

1) Descreva abaixo com suas palavras a definição de teste de caixa branca e teste de caixa preta:

Caixa Branca: Avalia o comportamento interno do componente de software, tem acesso ao código fonte do sistema.

Caixa Preta: Testa as entradas e saídas do sistema, não tem acesso ao código.

2) Descreva abaixo com suas palavras a diferença entre teste de stress e carga:

O teste de stress valida o comportamento do sistema em condições extremas, enquanto o teste de carga pretende descobrir qual a capacidade de uso, processamento, acessos etc...

3) Descreva abaixo com suas palavras a definição de teste de integração. Cite ao menos dois exemplos:

Valida a comunicação entre os componentes de um sistema, envio e recebimento de informações.

Exemplo:

Tenho um componente responsável pela geração de um boleto para pagar e um outro componente para realização de cálculo de juros. Para gerar o boleto ele precisa chamar o componente que calcula os juros, para retornar qual o valor de juros que deve ser acionado para esse boleto que vai ser gerado.

4) O que é e qual a diferença entre BDD e TDD?

BDD E TDD são técnicas de desenvolvimento que priorizam os testes de código, integração contínua e desenvolvimento Ágil.

BDD você escreve como deve se comportar seu problema, BDD serve para criar testes e integrar regras de negócios com a linguagem de programação focando no comportamento do software.

TDD você escreve os testes e a válida de forma que eles funcionem.

5) O que é Kanban?

É uma metodologia pautada na utilização de cartões que indicam o progresso de um projeto

6) O que é Product Backlog?

O Product Backlog é uma lista contendo todas as funcionalidades desejadas para um produto.

7) O que é Daily Meeting?

O Daily Meeting acontece diariamente durante o Sprint, sendo uma reunião rápida (por volta de 15 minutos), mas que consiga promover o alinhamento entre o time e dar a visão de como está o progresso dos itens do Sprint Backlog.

8) Considerando a história abaixo, defina os possíveis tipos de testes a serem realizados.

Os tipos de teste que poderia ser utilizado seriam "unitário", "integração", "aceite".

9) Considerando a história anterior, descreva os possíveis cenários de testes utilizando a técnica BDD:

Cenário 1: Validar Campos

Dado que sou um cliente

E tenho acesso a tela de cadastro

Então valido os campos Cidade, Bairro, Endereço e N°

Cenário 2: Consultar CEP válido

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar um CEP válido

Então o sistema deve consultar as informações do endereço utilizando o serviço externo

E preencher os campos Cidade, Endereço e N° automaticamente

Cenário 3: Campo CEP em branco

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando não informo nenhum CEP

Então o sistema deve apresentar a mensagem "Ocorreu um problema ao consultar informações do endereço"

Cenário 4: Consultar CEP com caracteres

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar o campo CEP com caracteres

Então o sistema deve apresentar a mensagem "Ocorreu um problema ao consultar informações do endereço"

Cenário 5: Consultar CEP com o serviço indisponível

Dado que sou um cliente

E tenho acesso a tela de cadastro

E o serviço de consultar CEP não está disponível

Quando eu informar o campo CEP válido

Então o sistema deve apresentar a mensagem "Ocorreu um problema ao consultar informações do endereço"

Cenário 6: Consultar CEP com menos de 8 dígitos

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar o campo CEP com menos de 8 dígitos

Então o sistema deve apresentar a mensagem "Ocorreu um problema ao consultar informações do endereço"

Cenário 7: Consultar CEP com um número não existente na base de serviço do correio

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar o campo CEP com 8 dígitos

E não existente na base de serviço externo dos correios

Então o sistema deve apresentar a mensagem "Ocorreu um problema ao consultar informações do endereço"

Cenário 8: Consultar CEP único

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar um CEP único

Então o sistema deve consultar as informações do endereço utilizando o serviço externo

E preencher os campos Cidade, Endereço e N° automaticamente

Cenário 9: Consultar CEP associado a um endereço sem número (S/N)

Dado que sou um cliente

E tenho acesso a tela de cadastro

Quando eu informar um CEP que não tenha número cadastrado (S/N)

Então o sistema deve consultar as informações do endereço utilizando o serviço externo

E preencher os campos Cidade, Endereço e N° (S/N) automaticamente

10) Considerando o sistema de controle de versão GIT, descreva os comandos abaixo:

a) Qual o comando utilizado para obter (clonar) um repositório?

