**Plano de Testes para o desafio Grupo OLX: Sistema de locação, compra e venda de imóveis**

**Objetivo:** Validar a funcionalidade do sistema de imóveis, considerando cenários positivos e negativos, a fim de garantir a qualidade e robustez da aplicação.

**Contexto:**

* **Sistema:** https://www.zapimoveis.com.br/
* **Funcionalidade:** Buscar Imóveis
* **Casos de Teste:** Os casos de teste a seguir foram elaborados com base em testes exploratórios, abrangendo diferentes funcionalidades e cenários.

### **Matriz de Rastreamento dos Casos de Teste**

O rastreamento será realizado através de um arquivo .csv pois no momento não há uma tecnologia como o Jira para fazer o controle dos casos de teste como o exemplo abaixo:

|  |  |  |
| --- | --- | --- |
| Número | Nome | Status |
| CT01 | Pesquisar Imoveis que aceitam animais e que estejam mobilhados | Passou |
| CT02 | Pesquisar imoveis para compra | Não Passou |
| CT03 | Pesquisar imoveis para alugar |  |
| CT04 | Denunciar anuncio |  |
| CT05 | Conferir mais ofertas do anunciante |  |
| CT06 | Simular financiamento |  |
| CT07 | Criar alerta de imoveis similares |  |
| CT08 | Falar com anunciante pelo whatsapp |  |
| CT09 | Favoritar anuncio |  |
| CT10 | Compartilhar anuncio |  |
| CT11 | Criar login com dados invalidos |  |
| CT12 | Enviar uma mensagem com dados invalidos |  |

### **Estratégia de Teste**

* **Teste Funcional:** Verificar se cada funcionalidade do sistema está funcionando de acordo com os requisitos.
* **Teste de Regressão:** Garantir que novas funcionalidades não impactem o funcionamento das existentes.
* **Automação de testes**: Garantir a segurança do sistema através dos casos de teste regressivos automatizados, pois testar novamente tudo manual ficará mais dificil a medida que novas funcionalidades e testes para elas são criadas.

### **Técnicas de Teste**

* **Caixa Preta:** Testar o sistema sem conhecimento do código interno.
* **Teste exploratório:** Utilizar o sistema buscando as funcionalidades já entregues a serem testadas.

### **Ambiente de Teste**

* **Hardware:**
* Processador Ryzen 5 3600
* Memória DDR4 HyperXFury 3200mhz
* Fonte Corsair 750 watts serie ouro
* Placa mãe A320m
* Placa de video dedicada RX550 4GB
* **Software:** Navegador Google Chrome, Cypress, plugin esbuild e Cucumber preprocessor (@badeball/cypress-cucumber-preprocessor, @bahmutov/cypress-esbuild-preprocessor).
* **Dados:** Irá considerar o ambiente de produção que não pode ser acessado pelo Cypress, pois não foi fornecido um ambiente de testes para realização deste desafio.

### **Critérios de Aceitação**

* **Funcionalidade:** Todas as funcionalidades descritas nos requisitos devem estar implementadas e funcionando corretamente.
* **Desempenho:** O sistema deve responder às solicitações do usuário em um tempo razoável.
* **Usabilidade:** A interface do usuário deve ser intuitiva e fácil de navegar.
* **Robustez:** O sistema deve ser capaz de lidar com erros e entradas de dados inválidas.

**Observações:**

* **Priorização:** Priorizar os casos de teste de acordo com o risco e a criticidade da funcionalidade.
* **Automatização:** Automatizar os testes sempre que possível para aumentar a eficiência e reduzir o tempo de execução.
* **Cobertura de Código:** Analisar a cobertura de código para garantir que todos os componentes do sistema foram testados.

### **1. Introdução**

* **Objetivo:** Garantir a qualidade do sistema de imóveis.
* **Escopo:** Funcionalidade de Buscar Imóveis.
* **Referências:**

Sommerville, I. **Engenharia de software**. 11. ed. São Paulo: Pearson Education Brasil, 2019.

MYERS, G. J. **A arte do teste de software.** 2. ed. Rio de Janeiro: LTC, 2004.

### **2. Características do Software**

* **Funcionalidades:** Descrever detalhadamente cada funcionalidade do sistema.
* **Plataformas:** Web.
* **Tecnologias:** Listar as tecnologias utilizadas (linguagens de programação, banco de dados, etc.).

