica: Modificar os arquivos de dados (a fim de aumentar o número de transações) ou os testes a fim de aumentar o número de vezes que cada transação ocorre.

Critérios da Conclusão: Várias transações/vários usuários: Conclusão com êxito dos testes sem nenhum defeito e dentro de alocação de tempo aceitável.

## Teste de Estresse

O teste de estresse foi projetado para localizar erros devido a falta de recursos ou competição por recursos. Pouca memória ou espaço em disco podem revelar defeitos no software que não são aparentes sob condições normais. Outros defeitos podem resultar da competição por recurso compartilhado, como bloqueios de banco de dados ou largura da banda de rede. O teste de estresse identifica a carga de pico que o sistema pode manipular.

Objetivo do teste: Verificar que o sistema funciona corretamente e sem erros sob as seguintes condições de estresse:

* Pouca ou nenhuma memória disponível no servidor.
* Número máximo (real ou fisicamente capaz) de clientes conectados (ou simulados).
* Vários usuários executando as mesmas transações com os mesmos dados / contas.
* Conjunto / volume de transações no pior caso.

Técnica: Utilizar os testes desenvolvidos para o teste de desempenho. Para os testes de estresse restantes, deverão ser utilizados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto/volume de transações no pior caso.

Critérios de conclusão: Todos os testes planejados são executados e os limites do sistema especificados são alcançados/ excedidos sem falha no sistema (ou as condições sob as quais a falha do sistema ocorre estão fora das condições específicas).

## Ferramentas

As seguintes ferramentas serão empregadas para o teste do sistema:

|  |  |  |
| --- | --- | --- |
|  | Ferramenta | Versão |
| Teste Sistema (Funcional) | Behave Python | 1.2.6 |
| Teste de Desempenho | Jmeter | 5.3 |

## RECURSOS

Esta seção apresenta os recursos recomendados para o teste do <nome\_do\_sistema>

1. Recursos Humanos

|  |  |  |  |
| --- | --- | --- | --- |
| **Função no projeto** | **Responsável** | **Atividades** | **Testes atribuídos** |
| Programador(es) | Nome Nome | Desenvolver a aplicação | Testes Unitários |
| Analista de Teste | Caio Cesar |  | - |
| Coordenador de Testes | Jeckson Victor |  | - |
| Testador(es) | Nome |  | Teste do Sistema  Teste da Interface  Teste de Desempenho  Teste de Carga  Teste de Estresse |
| Gerente de Projetos | André Gustavo |  | - |
| Líder do projeto | Nome |  | - |

1. Recursos Computacionais

|  |  |  |  |
| --- | --- | --- | --- |
| **Recurso** | **Descrição** | | |
| Computador Cliente | https://docs.google.com/spreadsheets/d/1DmQREU\_oEvCy4f69ldbBqRrBzcnhtuYxv\_4DR30rbtY/edit#gid=627511950 | | |
| Servidor da Aplicação |
| Servidor do Banco de Dados |
| Criação de Relatórios | Google Drive (docs) | | |
| Browser Cliente | Google Chrome (Versão variante) | | |
| SO Cliente | Windows 10 | | |

1. RISCOS E RESTRIÇÕES

Não ter ambiente PRÓPRIO de QA acarreta um risco ao objetivo dos testes, haja vista que os testes devem ter seu próprio contexto devido sua particularidade única de ser um verificador e um validador.

## PRODUTOS GERADOS

Suíte de teste: Este artefato está agregado ao projeto de testes automatizado, os testes que pertencem a mesma suíte possuem uma mesma tag, sendo a que deve ser incorporada ao processo do CI/CD a suite @all, onde estão indicados todos os testes das funcionalidades que foram validadas com sucesso e que podem seguir para release.

Planilha de dados de teste: Deve existir uma planilha elaborada pelo testador sob orientação do coordenador de teste, que contenha todos os dados de teste que foram levados em consideração na implementação dos testes automatizados, seguindo as boas práticas orientadas no guia de testes (PROTESTE) assim como os demais testes que foram necessários para a validação ou verificação do item em questão.

Relatórios: Devem existir relatórios por sprints elaborados pelo testador resumindo as ações de testes daquela sprint, englobando eventuais bugs encontrados, indicando os objetos testados assim como sugestões de melhoria e considerações finais.

Projeto de teste de software: para cada produto de software deve existir um projeto associado, isso indica que tanto API’s quanto websites devem ter seus próprios projetos de teste.

Os projetos de testes de software a serem implementados no presente projeto usarão o framework Behave, usando a linguagem de programação Python 3.

## TAREFAS DO PROJETO

A seguir são mostradas as tarefas relacionadas ao teste do SIAI Obras e Serviços.

**Planejar Teste**

* Identificar Requisitos para o Teste
* Avaliar Risco
* Desenvolver Estratégia de Teste
* Identificar Recursos de Teste
* Criar Planejamento
* Gerar Plano de Teste

**Projetar Teste**

* Análise de Carga de Trabalho
* Desenvolver Conjunto de Teste
* Identificar e Descrever Casos de Teste
* Identificar e Estruturar Scripts de Teste
* Revisar e Acessar Cobertura de Teste

**Implementar Teste**

* Configurar Ambiente de Teste
* Registrar ou Programar Scripts de Teste
* Identificar funcionalidade específica do Teste no design e modelo de implementação
* Estabelecer Conjuntos de Dados Externos

**Executar Teste**

* Executar Script de Teste
* Avaliar Execução do Teste
* Recuperar-se de Teste Interrompido
* Verificar os Resultados
* Investigar Resultados Inesperados
* Registrar Defeitos

**Avaliar Teste**

* Avaliar a Cobertura dos Casos de Teste
* Avaliar Cobertura do Código
* Analisar Defeitos
* Determinar se os Critérios de Conclusão do Teste e os Critérios de Êxito foram alcançados
* Criar Relatório de Avaliação do Teste