
Plano de Testes do SUSEG

Versão: 1.6

Conteúdo

1. ESCOPO.....	5
1.1 ITENS DO ESCOPO	5
1.2 ITENS FORA DO ESCOPO.....	5
2. ITENS DE TESTE.....	5
2.1 REQUISITOS	5
2.2 MODELAGEM DE DADOS	6
2.3 PROTÓTIPOS	7
2.3.1 Tela inicial:.....	7
2.3.2 Dados de localização:	7
2.3.3 Dados do segurado:.....	8
2.3.4 Dados do veículo:	8
2.3.5 Dados dos condutores:	9
2.3.6 Detalhes da cotação:	9
3. PREMISSAS	10
3.1 COBERTURA DOS TESTES	10
3.2 ÍNDICE MÁXIMO DE ERROS ADMISSÍVEL	10
3.3 ÍNDICE ÓTIMO DESEJADO	10
4. ANÁLISE DE RISCOS.....	10
4.1 FALTA DE FERRAMENTAS	10
4.2 ENTREGA TARDIA DA APLICAÇÃO PARA TESTES	10
4.3 MUDANÇA DE REQUISITOS	10
4.4 CLASSIFICAÇÃO DOS DEFEITOS.....	10
5. CRONOGRAMA.....	11
6. AMBIENTE DE TESTES.....	11
7. FERRAMENTAS DE TESTE.....	11
8. CENÁRIOS E CASOS DE TESTES	11
8.1 TC001 - CRIAR NOVA COTAÇÃO.....	12
8.1.1 Cenário de teste:	12
8.1.2 Teste unitário:.....	12
8.2 TC002 - VALIDAR INFORMAÇÕES DE LOCALIZAÇÃO	12
8.2.1 Cenário de teste:	12
8.3 TC003 - INFORMAR OS DADOS DE LOCALIZAÇÃO	12
8.3.1 Cenário de teste:	12
8.3.2 Teste unitário:.....	12
8.4 TC004 - VALIDAR INFORMAÇÕES DO SEGURADO COMO PESSOA FÍSICA	13
8.5 TC005 - VALIDAR INFORMAÇÕES DO SEGURADO COMO PESSOA JURÍDICA	13
8.6 TC006 - INFORMAR OS DADOS DO SEGURADO.....	13
8.6.1 Cenário de teste:	13
8.6.2 Teste unitário:.....	14
8.7 TC007 - LISTAR FABRICANTES NA OPÇÃO PARA SELECIONAR O FABRICANTE.....	14
8.7.1 Cenário de teste:	14
8.7.2 Testes de integração:	14
8.8 TC008 - OPTAR PELO ANO DE MODELO COM BASE NO ANO DE FABRICAÇÃO SELECIONADO	14
8.8.1 Cenário de teste:	14
8.9 TC009 - LISTAR MODELOS NA OPÇÃO PARA SELECIONAR O MODELO.....	14
8.9.1 Cenário de teste:	14
8.10 TC010 - GRAVAR VEÍCULO NA COTAÇÃO.....	15
8.10.1 Cenário de teste:	15

8.10.2	Testes unitários:.....	15
8.11	TC011 - GRAVAR UM CONDUTOR NA COTAÇÃO	15
8.11.1	Cenário de teste:	15
8.11.2	Teste unitário:.....	16
8.12	TC012 - TESTAR CÁLCULO DO VALOR BASE PRÊMIO DA COTAÇÃO	16
8.12.1	Cenário de teste:	16
8.12.2	Teste de integração:.....	16
8.13	TC013 - TESTAR CÁLCULO DO VALOR PELO PERFIL DO CONDUTOR	16
8.13.1	Cenário de teste:	16
8.13.2	Teste unitário:.....	16
8.14	TC014 - TESTAR CÁLCULO DO VALOR DO PRÊMIO	16
8.14.1	Cenário de teste:	16
8.14.2	Teste unitário:.....	17
8.15	TC015 - TESTAR CÁLCULO DO VALOR DA FRANQUIA.....	17
8.15.1	Cenário de teste:	17
8.15.2	Teste de integração:.....	17
8.16	TESTE DE CARGA	17
8.17	TESTE DE PERFORMANCE.....	17
9.	FUNÇÕES E RESPONSABILIDADES.....	18
9.1	ANALISTAS DE TESTES: ELIZANDRO E RICARDO	18
9.2	DESENVOLVEDORES DE TESTES: ELIZANDRO E RICARDO.....	18
9.3	TESTADORES: ELIZANDRO E RICARDO	18
10.	GERÊNCIA DE DEFEITOS	18
11.	FORMA DE COMUNICAÇÃO	18
11.1	COMUNICAÇÃO.....	18