Git clone

b) Qual o comando utilizado para puxar todas as atualizações disponíveis de uma branch?

Git pull origin nome_da_branch

c) Qual o comando utilizado para trabalhar em uma branch específica?

Git checkout <NOME DA BRANCH>

d) Qual o comando utilizado para adicionar as atualizações ocorridas dentro de uma branch?

Git add

e) Qual o comando utilizado para realizar um commit?

Git commit

f) Qual o comando utilizado para realizar o envio do commit à um repositório remoto?

Git push origin nome_da_branch

11) Descreva a diferença entre os métodos Get, Post, Put e Delete em uma requisição?

O método GET é para busca de informações, POST para inclusão de informações, PUT para atualização e DELETE para exclusão de um registro.

12)

A) Consulte todos os produtos com valores acima de R\$7,00:

```
SELECT * FROM
PRODUCT
WHERE 1=1
AND VALUE > 7;
```

B) Consulte todos os produtos criados entre (10/05/2019 - 18/06/2019):

```
SELECT * FROM
PRODUCT
WHERE 1=1
AND CREATED BETWEEN '10/05/2019' AND '18/06/2019';
```

C) Consulte todos os clientes que iniciam com o nome “João”:

```
SELECT * FROM
CUSTOMER
WHERE
1=1
AND NAME LIKE '%João%'
```

D) Consulte o Nome do Cliente, Nome do Produto, Quantidade comprada e Data da Compra para vendas realizadas no mês de abril:

```
SELECT
C.NAME AS CLIENTE,
P.NAME AS PRODUTO,
S.QUANTITY AS QUANTIDADE,
S.DIA AS DIAS
FROM
SALE S
INNER JOIN CUSTOMER C ON 1=1
AND C.ID_CUSTOMER = S.ID_CUSTOMER
INNER JOIN PRODUCT P ON 1=1
AND P.ID_PRODUCT = S.ID_PRODUCT
WHERE
1=1
AND S.DIA BETWEEN '01/04/2019' and '30/04/2019'
```

E) Atualize o valor do produto “Iphone X” para R\$3.500:

```
UPDATE
PRODUCT
SET
VALUE = 3500
WHERE
1=1
NAME = 'Iphone X'
```

F) Realize a exclusão do cliente “João Teste”:

```
DELETE FROM
CUSTOMER
WHERE
1=1
AND NAME = 'João Teste';
```

13) Qual a definição de Abstração, Encapsulamento, Herança e Polimorfismo em orientação à objeto?

Abstração os objetos armazenam dados e possuem interface definida que determina o seu comportamento.

Encapsulamento uma forma de esconder detalhes de funcionamento do programa e também uma maneira de restringir o acesso a certos componentes.

Herança permite que uma classe possa herdar o comportamento de uma outra classe.

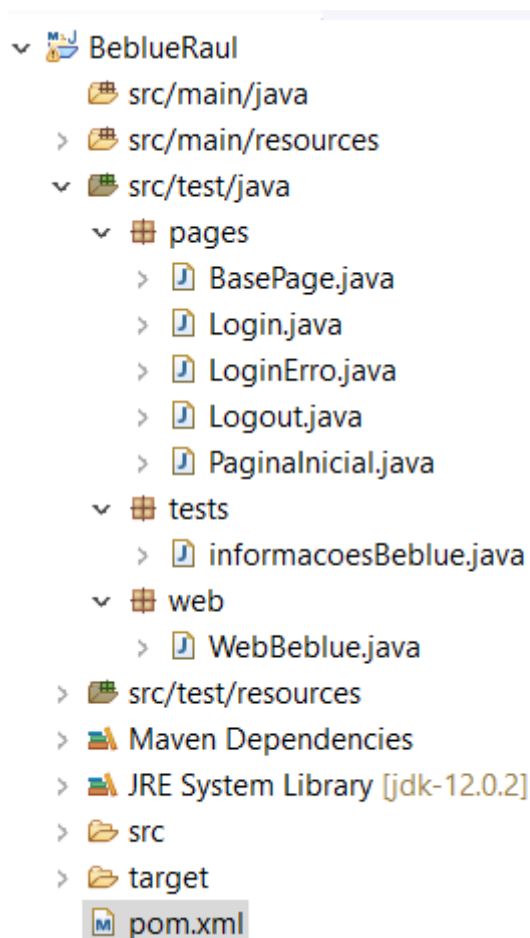
Polimorfismo permite que referências de tipos de classes mais abstratas representem o comportamento das classes concretas que referenciam.

