|  |  |
| --- | --- |

**Plano de Testes**

Curso Técnico em Desenvolvimento de Sistemas

| **Professor:Daniel Henrique**  Daniel Rodrigo Henrique | **Aluno:Renato Sussumu Takahashi Junior**  **Diego Silva Cabral**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --- | --- |

**Maio / 2023**

**Histórico de Revisões**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 07/04/2023 | 1.2 | O sistema consegue se conectar ao banco de dados e consegue guardar informações do usuário e outros. | Renato Sussumu Takahashi Junior |

1. **Introdução**

* 1. **Objetivos**

Esse documento do Plano de Testes compõe-se dos seguintes objetivos:

• Identificar informações de projeto existentes e os componentes de software que devem ser testados.

• Listar os Requisitos a Testar recomendados (alto nível).

• 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.

* 1. **Escopo**

O sitema passará pelos testes unitário, de integração e de sistema. Os testes unitários e de integração vão lidar com a qualidade funcional, das bases de dados, interface gráfica e do controle de acesso; enquanto que os testes de sistema tratarão as questões de performance.

Para a execução dos testes serão utilizadas máquinas do laboratório de informática, a fim de garantir a previsibilidade de performance e compatibilidade.

Os testes mais críticos serão os testes de banco de dados, que compõe a maior parte do sistema e os de performance:

1. Testaremos o tempo de resposta para operações que envolvam dados;
2. E também a correta atualização do banco de dados para as funções de cadastrar, atualizar e remover.
   1. **Identificação de Projeto**

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

| **Documento** | **Criado ou Disponível** | **Recebido ou Revisado** |
| --- | --- | --- |
| Especificação de Requisitos | ■ Sim □ Não | ■ Sim □ Não |
| Diagrama Caso de Uso | ■ Sim □ Não | ■ Sim □ Não |
| Diagrama de Classes | ■ Sim □ Não | ■ Sim □ Não |
| Diagrama de Objetos | ■ Sim □ Não | ■ Sim □ Não |
| Protótipo | ■ Sim □ Não | ■ Sim □ Não |

1. **Requisitos a Testar**

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

* 1. **Teste do Banco de Dados**
* Verifique que as informações do usuário podem ser cadastradas, consultadas e removidas.

As informações podem ser armazenadas no banco de dados , infelizmente não podem ser consultadas ou removidas dentro do sistema

* Verifique que as informações específicas de cada usuário podem ser acompanhadas.

As informações podem ser acompanhadas dentro do banco de dados .

* Verifique que as informações úteis cadastradas possam ser consultadas.

Dentro do banco de dados os dados podem ser cadastrados.

* Verifique que o usuário cadastrado pode acessar o sistema.

O usuário não pode acessar estas informações dentro do sistema

* Verifique que o sistema é capaz de buscar e manter atualizadas as informações.

O sistema não consegue manter atualizadas as informações

* 1. **Teste Funcional**
* Verifique que as informações úteis obtidas pelo sistema responsável são automaticamente e periodicamente atualizadas.
* Verifique que qualquer usuário pode acessar sua própria conta através de login e senha.
* **Criar mais casos de teste referente aos seus requisitos levantados!**
  1. **Teste do Ciclo de Negócios**

Nenhum.

* 1. **Teste da Interface do Usuário**
* Navegue através de todos os use cases, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada.

A tela de interface pode ser facilmente interpretada e facilmente utilizada

* Verifique se o sistema possui uma tela de ajuda e se funciona.

No meu projeto tive a ideia inicial de implementar a tela de ajuda contudo tive problemas na comunicação com o banco de dados e acabou que não consegui ter o tempo necessário para colocar no meu sistema.

* **Criar mais casos de teste referente aos seus requisitos levantados!**
  1. **Perfil da Performance**

Verifique o tempo de resposta da rede interna, do servidor em relação aos terminais.

O tempo de resposta e rápida.

Verifique o tempo de consulta/atualização do sistema de informações úteis.

Verifique que o tempo de resposta para operações que envolvam dados não ultrapassam 30 segundos.

O tempo de resposta para operações não ultrapassam 30 segundos

* 1. **Teste de Carga**

Verificar a resposta do sistema com 10 usuários.

Verificar a resposta do sistema com 50 usuários.

Verificar a resposta do sistema com 100 usuários.

Verificar a resposta do sistema com 200 usuários.

Verificar a resposta do sistema com 500 usuários.

* 1. **Teste de Stress**

Nenhum.

* 1. **Teste de Volume**

Nenhum.

* 1. **Teste de Segurança e de Controle de Acesso**

Verificar que usuários não cadastrados não podem acessar informações restritas aos cadastrados.

Não podem

Verificar que além do administrador, ninguém mais pode inserir, atualizar ou remover dados do sistema.

Verificar que os usuários apenas vêem suas próprias informações.

sim

Verificar que os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário.