Histórico de Alterações

Data	Versão	Autor	Descrição
20/02/2016	1.0	Elizandro J. Schramm	Criação do documento
21/02/2016	1.1	Elizandro J. Schramm	Definição do escopo, dos itens de testes, das premissas, da análise de riscos, do cronograma e da comunicação.
22/02/2016	1.2	Ricardo Voigt	Definição do ambiente e ferramentas de testes, descrição de alguns casos de testes e definição das funções e responsabilidades.
23/02/2016	1.3	Elizandro J. Schramm	Descrição de alguns casos de testes.
26/02/2016	1.4	Ricardo Voigt	Descrição de alguns casos de testes.
29/02/2016	1.5	Elizandro J. Schramm	Adicionado testes de carga e performance.
29/02/2016	1.6	Ricardo Voigt	Adicionado descrição de gerência de defeitos e descrito alguns casos de testes. Entrega do plano de testes.

1. Escopo

Esta seção tem como objetivo apresentar o escopo de testes que serão realizados sobre a aplicação SUSEG, que foi desenvolvida para a disciplina “Trabalho Final Módulo 2” do curso de especialização de Engenharia de Software na UDESC de Ibirama. O SUSEG é uma solução WEB para a criação de cotações de seguro para veículos.

1.1 Itens do Escopo

Estão previstos como dentro do escopo os testes das funcionalidades abaixo:

- ✚ criação de uma cotação de seguro para um veículo;
- ✚ informar o veículo que terá o seguro cotado;
- ✚ adicionar condutores do veículo;
- ✚ informar o titular do seguro;
- ✚ calcular o valor do prêmio anual do seguro de um veículo.

Estão previstos também como dentro do escopo os testes não funcionais abaixo:

- ✚ sistema deve suportar a criação de mais de 50 cotações simultâneas;
- ✚ sistema deve responder ao usuário em no máximo 1 segundo por ação.

1.2 Itens fora do Escopo

As funcionalidades abaixo não serão testadas no primeiro momento, pois são de baixo risco e poderão ter seus testes postergados:

- ✚ importação dos veículos da tabela FIPE;
- ✚ incluir cláusulas específicas que não estão previamente cadastradas no sistema;
- ✚ informar classe de bônus;
- ✚ relatórios dos seguros que estão próximos da data de vencimento;
- ✚ envio de e-mail para o corretor com relatórios;
- ✚ relatório de classe de bônus por região e faixa etária;
- ✚ informar as cláusulas que vão compor o seguro;
- ✚ escolher a franquia desejada;
- ✚ informar local de pernoite do veículo;
- ✚ informar detalhes do local das diárias do veículo;
- ✚ corretor definir o seu percentual de comissão padrão;
- ✚ consultar um seguro cadastrado;
- ✚ gerar a apólice do seguro.

2. Itens de teste

Essa seção visa definir tecnicamente os itens do escopo dos testes.

2.1 Requisitos

Serão testados os Requisitos Funcionais abaixo:

RF002 - O sistema deve calcular o valor do prêmio anual do seguro de um veículo.

RF003 - O sistema deve permitir adicionar condutores do veículo com os dados: nome, sexo, idade, estado civil, se tem filho, CPF e CNH.

RF007 - O sistema deve permitir identificar o perfil do condutor.

RF008 - O sistema deve calcular o valor do seguro.

RF010 - O sistema deve gerar um código para cada cotação.

RF015 - O sistema deve permitir informar os dados do titular do seguro, que são: nome, endereço, identidade e CPF quando pessoa física, CNPJ e IE quando pessoa jurídica.

RF016 - O sistema deve permitir informar os dados do veículo, que são: placa, chassi, código renavam, ano de fabricação, ano de modelo e quilometragem média por mês.

RF020 - O sistema ao gerar a cotação deve exibir detalhes, com cláusulas e valores.

