05

Testando validações

Transcrição

Agora que conseguimos testar o cadastro de um leilão, testaremos as validações da nossa regra de negócios. Se abrirmos a aplicação no navegador e tentarmos cadastrar um leilão por meio do formulário, perceberemos que o sistema faz algumas validações. Por exemplo, ele não permite que seja cadastrado um leilão sem nome, valor inicial ou data de abertura.

Ou seja, se entrarmos na página de cadastro e clicarmos em "Salvar" sem preencher nenhum dos campos, continuaremos na mesma página e algumas mensagens de erro serão exibidas. Nesse cenário, testaremos se as validações são executadas e o cadastro não é bem sucedido quando alguém tenta cadastrar um leilão com valores inválidos.

Na classe de teste `LeiloesTest`, criaremos o método `deveriaValidarCadastroDeLeilao()` com a anotação `@Test`. O início desse teste será parecido com o `deveriaCadastrarLeilao()`, já que precisaremos logar, navegar até a tela de listagem e navegar pelo formulário. A princípio, portanto, podemos copiar o início desse código.

```
LoginPage paginaDeLogin = new LoginPage();
paginaDeLogin.preencheFormularioDeLogin("fulano", "pass");
this.paginaDeLeiloes = paginaDeLogin.efetuaLogin();
CadastroLeilaoPage paginaDeCadastro = paginaDeLeiloes.carregar
```

COPIAR CÓDIGO

Perceba que para testarmos o cadastro de um leilão, sempre precisamos fazer esse passo-a-passo. Sendo assim, isolaremos esse código em um método `beforeEach()` do JUnit.

```
@BeforeEach
public void beforeEach() {
    LoginPage paginaDeLogin = new LoginPage();
    paginaDeLogin.preencheFormularioDeLogin("fulano", "pass");
    this.paginaDeLeiloes = paginaDeLogin.efetuaLogin();
    CadastroLeilaoPage paginaDeCadastro = paginaDeLeiloes.carrega
}
```

COPIAR CÓDIGO

Teremos um erro de compilação no método `deveriaCadastrarLeilao()`, pois removemos a variável `paginaDeCadastro`. Resolveremos isso criando a variável global `paginaDeCadastro`, do tipo `CadastroLeilaoPage`.

```
public class LeiloesTest {

    private LeiloesPage paginaDeLeiloes;
    private CadastroLeilaoPage paginaDeCadastro;
```

COPIAR CÓDIGO

Com isso o `beforeEach()` executará todas as anotações que definimos antes dos métodos de teste, e assim poderemos nos preocupar somente com o que é específico de cada cenário. Em `deveriaValidarCadastroDeLeilao()`, desejamos

chamar o método `paginaDeCadastro.cadastrarLeilao()`, mas sim passar as informações que ele pede. Faremos isso incluindo o valor `null` em todas as posições.

```
@Test
public void deveriaValidarCadastroDeLeilao() {
    this.paginaDeLeiloes = paginaDeCadastro.cadastrarLeilao(null,
}
```

COPIAR CÓDIGO

Agora desejamos verificar se, após o envio das informações, continuaremos na página de leilões e se as mensagens de erro estão visíveis na página. Criaremos então uma assertiva `assertTrue()` verificando se a página atual equivale à página de cadastro com o método `this.paginaDeCadastro.isPaginaAtual()`.

```
@Test
public void deveriaValidarCadastroDeLeilao() {
    this.paginaDeLeiloes = paginaDeCadastro.cadastrarLeilao(null,

    Assert.assertTrue(this.paginaDeCadastro.isPaginaAtual());
}
```

COPIAR CÓDIGO

Criaremos esse método na classe `CadastroLeilaoPage` retornando a comparação entre a URL atual (`browser.getCurrentUrl()`) e a URL de cadastro de leilão (`equals(URL_CADASTRO_LEILAO)`). Copiaremos a variável `URL_CADASTRO_LEILAO)` da classe `LeiloesPage`.

```
public class CadastroLeilaoPage {

    private static final String URL_CADASTRO_LEILAO = "http://loc

//...código omitido

    public boolean isPaginaAtual() {

        return browser.getCurrentUrl().equals(URL_CADASTRO_LEILAO
    }
}
```

COPIAR CÓDIGO

Também precisamos validar se as mensagens de erro estão aparecendo. Para isso, faremos um novo `assertTrue()`, dessa vez recebendo o método `this.paginaDeCadastro.isMensagemDeValidacaoVisiveis()`, que também criaremos na classe `CadastroLeilaoPage`.

```
@Test
public void deveriaValidarCadastroDeLeilao() {
    this.paginaDeLeiloes = paginaDeCadastro.cadastrarLeilao(null,

    Assert.assertTrue(this.paginaDeCadastro.isPaginaAtual());
    Assert.assertTrue(this.paginaDeCadastro.isPaginaDeValidacaoVi
}
```

COPIAR CÓDIGO

Existem várias maneiras de verificarmos a presença dessas mensagens, por exemplo inspecionando o elemento ou verificando o código-fonte da própria página. Usaremos a segunda abordagem. Criaremos uma variável `pageSource` c

tipo String recebendo `browser.getPageSource()`. A partir dela, usaremos o método `constains()` para verificarmos se as quatro mensagens de validação exibidas na página estão presentes no código, e retornaremos o resultado dessa verificação.