Verificar que a atualização do sistema pode ser feita apenas a partir da rede interna.

Não consegue fazer a atualização por meio da rede interna

* 1. **Teste de Falha/Recuperação**

Nenhum.

* 1. **Teste de Instalação**

Verifique que a instalação do sistema ocorre normalmente em todas as máquinas.

Pode ser instalado em todas as maquinas

Verifique que qualquer terminal do sistema do hotel é capaz de rodar o sistema normalmente.

Verifique que o sistema é capaz de obter e atualizar as informações úteis a que se propõe a disponibilizar.

Verifique que a atualização dos dados no servidor se reflete em todos os terminais.

Verifique que o espaço disponível em disco para informações deve ser capaz de armazenar todos os dados/atualizações que forem cadastrados.

1. **Estratégia de Teste**
   1. **Tipos de Teste**

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

* + 1. *Teste de Integridade de Dados e do Banco de Dados*

| Objetivo do Teste: | Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados. |
| --- | --- |
| Técnica: | * Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos. * Inspecionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas. |
| Critério de Finalização: | Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados. |
| Considerações Especiais: | * O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente nas base de dados * Processos devem ser invocados manualmente * Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis. |

* + 1. *Teste de Função*

| Objetivo do Teste: | Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação. |
| --- | --- |
| Técnica: | Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:   * Os resultados esperados ocorrem quando dados válidos são usados * As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados. * Cada regra de negócio é aplicada apropriadamente |
| Critério de Finalização: | * Todos os testes planejados foram executados. * Todos os defeitos identificados foram tratados. |
| Considerações Especiais: | Nenhum |

* + 1. *Teste da Interface do Usuário*

| Objetivo do Teste: | Verificar o seguinte:   * A navegação através dos alvos de teste reflete as funções e os requisitos do negócio apropriadamente, incluindo janela-a-janela, campo-a-campo, e o uso de métodos de acesso (tecla tab, movimentos do mouse, teclas aceleradoras) * Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões. |
| --- | --- |
| Técnica: | Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação. |
| Critério de Finalização: | É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis. |
| Considerações Especiais: | Nem todas as propriedades para objetos personalizados e terceirizados podem ser acessadas. |

* + 1. *Teste de Performance*

| Objetivo do Teste: | Verificar que os comportamentos de performance para as transações designadas ou funções de negócio sob as seguintes condições:   * Carga de trabalho normal antecipada * Carga de trabalho no pior caso antecipada |
| --- | --- |
| Técnica: | * Usar Procedimentos de Teste desenvolvidos para Teste da Função ou Ciclo de Negócio * Modificar os arquivos de dados para aumentar o número de transações ou os scripts para aumentar o número de iterações que ocorre a cada transação. * Scripts devem ser rodados em uma máquina (melhor caso para comparar um único usuário, uma única transação) e ser repetidas com múltiplos clientes (virtual ou real, ver Considerações Especiais abaixo). |
| Critério de Finalização: | Único usuário ou transação: finalização com sucesso dos scripts de testes sem nenhuma falha e dentro da alocação de tempo por transação esperada ou requisitada.  Múltiplas transações ou usuários: finalização bem sucedida dos scripts de teste sem qualquer falha e dentro da alocação de tempo aceitável. |
| Considerações Especiais: | Um teste abrangente de performance inclui ter uma carga de trabalho no servidor.  Há vários métodos que podem ser usados para executar isso, incluindo:   * “Direcionar transações” diretamente para o servidor, usualmente na forma de chamadas SQL. * Criar carga de usuário “virtual” para simular muitos clientes, normalmente várias centenas. Ferramentas de Emulação de Terminal Remoto (RTE) são usadas para atingir essa carga. Essa técnica também pode ser usada para carregar uma rede com “tráfego”. * Usar múltiplos clientes físicos, cada um rodando scripts de teste para gerar uma carga no sistema.   O teste de performance deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite controle total e mensuração precisa.  As bases de dados usadas para o Teste de Performance devem ser ou do tamanho real ou proporcionalmente iguais. |

* + 1. *Teste de Carga*

| Objetivo do Teste: | Verifique o tempo de resposta para as transações designadas ou casos de negócio sob condições variantes de carga de trabalho. |
| --- | --- |
| Técnica: | * Use testes desenvolvidos para o Teste do Ciclo de Negócio ou Função. * Modifique os arquivos de dados para aumentar o número de transações ou os testes para aumentar o número de vezes que cada transação ocorre. |
| Critério de Finalização: | Múltiplas transações ou usuários: finalização bem sucedida dos testes sem qualquer falha e dentro da alocação de tempo aceitável. |
| Considerações Especiais: | * O teste de carga deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite o controle total a e mensuração precisa. * As bases de dados usadas para os testes de carga devem ou ser do tamanho real ou igualmente dimensionadas. |