RF023 - O sistema deve permitir a consulta de um seguro pelo CPF ou CNPJ do titular do seguro.

RF026 - O sistema deve permitir informar os dados do endereço de localização do veículo.

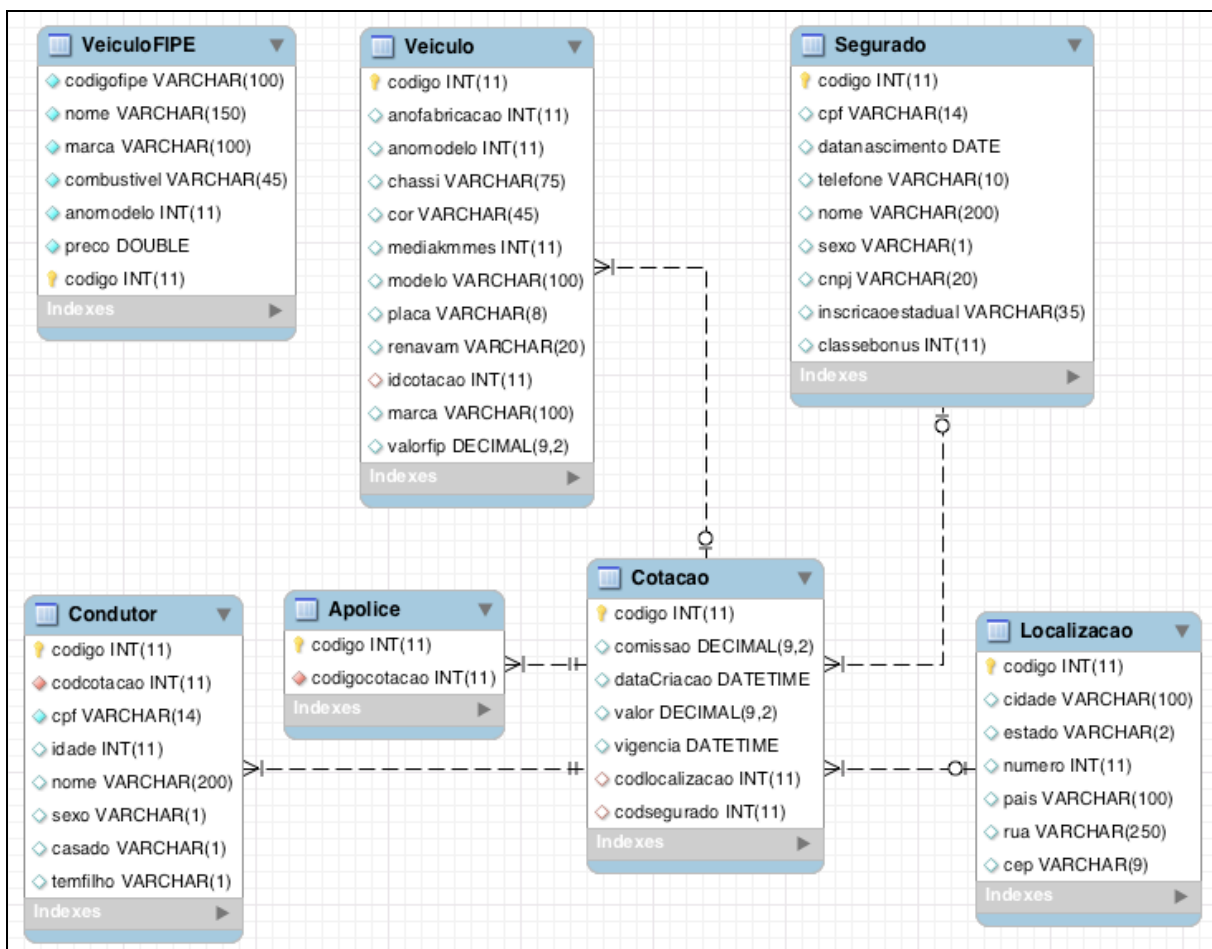
Serão testados os Requisitos Não Funcionais abaixo:

RNF001 - O sistema deve suportar a criação de mais de 50 cotações simultâneas.

RNF002 - O tempo máximo de resposta do sistema para cada ação do usuário é 1 seg.