### **3. Itens a Serem Testados**

* **Requisitos Funcionais:** Garantir o funcionamento correto das funcionalidades que envolvem a atividade de buscar imóveis.
* **Requisitos Não Funcionais:** Verificar a velocidade das querys da aplicação, buscando melhorias caso eles venham a demonstrar que não são performáticos.
* **Critérios de Aceitação:** Deverá ser verificado pela equipe de gestão e recrutamento do Grupo OLX.

### **4. Características dos Testes**

* **Tipos de Teste:**
  + **Funcional:** Verificar se as funcionalidades atendem aos requisitos.
  + **Não Funcional:** Avaliar desempenho, usabilidade, segurança, etc.
  + **Integração:** Verificar a interação entre diferentes módulos.
  + **Regressão:** Garantir que novas funcionalidades não afetem as existentes.
  + **Usabilidade:** Avaliar a facilidade de uso do sistema.
  + **Exploratório:** Testar o sistema de forma livre para descobrir problemas inesperados.
* **Níveis de Teste:**
  + **Unitário:** Testar unidades individuais de código.
  + **Integração:** Testar a interação entre módulos.
  + **Sistema:** Testar o sistema como um todo.
* **Abordagem de Teste:**
  + **Manual:** Testes executados manualmente por um testador.
  + **Automatizado:** Testes executados por ferramentas automatizadas.

### **5. Itens a Serem Entregues**

* **Relatórios de Teste:** Especificar o formato e conteúdo dos relatórios.
* **Itens Testáveis:** Listar os itens que serão testados (módulos, funcionalidades, etc.).
* **Documentação:** Definir a documentação a ser gerada (planilhas, scripts, etc.).

### **6. Recursos**

* **Recursos Humanos:** Definir os papéis e responsabilidades de cada membro da equipe de testes.
* **Hardware:** Especificar os equipamentos necessários (computadores, dispositivos móveis, etc.).
* **Software:** Listar as ferramentas de teste a serem utilizadas.

### **7. Cronograma**

* **Tarefas:** Detalhar as tarefas de teste e suas dependências.
* **Prazo:** Definir prazos para cada tarefa e para a conclusão do projeto de testes.

### **8. Riscos e Mitigantes**

* **Riscos:** Identificar os possíveis riscos que podem afetar o projeto de testes.
* **Mitigantes:** Definir ações para mitigar os riscos identificados.

### **9. Critérios de Entrada e Saída**

* **Critérios de Entrada:** Definir as condições que devem ser atendidas para iniciar os testes.
* **Critérios de Saída:** Definir os critérios para encerrar os testes.

### **10. Aprovações**

* **Aprovações:** Definir quem irá aprovar o plano de testes.

**Exemplo de Matriz de Rastreamento (mais detalhada):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID Caso de Teste** | **Descrição Detalhada** | **Pré-condições** | **Passos** | **Resultados Esperados** | **Critérios de Aceitação** | **Status** |
| CT01.1 | Pesquisar imóveis que aceitam animais de pequeno porte | Usuário logado, pesquisa configurada para "pequeno porte" | ... | ... | ... | Não Executado |

Exportar para as Planilhas

**Ferramentas para Gerenciar o Plano de Testes:**

* **Planilhas:** Excel, Google Sheets
* **Ferramentas de Gestão de Testes:** TestRail, Zephyr, Xray
* **Ferramentas de Automação:** Selenium, Appium, JUnit

**Dicas Adicionais:**

* **Seja específico:** Quanto mais detalhado o plano, melhor.
* **Revise constantemente:** O plano de testes deve ser atualizado ao longo do projeto.
* **Colabore:** Envolva toda a equipe no desenvolvimento do plano.
* **Utilize templates:** Existem diversos templates disponíveis online para te ajudar a criar seu plano.

**Lembre-se:** Um bom plano de testes é essencial para garantir a qualidade do software e aumentar a confiança dos usuários.

**Gostaria de que eu elabore um modelo mais específico para alguma parte do seu plano?**

**Por exemplo:**

* **Casos de teste detalhados para uma funcionalidade específica.**
* **Matriz de risco completa.**
* **Modelo de relatório de testes.**