* + 1. *Teste de Segurança e Controle de Acesso*

| Objetivo do Teste: | * Segurança do Nível de Aplicação: Verifique que um ator pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão. * Segurança do Nível de Sistema: Verifique que apenas aqueles atores com acesso ao sistema e aplicações têm permissão de acessá-los. |
| --- | --- |
| Técnica: | * Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão. * Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicos para cada tipo de usuário. * Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados. * Acesso de Nível de Sistema: Ver Considerações Especiais abaixo. |
| Critério de Finalização: | Para cada tipo de ator conhecido as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores. |
| Considerações Especiais: | O Acesso ao sistema deve ser revisado ou discutido com o administrador de rede ou de sistema apropriado. Esse teste pode não ser necessário já que ele pode ser uma função da administração da rede ou sistema. |

* + 1. *Teste de Instalação*

| Objetivo do Teste: | Verifique que os alvos de teste instalam apropriadamente em cada configuração de hardware necessária sobre as seguintes condições:   * Uma nova instalação, em um nova máquina, que nunca fora anteriormente instalada com o sistema. * atualização, numa máquina onde o sistema já fora previamente instalado, para a mesma versão * atualização, numa máquina que já disponha do sistema instalado, de uma versão mais velha |
| --- | --- |
| Técnica: | Manualmente ou desenvolva scripts automatizados, para validar a condição da máquina alvo – novo sistema nunca instalado; sistema na mesma versão ou versão mais velha já instalada.  Começar ou executar a instalação  Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações. |
| Critério de Finalização: | As transações do sistema executam de forma bem sucedida, sem falha. |
| Considerações Especiais: | Quais transações do sistema devem ser selecionadas para abranger um teste de confiança de que a aplicação foi instalada de forma bem sucedida e que nenhum componente importante de software está faltando? |

* 1. **Ferramentas**

As seguintes ferramentas serão empregadas para esse projeto:

|  | Ferramenta | Vendedor |
| --- | --- | --- |
| Gerenciamento de Teste | Rational RequisitePro  Rational Unified Process | Rational |
| Projeto de Teste | Rational Rose | Rational |
| Gerenciamento de Projeto | Microsoft Project  Microsoft Word | Microsoft |
| Ferramentas do SGBD | MySQL Workbench | Oracle |

1. **Recursos**

Essa seção apresenta os recursos recomendados para o projeto do sistema, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

* 1. **Trabalhadores**

Essa tabela mostra as suposições de recrutamento para o projeto.

| Recursos Humanos | | |
| --- | --- | --- |
| Trabalhador | Recursos Mínimos Recomendados | Responsabilidades Específicas ou Comentários |
| Gerente de Teste,  Gerente do Projeto de Teste | Renato Takahashi | Fornece supervisionamento gerencial.  Responsabilidades:   * provê direcionamento técnico * adquire recursos apropriados * fornece relatórios de gerenciamento |
| Test Designer | Renato Takahashi | Identifica, prioriza, e implementa os casos de teste.  Responsabilidades:   * gera o plano de teste * cria o modelo de teste * avalia a efetividade do esforço de teste |
| Testador | Renato Takahashi | Executa os testes.  Responsabilidades:   * executar os testes * registrar os resultados * reestabelecer-se dos erros * documentar solicitações de mudança |
| Administrador do Sistema de Teste | Renato Takahashi | Garante que o ambiente e os bens de teste sejam gerenciados e mantidos.  Responsabilidades:   * administrar o sistema de gerenciamento teste * instalar e gerenciar o acesso do trabalhador ao sistema de testes |
| Gerente do Banco de Dados,  Administrador do Banco de Dados | Renato Takahashi | Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos.  Responsabilidades:   * administrar os dados de teste (base de dados) |
| Designer | Renato Takahashi | Identifica e define as operações, atributos, e associações das classes de teste.  Responsabilidades:   * identificar e definir as classes de teste * identificar e definir os pacotes de teste |
| Implementador | Renato Takahashi | Implementa e faz os testes unitários das classes e pacotes de teste.  Responsabilidades:   * cria as classes e pacotes de teste implementados no modelo de teste |

* 1. **Sistema**

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

| Recursos do Sistema |
| --- |
| Servidor de Banco de Dados |
| — MySQL Server |
| Terminais Clientes |
| —2 PCs (conectados via LAN e à internet) |
|  |
| Repositório de Testes |
| —1 PC |
|  |

1. **Cronograma**

| **Milestone** | **Data de Início** | **Data de Término** |
| --- | --- | --- |
| Planejar Teste | 07/04/2023 | 07/04/2023 |
| Projetar Teste | 07/04/2023 | 07/04/2023 |
| Implementar Teste | 07/04/2023 | 07/04/2023 |
| Executar Teste | 07/04/2023 | 07/04/2023 |
| Avaliar Teste | 07/04/2023 | 07/04/2023 |