- minimo 3 caracteres
- não deve estar em branco
- deve ser um valor maior de 0.1
- deve ser uma data no formato dd/MM/yyyy

```
public boolean isPaginaDeValidacaoVisiveis() {
    String pageSource = browser.getPageSource();
    return pageSource.contains("minimo 3 caracteres")
            && pageSource.contains("não deve estar em branco")
            && pageSource.contains("deve ser um valor maior de 0.
            && pageSource.contains("deve ser uma data no formato
}
```

COPIAR CÓDIGO

Nosso teste fará essas duas verificações: se permanecemos na página de formulário e se as mensagens de validação estão visíveis quando tentamos cadastrar um leilão passando "nulo" em todos os campos. Ao executarmos o teste, teremos uma exceção do tipo "NullPointerException", que ocorre porque inicializamos uma variável local `paginaDeCadastro` ao invés de atribuirmos a chamada de `carregarFormulario` à variável global que criamos anteirormente.

```
@BeforeEach
public void beforeEach() {
    LoginPage paginaDeLogin = new LoginPage();
    paginaDeLogin.preencheFormularioDeLogin("fulano", "pass");
```

```
        this.paginaDeLeiloes = paginaDeLogin.efetuaLogin();
        this.paginaDeCadastro = paginaDeLeiloes.carregarFormulario();
    }
```

COPIAR CÓDIGO

Entretanto, ao executarmos novamente, teremos outro tipo de exceção: uma "IllegalArgumentException" com a mensagem "Keys to send should be a not null CharSequence" - ou seja, os caracteres enviados no formulário não deveriam ser uma sequência nula, pois o `sendKeys()` do Selenium não aceita o tipo `null`.

Para solucionarmos o problema, enviaremos uma string vazia ao invés de um nulo.

```
    @Test
    public void deveriaValidarCadastroDeLeilao() {
        this.paginaDeLeiloes = paginaDeCadastro.cadastrarLeilao("", "

        Assert.assertTrue(this.paginaDeCadastro.isPaginaAtual());
        Assert.assertTrue(this.paginaDeCadastro.isPaginaDeValidacaoV
    }
```

COPIAR CÓDIGO

Esse erro foi incluído propositalmente para alertar sobre o cuidado de, quando desejar testar um campo em branco, não enviar um nulo, mas sim uma string vazia. Tendo preenchido corretamente, executaremos outra vez o teste.

Dessa vez o nosso `assertTrue()` que verifica a página atual irá falhar, indicando que não estamos na página cadastro. Ao testarmos manualmente, perceberemos que isso ocorre porque, quando tentamos salvar um formulário branco, a URL

deixa de ser http://localhost:8080/leiloes/new (http://localhost:8080/leiloes/new) e se torna somente http://localhost:8080/leiloes (http://localhost:8080/leiloes).

Uma maneira de corrigirmos isso é substituindo o `assertTrue()` dessa verificação por um `assertFalse()`. Também podemos incluir um novo `assertTrue()` que verifica se a página atual é a página de leilões, já que ambas possuem a mesma URL.

```
@Test
public void deveriaValidarCadastroDeLeilao() {
    this.paginaDeLeiloes = paginaDeCadastro.cadastrarLeilao("", '

    Assert.assertFalse(this.paginaDeCadastro.isPaginaAtual());
    Assert.assertTrue(this.paginaDeLeiloes.isPaginaAtual());
    Assert.assertTrue(this.paginaDeCadastro.isPaginaDeValidacaoV
}
```

COPIAR CÓDIGO

Criaremos em `LeiloesPage` o método `isPaginaAtual()`, que retorna a comparação entre a URL atual (`getCurrentURL()`) e a URL de leilões (`equals(URL_LEILOES)`). Também criaremos a variável `URL_LEILOES` recebendo o endereço http://localhost:8080/leiloes (http://localhost:8080/leiloes).

```
public class LeiloesPage {

    private static final String URL_CADASTRO_LEILAO = "http://loc
    private static final String URL_LEILOES = "http://localhost:8
    private WebDriver browser;

//...código omitido
```

```
    public boolean isPaginaAtual() {
        return browser.getCurrentUrl().equals(URL_LEILOES);
    }
}
```

COPIAR CÓDIGO

Ao executarmos, nosso teste passará corretamente. Assim, conseguimos verificar tanto o cadastro bem sucedido de um leilão quanto a validação que ocorre quando deixamos de preencher os campos do formulário.