2.2 Modelagem de dados

A modelagem de dados do SUSEG é apresentada na figura a seguir:



A tabela VeiculoFIPE não está ligada às outras tabelas por ser uma tabela de importação de veículos da tabela FIPE. Essa importação é executada semanalmente, o que pode alterar os dados de um veículo já cotado e já segurado. Quando uma nova cotação é realizada, um registro ou vários registros são adicionados na tabela Veiculo com parte dos dados provenientes da tabela VeiculoFIPE e outros informados pelo usuário ao criar a cotação.

2.3 Protótipos

Abaixo estão os protótipos, que foram criados com alta fidelidade, das principais telas do sistema. A ideia do sistema é ser executado em forma de *wizard*, onde o usuário navega entre as telas informando um conjunto de dados. A cada tela que é navegada, ocorre um requisição ao servidor WEB para gravar as informações. Ao final, o servidor processa as informações e retorna o cálculo da cotação, que é apresentada ao usuário.

2.3.1 Tela inicial:



2.3.2 Dados de localização:



2.3.3 Dados do segurado:

SUSEG SEGUROS

Gerador de cotações e apólices de seguros.

INÍCIOLOCALIZAÇÃOSEGURADOVEÍCULOCONDUTORESCOTAÇÃO

Informações sobre o segurado

Tipo do segurado:

☒ Pessoa física ☐ Pessoa jurídica

Nome

João da Silva

Data nascimento

12/10/1990

CPF (somente números)

00000000000

Telefone

(47) 0000-0000

Sexo:

☒ Masculino ☐ Feminino

Classe de bônus:

0

VOLTARPRÓXIMO

2.3.4 Dados do veículo:

SUSEG SEGUROS

Gerador de cotações e apólices de seguros.

INÍCIOLOCALIZAÇÃOSEGURADOVEÍCULOCONDUTORESCOTAÇÃO

Informe os dados do seu veículo

Fabricante

BMW

Ano de fabricação

2015

Ano de modelo

☐ 2015 ☒ 2016

Modelo

320iA GT Sport 2.0 Turbo 16V 184cv 5p

Placa:

ABC-1234

Chassi:

Renavam:

Cor:

Azul

Média KM mês:

1000

VOLTARPRÓXIMO

2.3.5 Dados dos condutores:

SUSEG SEGUROS

Gerador de cotações e apólices de seguros.

INÍCIO

LOCALIZAÇÃO

SEGURADO

VEÍCULO

CONDUTORES

COTAÇÃO

Informações sobre os condutores

#	Name	CPF	Idade	Sexo	Filhos	Casado
1	<input type="text" value="Elizandro"/>	<input type="text" value="123.123.123-20"/>	<input type="text" value="27"/>	<div><input checked="" type="radio"/> Masculino</div> <div><input type="radio"/> Feminino</div>	<div><input type="radio"/> Sim</div> <div><input checked="" type="radio"/> Não</div>	<div><input type="radio"/> Sim</div> <div><input checked="" type="radio"/> Não</div>
2	<input type="text" value="Name"/>	<input type="text" value="000.000.000-00"/>	<input type="text" value="18"/>	<div><input checked="" type="radio"/> Masculino</div> <div><input type="radio"/> Feminino</div>	<div><input checked="" type="radio"/> Sim</div> <div><input type="radio"/> Não</div>	<div><input checked="" type="radio"/> Sim</div> <div><input type="radio"/> Não</div>

ADD ROW

DELETE ROW

VOLTAR

PRÓXIMO

2.3.6 Detalhes da cotação:

SUSEG SEGUROS

Gerador de cotações e apólices de seguros.

INÍCIO

LOCALIZAÇÃO

SEGURADO

VEÍCULO

CONDUTORES

COTAÇÃO

Detalhes do orçamento

Dados pessoas

Segurado: 0

Sexo: M

Data nascimento: 08/09/1988

CPF: 0

Telefone: 0

Endereço: Rua: 0, 0, 0

Cidade: 0

Estado: 0

Dados do veículo

Veículo: A3 Sedan1.8 16V TB FSI 180cv S-tronic 4p

Ano: 2014Modelo: 2015

Placa: ABC-1234

Chassi: aaaaa

Coberturas

Valor segurado (Casco): 105%

Cobertura: Colisão, incêndio, roubo e furto

Dano Moral e Estético: R\$20.000,00

Danos Materiais (RCFV): R\$50.000,00

Danos Corporais (RCFV): R\$100.000,00

Médico-Hospitalares: R\$5.000,00

Nesta última tela, o usuário pode optar por salvar a cotação para futuras alterações, ou finalizá-la, o que a transformará em uma apólice que não poderá mais ser alterada, porém não estão sendo previstos testes automatizados para essa etapa, conforme item 1.2 Itens fora do escopo.