14) O que é Page Object?

O Page Object é um padrão que funciona como interface de acesso a elementos da camada de visão – para aplicações web ele representa uma página HTML. Ele é aplicado para abstrair as páginas de uma aplicação com o objetivo de reduzir o acoplamento entre os casos de teste e a aplicação a ser testada.

15) Utilizando o site "<http://the-internet.herokuapp.com>", automatize à funcionalidade "Form Authentication" com o framework e linguagem no qual possui domínio..

Diferenciais: - Utilização do padrão Page Object - Código limpo



```

1<project xmlns="http://maven.apache.org/POM/4.0.0"
2  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4  <modelVersion>4.0.0</modelVersion>
5  <groupId>br.ce.wcaquino</groupId>
6  <artifactId>CursoCucumber</artifactId>
7  <version>0.0.1-SNAPSHOT</version>
8
9  <dependencies>
10   <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-java -->
11   <dependency>
12     <groupId>info.cukes</groupId>
13     <artifactId>cucumber-java</artifactId>
14     <version>1.2.5</version>
15   </dependency>
16
17   <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-junit -->
18   <dependency>
19     <groupId>info.cukes</groupId>
20     <artifactId>cucumber-junit</artifactId>
21     <version>1.2.5</version>
22     <scope>test</scope>
23   </dependency>
24
25   <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
26   <dependency>
27     <groupId>org.seleniumhq.selenium</groupId>
28     <artifactId>selenium-java</artifactId>
29     <version>3.11.0</version>
30   </dependency>
31
32   <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
33   <dependency>
34     <groupId>commons-io</groupId>
35     <artifactId>commons-io</artifactId>
36     <version>2.6</version>
37   </dependency>
38
39   <!-- https://mvnrepository.com/artifact/junit/junit -->
40   <dependency>
41     <groupId>junit</groupId>
42     <artifactId>junit</artifactId>
43     <version>4.12</version>
44     <scope>test</scope>
45   </dependency>
46
47
48   <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
49   <dependency>
50     <groupId>org.seleniumhq.selenium</groupId>
51     <artifactId>selenium-java</artifactId>
52     <version>3.6.0</version>
53
54   </dependency>
55
56   <!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
57   <dependency>
58     <groupId>commons-io</groupId>
59     <artifactId>commons-io</artifactId>
60     <version>2.5</version>
61   </dependency>
62
63   <!-- https://mvnrepository.com/artifact/org.easymock/easymock -->
64   <dependency>
65     <groupId>org.easymock</groupId>
66     <artifactId>easytest-core</artifactId>
67     <version>1.4.0</version>
68   </dependency>
69
70   <!-- https://mvnrepository.com/artifact/org.easymock/easymock -->
71   <dependency>
72     <groupId>org.easymock</groupId>
73     <artifactId>easytest-core</artifactId>
74     <version>1.4.0</version>
75   </dependency>
76
77 </dependencies>
78
79
80 </project>

