

Plano de Teste

1. Sistema de Revenda de Carro JS (SRCJS)

- Verificando o Design de Interface e a Usabilidade do site SRCJS

2. Resumo do Projeto

- O SRCJS é um sistema de gerenciamento de veículos, que tem as principais especificações dos veículos em estoque como, modelo, fabricante, ano, combustível, câmbio, quilometragem, valor comprado, valor de venda. Contando ainda o campo de filtro, onde o usuário pode encontrar as informações de um determinado veículo encontrando pela letra inicial do carro
- A interface da tabela dos veículos tem a coluna de status, onde o vendedor poderá colocar a situação atual do veículo, consiste nos botões de: vendido, disponível e em análise. Tendo também a opção de deletar a linha da tabela com o botão delete, caso o carro já tiver sido vendido
- O usuário poderá adicionar um novo carro na tabela informando todas as especificações obrigatórias do carro como: modelo, fabricante, ano e etc... quando o usuário informar no campo "valor comprado" do veículo, o sistema automaticamente adicionará uma margem de 45% de lucro na coluna "Valor de Venda" na tabela. Ex: o usuário informou que comprou o carro por R\$30.000,00 (trinta mil reais) o sistema colocará o valor de R\$45.000,00 (quarenta e cinco mil reais) na tabela.

2.1 Resumo do Planejamento de teste

- Hipótese: crença inicial sobre a usabilidade do software a ser testado, seja boa ou não
- teste de usabilidade : deseja-se verificar se a usabilidade da interface visual do sistema SRCJS é boa e para isso serão aplicados testes com usuários pertencentes ao público-alvo.
- os testes serão feitos de forma manual, com as ferramentas **Qase** e **Jmeters**

3. Pessoas Envolvidas

- Analista de teste Douglas Figueiredo
- usuários aleatórios para fazerem o teste de usabilidade

4. Funcionalidades a serem testadas

Partindo da premissa que já tenha 5 veículos em estoque adicionados na tabela

4.1- Campo de Filtro

Dado que eu como usuário queira fazer uma busca na tabela

Quando colocar a letra inicial no campo de busca

Então deve aparecer todos os carros de acordo com a letra escolhida

4.2- Botão de Vendido

Given eu como usuário queira marcar o carro como vendido

When apertar o botão vendido

Then deve aparecer um fundo verde na linha da tabela selecionada

4.3- Botão de Disponível

Given eu como usuário queira marcar o carro como disponível

When apertar o botão disponível

Then deve aparecer um background transparente na linha da tabela selecionada

4.4- Botão de Em Análise

Given eu como usuário queira marcar o carro em análise

When apertar o botão Em Análise

Then deve aparecer um background amarelo na linha da tabela selecionada

4.5- Botão de deletar

Given eu como usuário queira deletar o carro na tabela

When apertar o botão deletar

Then deve apagar toda a linha da tabela selecionada

4.6 – Campo de Modelo (nome do carro) Cenário 1

Given eu como usuário tenho um carro pra inserir na tabela

When insiro o nome do carro no campo modelo

Then o campo aceita qualquer tipo de caractere

4.6.1 – Campo de Modelo (nome do carro) Cenário 2

Given *que eu como usuario*

When eu adicionar o carro na tabela com o campo modelo vazio

Then deve aparecer uma mensagem de erro "Insira o nome do carro por favor" em vermelho.

4.7 – Campo de Fabricante Cenário 1

Given eu como usuario tenha um carro pra adicionar na tabela

When insiro o nome do carro no campo fabricante

Then o campo deve aceitar todos os tipos de caracter

4.7.1 – Campo de Fabricante Cenário 2

Given que eu como usuario

When eu tentar adicionar o carro na tabela com o campo fabricante vazio

Then deve aparecer uma mensagem de erro "Insira o nome do fabricante por favor" em vermelho.