3. Premissas

Serão apresentadas nessa seção os critérios de sucesso e falha que conduzem o processo de testes.

3.1 Cobertura dos testes

Será realizada uma cobertura parcial dos testes, pois há funcionalidades que não serão testadas, conforme apresentado na seção 1.2 Itens fora do escopo.

3.2 Índice máximo de erros admissível

O número máximo de erros que aceitaremos será 10. Encontrando mais que 10 erros, o sistema voltará para o desenvolvimento e não será liberado para os clientes.

3.3 Índice ótimo desejado

O índice ótimo de erros encontrados nesse aplicativo é de 0 erros.

4. Análise de riscos

Essa seção apresenta os riscos do projeto e suas análises.

4.1 Falta de ferramentas

Caso não sejam disponibilizadas todas as ferramentas necessárias para a equipe de testes, um risco alto será criado no projeto para que o Gerente de Projetos tome as atitudes cabíveis para disponibilizar as ferramentas o mais rápido possível.

4.2 Entrega tardia da aplicação para testes

A partir da publicação deste plano de testes, a equipe de desenvolvimento terá 2 semanas para entregar o sistema completo à equipe de testes, caso contrário os testes não serão iniciados.

4.3 Mudança de requisitos

A equipe de testes deve ser envolvida antes da mudança de requisitos efetivamente ocorrer, pois dependerá do nosso parecer se aceitamos a mudança ou não.

4.4 Classificação dos defeitos

Os defeitos encontrados na execução dos testes serão classificados em 4 tipos:

- Crítico: erro que impede o prosseguimento das execuções dos demais testes;
- Alto: erro que impede a execução da funcionalidade testada, porém os demais testes podem prosseguir;

- ✚ Médio: erro que não impede a execução da funcionalidade, porém ocasiona erro ao prosseguir. É um potencial erro alto.
- ✚ Baixo: erro que não impede a execução da funcionalidade e não impacta nas demais funcionalidades. Trata-se de um erro específico que pode ser contornado.

5. Cronograma

Segue abaixo cronograma para a realização dos testes.

- ✚ 20/02/2016 - Criação do plano de testes
- ✚ 29/02/2016 - Entrega do plano de testes
- ✚ 06/03/2016 - Entrega dos testes unitários
- ✚ 08/03/2016 - Execução dos testes de integração
- ✚ 08/03/2016 - Execução dos testes de carga
- ✚ 12/03/2016 - Entrega do script de testes de sistema
- ✚ 14/03/2016 - Entrega do resultado dos testes
- ✚ 14/03/2016 - Encerramento dos testes

6. Ambiente de testes

Essa seção apresenta o ambiente computacional para a realização dos testes.

O ambiente para execução dos testes, requer Eclipse, com JRE 1.7 (alterar o build path para a JRE do seu computador) e MySQL instalado com uma base de dados de nome suseg_banco. Anexo ao projeto há um arquivo banco.sql que deverá ser importado sobre a base de dados. O ambiente MySQL deve estar configurado para executar no endereço localhost (local) e com usuário “root” e senha em branco. Caso seja necessário alterar as configurações de conexão ao banco de dados, deve-se alterar o arquivo DBConnection.java do pacote com.br.DAO.

É necessário também ter um servidor WEB JEE, como Glassfish ou Tomcat, efetuar o deploy da aplicação susegbackend nele e rodá-lo na porta 8080. Caso o servidor esteja configurado para executar em outra porta, então é necessário alterar todas as ocorrências de “8080” do arquivo wizard.js da pasta “WebContent/assets/js” do projeto.

7. Ferramentas de teste

As ferramentas utilizadas para os testes são:

- ✚ JUnit, para a execução dos testes unitários e validação dos casos de testes descritos abaixo; *Executamos o JUnit diretamente pela IDE Eclipse.*
- ✚ JMeter, para a execução dos testes de carga;
- ✚ PhantomJS, para a execução dos scripts para os testes de sistema.

