Plano de Teste

|  |  |
| --- | --- |
| ***Imobiliária comercial - locação de salas comerciais flexíveis*** | |
| **Gestor do Projeto** | **Gerente de Projeto** |
| Brenda Rodrigues | Matheus Henrique Sans |
| *brenarodriguesnl@gmail.com* | *sans.matheus@icloud.com* |
| 3193145297 | 31991753580 |

|  |
| --- |
| ***Objetivo deste Documento*** |
| Este documento tem como objetivo planejar as atividades a serem realizadas, definir os métodos a serem empregados e estabelecer métricas e formas de acompanhamento do processo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Histórico de Revisão*** | | | | |
| **Data** | **Demanda** | **Autor** | **Descrição** | **Versão** |
| 27/03/2020 | 0001 | Matheus Henrique Sans | Inicio do Projeto | 1.0 |

* **INTRODUÇÃO**

Documento de orientação para projeto de uma empresa imobiliária comercial que oferece em um aplicativo mobile espaços de trabalho compartilhados para startups e pequenas empresas para serem alugados

* **Escopo**

01 - Efetuar login

O sistema deve validar se o usuário está cadastrado e com todos os documentos validados.

02 - Visualizar imóveis disponíveis

O sistema deve mostrar todas as opções de imóveis disponíveis para locação.

03 - Detalhar e escolher imóvel

O sistema deve detalhar o imóvel escolhido, suas imagens cadastradas, horários disponíveis, e a opção de seleciona-lo.

04 - Pagamento

O sistema deve abrir uma tela terceirizada por api’s para realização do pagamento.

05 - Visualizar informações e histórico de reserva

O sistema deve exibir um histórico com todos os imóveis que o usuário já alugou, e os detalhes da sua compra atual, se existir.

* **ESTÁGIOS DE TESTE**

Os testes podem ser divididos em 3 partes, é importante o frisar a necessidade dos 3 testes para alcançarmos uma melhor qualidade do sistema.

**No desenvolvimento:**

**- Testes Unitário:** Realizado durante a construção do código. Consiste em validar dados válidos e inválidos via I/O sendo aplicado por desenvolvedores ou analistas de teste. Realizado pelos desenvolvedores.

**- Teste de Integração:** Verifica os requisitos funcionais, de desempenho e de confiabilidade na modelagem do sistema. Com ele é possível descobrir erros de interface entre os componentes do sistema. Realizado pelos desenvolvedores com skill em projetos.

.

**Na entrega:**

**- Teste de Sistema:** Testa todas as funcionalidades do sistema. Realizado pelos desenvolvedores com skill em qualidade.

**Na cliente:**

**- Teste de Sistema:** Testa todas as funcionalidades do sistema no ambiente real de utilização, seus feedbacks são importantes para corrigir erros que podem ter ocorrido e não identificado anteriormente. Realizado pelos usuários.

* **TIPOS DE TESTES**

• Estresse: Identifica problemas na aplicação que se tornam aparente somente diante das condições de super utilização.

• Acessibilidade: O teste de usabilidade é uma técnica que visa avaliar o sistema do ponto de vista do usuário final. Levando em consideração a estética e a facilidade de uso.

• Segurança: Busca garantir que o software desenvolvido esteja bem estruturado internamente, portanto, funcionando corretamente.

• Regressão: O teste de regressão é uma técnica aplicável a cada alteração realizada no software, cujo objetivo é garantir que as mudanças realizadas nessa nova versão não gerarão erros em componentes prontos e testados. Consiste em aplicar, antes e depois da alteração, todos os testes que já foram aplicados nas versões anteriores.

• Funcional: Busca garantir que a saída gerada após a entrada dos dados especificados está correta.

• Performance: Busca medir velocidade e capacidade da aplicação.

* **RECURSOS NECESSÁRIOS**
* **Recursos Humanos**

Os recursos humanos necessários para realizar os teste são:

• Desenvolvedores;

• Testers;

• Cliente;

* **Recursos Computacionais**

Os recursos computacionais necessários para realizar os testes são:

• Servidor;

• Computador para desenvolvimento;

• Dispositivo móvel para utilização;

• Internet;

* **RISCOS E RESTRIÇÕES**

• Ataques;

• Falha de Contingência;

• Vulnerabilidades (Inconsistências e erros);

• Falha na API de pagamentos;

• Vulnerabilidade na API de pagamentos (Falsificações);

• Sobrecargas;

* **PRODUTOS GERADOS**

Script de teste segundo a norma IEEE 829;

* **REFERÊNCIAS**

|  |  |  |
| --- | --- | --- |
| **#** | **Documento** | **Versão** |
|  | Técnicas de testes - demoiselle.sourceforge.net |  |