



Estratégia de testes

Proposta

A empresa **XBookMarket** lançou um novo serviço de empréstimos e vendas online de livros digitais e deseja gerar **Planos de Testes** completos que contemplem todas as etapas do processo de desenvolvimento de software. Os Clientes devem realizar seu cadastro através da aplicação e apenas assim estarão habilitados para realizarem e confirmarem suas solicitações de empréstimos e /ou venda de obras literárias em formato digital. O desafio consiste em construir em Java a parte operacional da aplicação e uma proposta de camada de integração (recomenda-se a implementação de web services) que deverá ser usada como **objeto de teste**. Após isso, a equipe deverá planejar e documentar os testes automatizados, demonstrando a estabilidade, segurança, performance e confiabilidade dos principais processos do sistema, principalmente as operações envolvendo **CADASTROS**.

Requisitos:

- 1) O cliente deve possuir: Nome, E-mail, Telefone, CPF, RG e Endereço.
- 2) A IDE usada deverá ser o Eclipse.
- 3) Deverão ser feitos testes automatizados usando, no mínimo, as ferramentas Jmeter, SoapUI, Selenium e Cucumber.
- 4) Deverá ser usado o Git (para versionamento do código da aplicação) e Github para o repositório da equipe.
- 5) Métodos Ágeis (e suas respectivas ferramentas de suporte) deverão ser utilizados para organizar e estruturar as tarefas que serão desenvolvidas pela equipe.



Objetivos

Testar e validar as funcionalidades do sistema de gestão de compras e empréstimos da empresa **XBookMarket** utilizando os seguintes tipos de teste:

- [Teste de Estresse](#)
- [Teste de Integração](#)
- [Teste de Interface](#)

Tecnologias utilizadas

- Java - Java é uma linguagem multiplataforma, orientada a objetos.
- JMeter - Fazer testes de stress e carga para garantir o desempenho da aplicação.
- SOAPUI - Ferramenta open source escrita em Java cuja função principal é consumir e testar Web Services.
- Cucumber - Utilizar BDD baseado na linguagem Gherkin a fim de descrever em nível “menos técnico” os casos de uso a serem efetuados.
- Selenium - Ferramenta utilizada para automação de testes de usabilidade em ambiente web.
- Maven - Automatizar a execução de testes unitários durante a fase de build, entre outras automatizações.
- Eclipse - IDE para desenvolvimento Java e outras linguagens.



Relatório de testes

Teste de Stress

Ferramenta	Estratégia	Dados de Entrada	Dados de Saída
Jmeter	Será realizada a inserção de vários registros diferentes no banco com diferentes quantidades de usuários virtuais para encontrar o limite de atuação para as atividades.	INSERT INTO `Pessoa` (`cpf`,`nome`,`nacionalidade`,`idade`,`sexo`,`email`,`telefone`,`rg`,`endereco`) VALUE ("868993-8","Thaddeus Dawson", "Spain",40,"M","turpis.nec@googlee.com","65434555","169125449","P.O. Box 614, 3953 Morbi Street"),	java.sql.SQLException : Cannot create PoolableConnectionFactory(Data source rejected establishment of connection, message from server: "Too many connections") ----- Mensagem de resposta:java.sql.SQLException: Error preloading the connection pool

Foi identificado que a quantidade de registros em uma mesma requisição de inserções não afeta o desempenho do banco, mas sim o número de acessos concorrentes ao banco. Foram realizados testes de estresse e identificado que até 150 acessos simultâneos não ocorreram falhas ou ocorreram pequenas falhas, menos de 1%, porém, acima deste número, o banco começa a apresentar erros referentes à conexão conforme print's demonstrados abaixo.

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Min.	Máx.	% de Erro	Vazão	KB/s	Sent KB/sec
INSERT Pessoa - ...	150	2338	2393	3067	3212	3617	567	3707	0,00%	38,5/sec	0,45	0,00
TOTAL	150	2338	2393	3067	3212	3617	567	3707	0,00%	38,5/sec	0,45	0,00

Em 151* acessos ao sistema apresenta taxa de erro 1,99%. Acima disso as taxas de erro vão aumentando gradativamente de acordo com o número de utilização das portas.