8. Cenários e casos de testes

Serão apresentados os cenários para a criação e execução dos testes dentro dos casos de testes.

8.1 TC001 - Criar nova cotação

8.1.1 Cenário de teste:

1. Acessar o sistema através da URL <http://localhost:8080/susegbackend>.
2. Sistema apresenta a tela inicial conforme protótipo da seção 2.3.1.
3. Clicar no botão “Novo”.
4. Sistema insere uma cotação no banco (select * from Cotacao).
5. Sistema apresenta a tela de localização conforme protótipo da seção 2.3.2.
 - a. Este cenário deve ser automatizado por um script chamado TC001.js que deve ser executado pelo phantomjs (comando: ./phantomjs TC001.js).

8.1.2 Teste unitário:

1. Executar o teste testInserirNovaCotacao() da classe CotacaoTest.java.

8.2 TC002 - Validar informações de localização

8.2.1 Cenário de teste:

1. Executar o TC001.
2. Na tela de Localização, não informar valor para o campo “Rua”.
3. Clicar no botão “Próximo”.
4. Sistema não muda a tela e exibe a mensagem “Você deve informar a Rua” em vermelho abaixo do campo “Rua”.
5. Efetuar os passos anteriores com os demais campos da tela.
6. Sistema não deve permitir ir para a próxima tela até que todos os campos estejam informados.
7. Para todos os campos, o sistema deve alertar que o usuário precisa informá-los.

8.3 TC003 - Informar os dados de localização

8.3.1 Cenário de teste:

1. Executar o TC001.
2. Na tela de Localização, informar todos os dados.
3. Clicar no botão “Próximo”.
4. Sistema insere uma localização no banco (select * from Localizacao).
5. Sistema apresenta a tela para informar o seguro conforme protótipo da seção 2.3.3.
 - a. Este cenário deve ser automatizado por um script chamado TC003.js que deve ser executado pelo phantomjs (comando: ./phantomjs TC003.js).

8.3.2 Teste unitário:

1. Executar o teste testInserirNovaLocalizacao() da classe LocalizacaoTest.java.

8.4 TC004 - Validar informações do segurado como pessoa física

1. Executar o TC001.
2. Executar o TC003.
3. Selecionar a opção “Pessoa física” para o tipo do segurado.
4. Sistema exibe os campos para que o usuário preencha: nome, data nascimento, CPF, telefone, sexo e classe bônus.
5. Usuário informa os dados, porém deixa o campo nome em branco.
6. Usuário clica no botão próximo.
7. Sistema não muda a tela e exibe a mensagem “Você deve informar o Nome do segurado” em vermelho abaixo do campo “Nome”.
8. Efetuar os passos anteriores com os demais campos da tela.
9. Sistema não deve permitir ir para a próxima tela até que todos os campos estejam informados.
10. Para todos os campos, o sistema deve alertar que o usuário precisa informá-los.

8.5 TC005 - Validar informações do segurado como pessoa jurídica

1. Executar o TC001.
2. Executar o TC003.
3. Selecionar a opção “Pessoa jurídica” para o tipo do segurado.
4. Sistema exibe os campos para que o usuário preencha: razão social, IE, CNPJ, telefone e classe bônus.
5. Usuário informa os dados, porém deixa o campo razão social em branco.
6. Usuário clica no botão próximo.
7. Sistema não muda a tela e exibe a mensagem “Você deve informar o Nome da empresa” em vermelho abaixo do campo “Razão social”.
8. Efetuar os passos anteriores com os demais campos da tela.
9. Sistema não deve permitir ir para a próxima tela até que todos os campos estejam informados.
10. Para todos os campos, o sistema deve alertar que o usuário precisa informá-los.

8.6 TC006 - Informar os dados do segurado

8.6.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Na tela de Segurado, informar todos os dados do segurado.
4. Clicar no botão “Próximo”.
5. Sistema insere o segurado no banco.

6. Sistema apresenta a tela para informar o veículo conforme protótipo da seção 2.34.

8.6.2 Teste unitário:

1. Executar o teste `testInserirNovoSegurado()` da classe `SeguradoTest.java`.

8.7 TC007 - Listar fabricantes na opção para selecionar o fabricante

8.7.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC006.
4. Na tela de veículo, o sistema apresenta o campo Fabricante em um *combobox* com as marcas cadastradas na base de dados.
5. Usuário clica sobre o *combobox* Fabricante e então as marcas são disponibilizadas para ele escolha a desejada.