4.8 – Campo de Ano Cenário 1

Given eu como usuario tenha um carro pra adicionar na tabela

When insiro o ano do carro em valor numérico no campo ano

Then o campo deve aceitar somente 4 números

4.8.1 – Campo de Ano Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo ano vazio

Then deve mostrar uma mensagem de erro dizendo "Insira o ano do carro por favor" em vermelho

4.9 – Campo de Motor Cenário 1

Given que eu como o usuario tenha um carro pra adicionar na tabela

When insiro a potencia do carro em valor numérico no campo motor

Then o campo deve aceitar somente números e caracteres especiais

4.9.1 – Campo de Motor Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo motor vazio

Then deve mostrar uma mensagem de erro dizendo "Insira a potência do carro por favor" em vermelho

4.10 – Campo de Combústivel Cenário 1

Given que eu como usuario tenha um carro pra adicionar na tabela

When insiro o tipo de combustível do carro no campo

Then o campo deve aceitar somente letras

4.10.1 – Campo de Combústivel Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo combustivel vazio

Then deve mostrar uma mensagem de erro dizendo "declare se o carro é flex, GNV, gasolina ou alcool, por favor" em vermelho

4.11 – Campo de Cambio Cenário 1

Given eu como usuario tenha um carro pra adicionar na tabela

When insiro o tipo de cambio do carro no campo

Then o campo deve aceitar somente letras

4.11.1 – Campo de Cambio Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo cambio vazio

Then deve mostrar uma mensagem de erro dizendo "Insira o tipo de cambio, manual ou automatico, por favor" em vermelho

4.12 – Campo de Kilometragem Cenário 1

Given eu como usuario tenha um carro pra adicionar na tabela

When insiro a kilometragem atual do carro

Then o input deve aceitar somente números e caracteres especiais

4.12.1 – Campo de Kilometragem Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo kilometragem vazio

Then deve mostrar uma mensagem de erro dizendo "Insira a kilometragem que o carro se encontra, por favor" em vermelho

4.13 – Campo de Valor Comprado Cenário 1

Given eu como usuario tenha um carro pra adicionar na tabela

When insiro o valor pago no carro do carro em números

Then o campo deve aceitar somente números

4.13.1 – Campo de Valor Comprado Cenário 2

Given que eu tenha um carro para adicionar na tabela

When eu tentar adicionar o carro na tabela com o campo valor comprado vazio

Then deve mostrar uma mensagem de erro dizendo "Insira o valor em reais que o carro foi comprado , por favor" em vermelho

4.14 – Campo de Valor Comprado

Given que eu adicione um carro na tabela

When preencher todos os campos obrigatorios

Then no campo valor de venda na tabela, deve aparecer uma valor de 45% a mais que o valor informado no campo de valor comprado

5. Estrategia de Teste

O plano de testes abrange todas as funcionalidades descritas acima. Esse plano de testes exclui a automatização de teste ate o momento

Testes Manuais: Todas as funcionalidades serão testadas manualmente pelo time de qualidade seguindo a documentação de Cenários de teste.

Teste de Performance: Será feito todos os teste de performance atraves da ferramenta Jmeters

Versão Beta: Será lançada uma versão beta para 20 usuários pré-cadastrados antes do release.

6. Local dos teste Alfa e Beta

- O local dos testes será definido com o cliente, será feito no escritório ou tentaremos emular o ambiente real de uso

7. Classificação de Bugs

Os Bugs serão classificados com as seguintes severidades:

ID	Nivel de Severidade	Descrição
1	Blocker	<ul style="list-style-type: none">• Bug que bloqueia o teste de uma função ou feature causa crash na aplicação.• O Botão não funciona impedindo o uso completo da funcionalidade.• Bloqueia release.
2	Critical	<p>Feature funciona pobremente</p> <ul style="list-style-type: none">• Input esperados causam crash ou efeitos indesejados• Input incomum causa efeitos irreversíveis

3	Major	<p>Feature não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada</p> <ul style="list-style-type: none"> • Mensagem de erro ou sucesso não é exibida • Input incomum causar efeitos indesejados, mas contornáveis
4	Minor	<p>Quase nenhum impacto na funcionalidade, mas ainda é um erro válido</p> <ul style="list-style-type: none"> • Erro ortográfico • Pequenos erros de UI

8.Definição de Pronto

-Será considerada pronta as funcionalidades que passarem pelas verificações e testes descritas nestes TestPlan, não apresentarem bugs com a severidade acima de Minor, e passarem por uma validação de negócio de responsabilidade do time de produto.