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Min.	Máx.	% de Erro	Vazão	KB/s	Sent KB/sec
INSERT Pessoa - ...	151	2207	2094	3076	3185	3303	344	3475	2,65%	41,8/sec	0,52	0,00
TOTAL	151	2207	2094	3076	3185	3303	344	3475	2,65%	41,8/sec	0,52	0,00

Mensagem de resposta:java.sql.SQLException: Cannot create PoolableConnectionFactory (Data source rejected establishment of connection, message from server: "Too many connections")

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line	99% Line	Min.	Máx.	% de Erro	Vazão	KB/s	Sent KB/sec
INSERT Pessoa - ...	600	1387	551	4198	5575	6852	2	7683	29,83%	72,4/sec	2,98	0,00
TOTAL	600	1387	551	4198	5575	6852	2	7683	29,83%	72,4/sec	2,98	0,00



Teste de Integração

Ferramenta	Estratégia	Dados de Entrada	Dados de Saída
SOAP UI	Será verificado se a requisição para busca de endereço utilizando CEP válido está retornando o resultado esperado e se ao informar um CEP inválido a mensagem de erro é apresentada.	<pre><cep>72592105</cep> > ----- <tem:nCdServico>04030</tem:nCdServico></pre>	<pre><bairro>Residencial Santos Dumont (Santa Maria)</bairro> <cep>72592105</cep> > <cidade>Brasilia</cidade> <complemento2/> <end>QRC 5</end> <uf>DF</uf> <DataMaxEntrega>02/09/2022</DataMaxEntrega></pre>

Foi realizado teste em dois end points, um de consultar um endereço utilizando o CEP e o segundo de consulta de prazo para a devolução dos livros emprestados. O tempo de resposta média foi de 130ms para o cálculo do prazo e de 1096ms, um valor considerável, que poderia impactar na qualidade da aplicação.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:consultaCEPResponse xmlns:ns2="http://cliente.bean.master.sigep.bsb.corre">
      <return>
        <bairro>Residencial Santos Dumont (Santa Maria)</bairro>
        <cep>72592105</cep>
        <cidade>Brasilia</cidade>
        <complemento2/>
        <end>QRC 5</end>
        <uf>DF</uf>
      </return>
    </ns2:consultaCEPResponse>
  </soap:Body>
</soap:Envelope>
```

- SOAP Response - VALID
- Validacao_Cidade - VALID
- Validacao_end - VALID
- Validacao_UF - VALID
- Validacao_Bairro - VALID

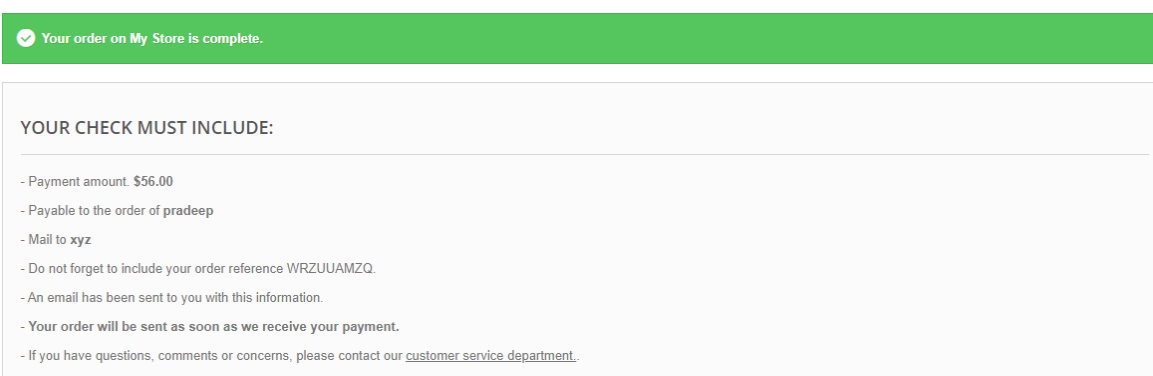
NFO:Got response for [AtendeClienteServiceSoapBinding.consultaCEP:CT001 - Consulta_Endereço] in 1096ms (400 bytes)



Teste de Interface

Ferramenta	Estratégia	Dados de Entrada	Dados de Saída
Eclipse, Selenium, Maven e Cucumber	Será realizado automação do processo de compra no site e uma automação buscando um caso de falha quando não aceito os termos de serviço.	http://automationpractice.com/	https://github.com/EricMarques/desafioGama/tree/master/livraria-testes-interface

No primeiro cenário, foi realizado o caminho feliz da compra de um livro e não foram encontradas falhas no fluxo e nem bugs na aplicação.



No segundo cenário, foi testado se o sistema apresenta a mensagem de erro caso não aceitem os termos e condições, onde também não foram encontradas falhas e nem bugs.



Referências

Ferramentas:

<https://jmeter.apache.org/>

<https://www.soapui.org/>

<https://cucumber.io/>

<https://maven.apache.org/>

<https://www.selenium.dev/>

Links úteis:

<http://tree.taiga.io/>

<http://automationpractice.com/>

<https://generatedata.com>

Repositório:

<https://github.com/EricMarques/desafioGama>