8.7.2 Testes de integração:

1. Executar o teste `testGetFabricantesFIPE()` da classe `TestesIntegracaoFIPE.java` para garantir que um JSON seja retornado com todas marcas, pois só assim o *combobox* poderá ser preenchido para o usuário.
2. Se possível, implementar um teste para retornar os modelos de acordo com uma marca simulando em código fonte uma marca selecionada pelo usuário (`testGetModelosPorFabricanteFIPE()` na classe `TestesIntegracaoFIPE.java`).

8.8 TC008 - Optar pelo ano de modelo com base no ano de fabricação selecionado

8.8.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC006.
4. Na tela de Veículo selecionar no campo fabricante o item BMW.
5. No campo ano de fabricação selecionar 2015.
6. O sistema deverá exibir para seleção de ano modelo, o ano de fabricação selecionado e o ano de fabricação selecionado mais um, que neste cenário será 2015 e 2016 nesta ordem.

8.9 TC009 - Listar modelos na opção para selecionar o modelo

8.9.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.

3. Executar o TC006.
4. Na tela de Veículo selecionar no campo fabricante o item BMW.
5. No campo ano de fabricação, selecionar 2015.
6. No campo de ano de modelo, selecionar 2016.
7. No combo de modelo, o sistema exibirá somente os modelos do fabricante com o ano de fabricação e ano de modelo informados.

8.10 TC010 - Gravar veículo na cotação

8.10.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC006.
4. Na tela de Veículo informar todos os dados.
5. Clicar no botão "Próximo".
6. Sistema insere um veículo no banco de dados.
7. Sistema apresenta a tela para informar o condutor conforme protótipo da seção 2.3.5.

8.10.2 Testes unitários:

1. Executar o teste `testInserirVeiculoNaCotacao()` da classe `TestesIntegracaoCotacao.java`.
2. Executar o teste `testInserirListaVeiculosNaCotacao()` da classe `TestesIntegracaoCotacao.java`.

8.11 TC011 - Gravar um condutor na cotação

8.11.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC006.
4. Executar o TC010.
5. Na tela de Condutor no campo Nome informar um nome masculino.
6. No campo CPF informar um CPF válido, exemplo "457.698.492-66".
7. No campo idade informar "27".
8. No campo sexo, selecionar a opção "masculino".
9. No campo filhos, selecionar a opção "não".
10. No campo casado, selecionar a opção "não".
11. Clicar no botão "Próximo".
12. Sistema insere um condutor no banco de dados.

13. Sistema apresenta tela com o resultado da cotação conforme protótipo da seção 2.3.6.

8.11.2 Teste unitário:

1. Executar o teste `testInserirCondutorNaCotacao()` da classe `TestesIntegracaoCotacao.java`.

8.12 TC012 - Testar cálculo do valor base prêmio da cotação

8.12.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC006.
4. Na tela de Veículo selecionar no campo fabricante o item BMW.
5. No campo ano de fabricação selecionar 2015.
6. No *radio button* de Ano de modelo selecionar 2016.
7. No campo de modelo informar 320iA GT Sport 2.0 Turbo 16V 184cv 5p.
8. Informar os dados de Placa, chassi, RENAVAM, cor e média km mês nos demais campos.
9. Clicar no botão "Próximo".
10. Sistema insere um veículo no banco de dados.
11. Sistema apresenta a tela para informar o condutor conforme protótipo da seção 2.3.5.
12. Executar o TC011.

8.12.2 Teste de integração:

1. Executar a suíte de testes `TestSuite.java` para testar a base do valor do prêmio (`testCalculoValorBasePremioVeiculoValor20Mil()` da classe `TestesIntegracaoCotacao.java`).