```

```
informacoesBeblue.java LoginErro.java Logout.java PaginaInicial.java Login.java BeblueRaul/pom.xml WebBeblue.java
1 package web;
2
3 import java.util.concurrent.TimeUnit;
4
5 public class WebBeblue {
6
7     public static WebDriver createChorme(){
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\raul\\drivers\\chromedriver.exe");
9         WebDriver navegador = new ChromeDriver();
10        navegador.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
11        navegador.get("http://the-internet.herokuapp.com/");
12        return navegador;
13    }
14 }
15
informacoesBeblue.java
1 package tests;
2
3
4
5 import org.junit.Before;
6
7 public class informacoesBeblue {
8
9     private WebDriver navegador;
10
11     @Before
12     public void antesExecutar() {
13         navegador = WebBeblue.createChorme();
14     }
15
16     @Test
17     public void principal() {
18
19         // Primeiro realiza o login com usuario certo, e gera a confirmação de OK.
20         // E finaliza com um login invalido com a mensagem NOK.
21         new
22         PaginaInicial(navegador).telaInicial().realizarlogin()
23         .validacaoMensagemLogin().voltar().validacaoMensagemVoltaLogin().realizarloginErro().validacaoMensagemErro();
24     }
25
26     @After
27     public void fechar(){
28         navegador.quit();
29     }
30 }
31
32
PaginaInicial.java
1 package pages;
2
3 import org.junit.Test;
4
5 public class PaginaInicial extends BasePage {
6
7     public PaginaInicial(WebDriver navegador) {
8         super(navegador);
9     }
10
11     @Test
12     public void Login telaInicial() {
13
14         navegador.findElement(By.xpath("//div[@id=\"content\"]//a[@href=\"/login\"]")).click();
15         //navegador.findElement(By.linkText("Form Authentication")).click();
16
17         return new Login(navegador);
18     }
19 }
20
21
22
```

```
PaginalInicial.java Logout.java
1 package pages;
2
3 import static org.junit.Assert.assertEquals;
4
5 public class Logout extends BasePage {
6
7     public Logout(WebDriver navegador) {
8         super(navegador);
9     }
10
11     @Test
12     public Logout validacaoMensagemLogin() {
13
14         String excluir = navegador.findElement(By.xpath("//div[@id='flash']")).getText();
15         assertEquals("You logged into a secure area!\n", excluir);
16         return this;
17     }
18
19     @Test
20     public LoginErro validacaoMensagemVoltaLogin() {
21
22         String excluir = navegador.findElement(By.xpath("//div[@id='flash']")).getText();
23         assertEquals("You logged out of the secure area!\n", excluir);
24         return new LoginErro(navegador);
25     }
26
27     @Test
28     public Logout voltar() {
29
30         navegador.findElement(By.xpath("//div[@id='content']/a[@href='/logout']")).click();
31         return new Logout(navegador);
32     }
33 }
34
35
```

```
PaginalInicial.java Logout.java LoginErro.java
1 package pages;
2
3 import static org.junit.Assert.assertEquals;
4
5 public class LoginErro extends BasePage{
6
7     public LoginErro(WebDriver navegador) {
8         super(navegador);
9     }
10
11     @Test public LoginErro realizarloginErro() {
12         navegador.findElement(By.id("username")).sendKeys("TESTE");
13         navegador.findElement(By.id("password")).sendKeys("TESTE");
14         navegador.findElement(By.xpath("//button[@class='radius']")).click();
15         return this;
16     }
17
18     @Test
19     public LoginErro validacaoMensagemErro() {
20
21         String excluir = navegador.findElement(By.xpath("//div[@id='flash']")).getText();
22         assertEquals("Your username is invalid!\n", excluir);
23         return this;
24     }
25 }
26
27
```



```
PaginalInicial.java Logout.java LoginErro.java Login.java
1 package pages;
2
3
4 import org.junit.Test;
5
6
7 public class Login extends BasePage {
8
9
10 public Login(WebDriver navegador) {
11     super(navegador);
12 }
13
14 @Test public Logout realizarlogin() {
15     navegador.findElement(By.id("username")).sendKeys("tomsmith");
16     navegador.findElement(By.id("password")).sendKeys("SuperSecretPassword!");
17     navegador.findElement(By.xpath("//button[@class='radius']")).click();
18     return new Logout(navegador);
19 }
20
21
22
23
24 }
25
```

```
PaginalInicial.java Logout.java LoginErro.java Login.java BasePage.java
1 package pages;
2
3
4 import org.openqa.selenium.WebDriver;
5
6
7 public class BasePage {
8     protected WebDriver navegador;
9
10 public BasePage(WebDriver navegador){
11     this.navegador = navegador;
12 }
13
14 }
15
```