8.13 TC013 - Testar cálculo do valor pelo perfil do condutor

8.13.1 Cenário de teste:

1. Executar o TC012.

8.13.2 Teste unitário:

1. Executar o teste `testCalculoPercentualPerfilCondutorHomemMaior25Anos()` da classe `TestesIntegracaoCotacao.java`.

8.14 TC014 - Testar cálculo do valor do prêmio

8.14.1 Cenário de teste:

1. Executar o TC012.

8.14.2 Teste unitário:

1. Executar o teste `testCalculoValorPremioCondutorHomemMaior25Anos ()` da classe `TestesIntegracaoCotacao.java`.

8.15 TC015 - Testar cálculo do valor da franquia

8.15.1 Cenário de teste:

1. Executar o TC001.
2. Executar o TC003.
3. Executar o TC005.
4. Executar o TC010.
5. Executar o TC011.
6. O sistema apresenta a tela com o valor da franquia (10% do valor do veículo selecionado) conforme o protótipo da seção 2.3.6.

8.15.2 Teste de integração:

1. Executar a suíte de testes `TestesIntegracaoCalculos.java` para garantir que o cálculo da franquia está sendo realizado corretamente (`testCalculoValorFranquia()` da classe `CotacaoControleTest.java`).

8.16 Teste de carga

Deve ser implementado um teste de carga para garantir que o sistema suporte a criação de 50 cotações ao mesmo tempo, simulando a ação de 50 usuários simultâneos.

Para isso, o teste de carga deve fazer insert na tabela `Cotacao` por pelo menos 10 vezes com 50 usuários simultâneos.

Comando: `INSERT INTO COTACAO(DATACRIACAO, VALOR, VIGENCIA) VALUES(NOW(), 99999.99, NOW());`

8.17 Teste de performance

Deve ser implementado um teste de performance para garantir que o sistema responda às ações do usuário em até 1s.

Para isso, podem ser realizadas chamadas de web service para algum servlet do sistema, como por exemplo para o `RetornaDadosFIPE`, que é chamado quando o usuário entra na tela para informar um veículo.

Este servlet deve responder da seguinte forma:

- quando chamado assim “`RetornaDadosFIPE?acao=marcas`” ele retorna todas as marcas presentes na base de dados.

- quando chamado assim “`RetornaDadosFIPE?acao=veiculos&marca=Audi&anomodelo=2013`” ele retorna todos os modelos da marca Audi do ano de 2013.

9. Funções e responsabilidades

9.1 Analistas de testes: Elizandro e Ricardo

- ✚ Definir os Casos de Testes e criar o plano de testes.
- ✚ Analisar os resultados reportados sobre a execução dos testes.
- ✚ Reportar os erros encontrados e formular os relatórios previstos nesse plano de testes.

9.2 Desenvolvedores de testes: Elizandro e Ricardo

- ✚ Implementar os testes unitários descritos nos Casos de Testes.
- ✚ Criar a configuração do JMeter para executar os testes de carga.
- ✚ Automatizar os cenários de testes descritos nos Casos de Testes.

9.3 Testadores: Elizandro e Ricardo

- ✚ Executar os testes unitários e gravar os resultados.
- ✚ Executar os testes pelo JMeter e gravar os resultados.
- ✚ Executar os testes de script pelo phantomjs e gravar os resultados.

10. Gerência de defeitos

A gerência de defeitos do projeto SUSEG será baseado em métricas para cada defeito identificado. Com base nas métricas coletadas, teremos informações precisas sobre a origem do problema, resultando em uma melhoria contínua e eficaz do processo.

Os defeitos podem ser classificados em:

1. Erro: implementação incorreta de um algoritmo;
2. Bug: falha no sistema, como omissão de informações e cálculos incorretos;
3. Falha: ação inesperada no software.

As métricas utilizadas serão a contagem simples da quantidade de defeitos por classificação e por severidade dos defeitos encontrados.

11. Forma de comunicação

Esta seção define a comunicação da equipe de testes sobre o resultado dos testes.

11.1 Comunicação

A comunicação ocorrerá de forma única ao final da execução de todos os testes. Será criado um relatório chamado “Resultado de testes do SUSEG.pdf” que contemplará:

- ✚ Testes realizados através do JMeter serão apresentados com o gráfico “*Graph Results*” e com o relatório “*Summary Report*”.
- ✚ Testes unitários executados através do JUnit serão apresentados com um *screen shot* da execução do JUnit pelo Eclipse.
- ✚ Serão apresentados os índices com a quantidade de erros detectados.
- ✚ Será apresentado o parecer da equipe de testes se aceita ou não a liberação do sistema para os clientes.