

Relatório Geral de Testes  
Relatório de testes do Projeto Upload

Testador: Marcus Vinicius Carvalho da Silva  
URL do site testado: <https://automacaocombatista.herokuapp.com/>  
<https://github.com/MarcusVini476/ProjetoUpload>

versão 1.0

# Índice

1. Introdução.....	3
• 1.1. Objetivo.....	3
2. Resumo dos resultados de teste.....	3
3. Cobertura do código.....	4
• 3.1 TesteDeUpload.feature e POM.xml.....	4
• 3.2 BaseTest.java e hooks.java.....	6
• 3.3 Web.java.....	7
• 3.4 WebApplication.java.....	7
• 3.5 RobotSelenium.java.....	8
• 3.6 Drivers.java.....	8
• 3.7 Runner.java.....	9
• 3.8 Inicial.java.....	9
• 3.9 Treinamento.java.....	12
• 3.10 Upload.java.....	14
• 3.11 RealizaUpload.java.....	16
4. Resultados.....	18
5. Tendências de defeitos.....	19
6. Ações Sugeridas.....	19

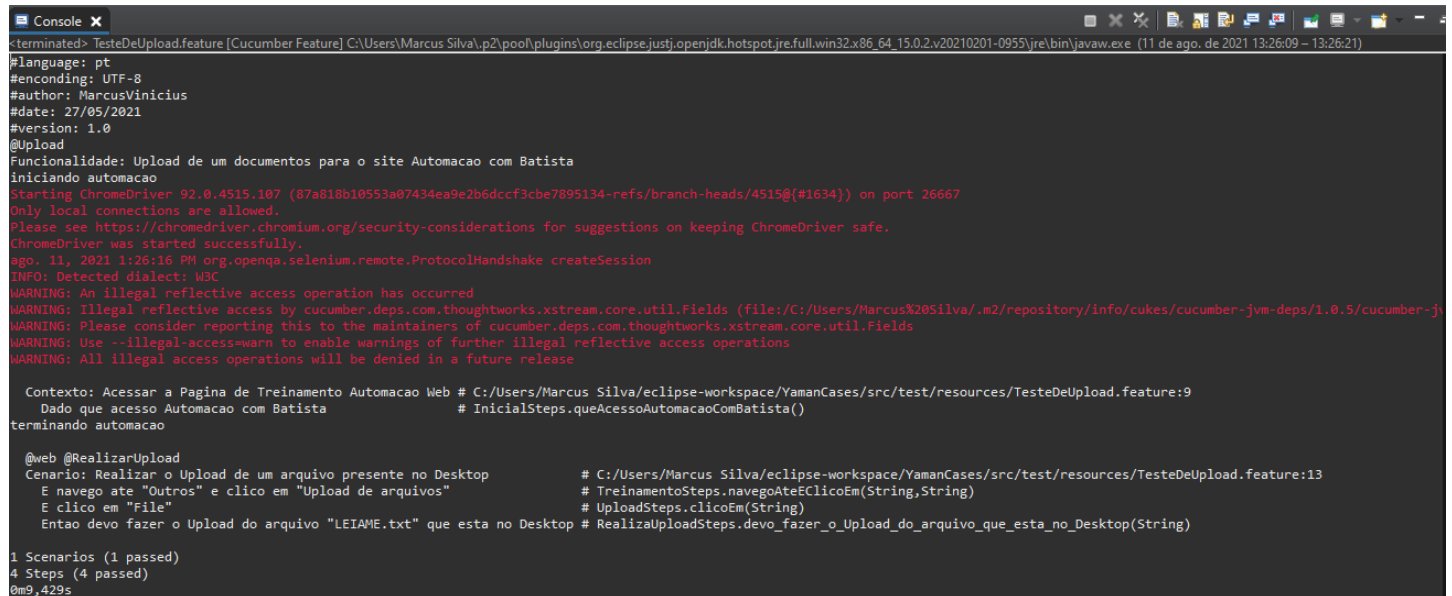
# 1. INTRODUÇÃO

## 1.1. OBJETIVO

Este relatório tem como função apresentar os resultados da automação de Upload utilizado Selenium, junit e Cucumber usando a biblioteca do selenium WebDriver, como navegador o google chrome e também estaremos mostrando evidências, sugestões de melhorias e as ferramentas utilizadas.

# 2. RESUMO DOS RESULTADOS DE TESTES

Utilizando o método BDD (Behavior Driven Development) como forma de escrita para realizar os testes, foi feita a entrada no site Automação com Batista e as interações com seus componentes. Tendo apenas 4 passos a serem seguidos, todos passaram e efetuaram suas funções.



```
<terminated> TesteDeUpload.feature [Cucumber Feature] C:\Users\Marcus Silva\.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (11 de ago. de 2021 13:26:09 - 13:26:21)
#language: pt
#encoding: UTF-8
#author: MarcusVinicius
#date: 27/05/2021
#version: 1.0
@Upload
Funcionalidade: Upload de um documentos para o site Automacao com Batista
iniciando automacao
Starting ChromeDriver 92.0.4515.107 (87a818b10553a07434ea9e2b6dccc3cbe7895134-refs/branch-heads/4515@{#1634}) on port 26667
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
ago. 11, 2021 1:26:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by cucumber.deps.com.thoughtworks.xstream.core.util.Fields (file:/C:/Users/Marcus Silva/.m2/repository/info/cukes/cucumber-jvm-deps/1.0.5/cucumber-jvm-deps-1.0.5.jar)
WARNING: Please consider reporting this to the maintainers of cucumber.deps.com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release

Contexto: Acessar a Pagina de Treinamento Automacao Web # C:/Users/Marcus Silva/eclipse-workspace/YamanCases/src/test/resources/TesteDeUpload.feature:9
Dado que acesso Automacao com Batista # InicialSteps.queAcessoAutomacaoComBatista()
terminando automacao

@web @RealizarUpload
Cenário: Realizar o Upload de um arquivo presente no Desktop # C:/Users/Marcus Silva/eclipse-workspace/YamanCases/src/test/resources/TesteDeUpload.feature:13
  E navego ate "Outros" e clico em "Upload de arquivos" # TreinamentoSteps.navegoAteEclicoEm(String,String)
  E clico em "File" # UploadSteps.clicoEm(String)
  Entao devo fazer o Upload do arquivo "LEIAME.txt" que esta no Desktop # RealizaUploadSteps.devo_fazer_o_Upload_do_arquivo_que_esta_no_Desktop(String)

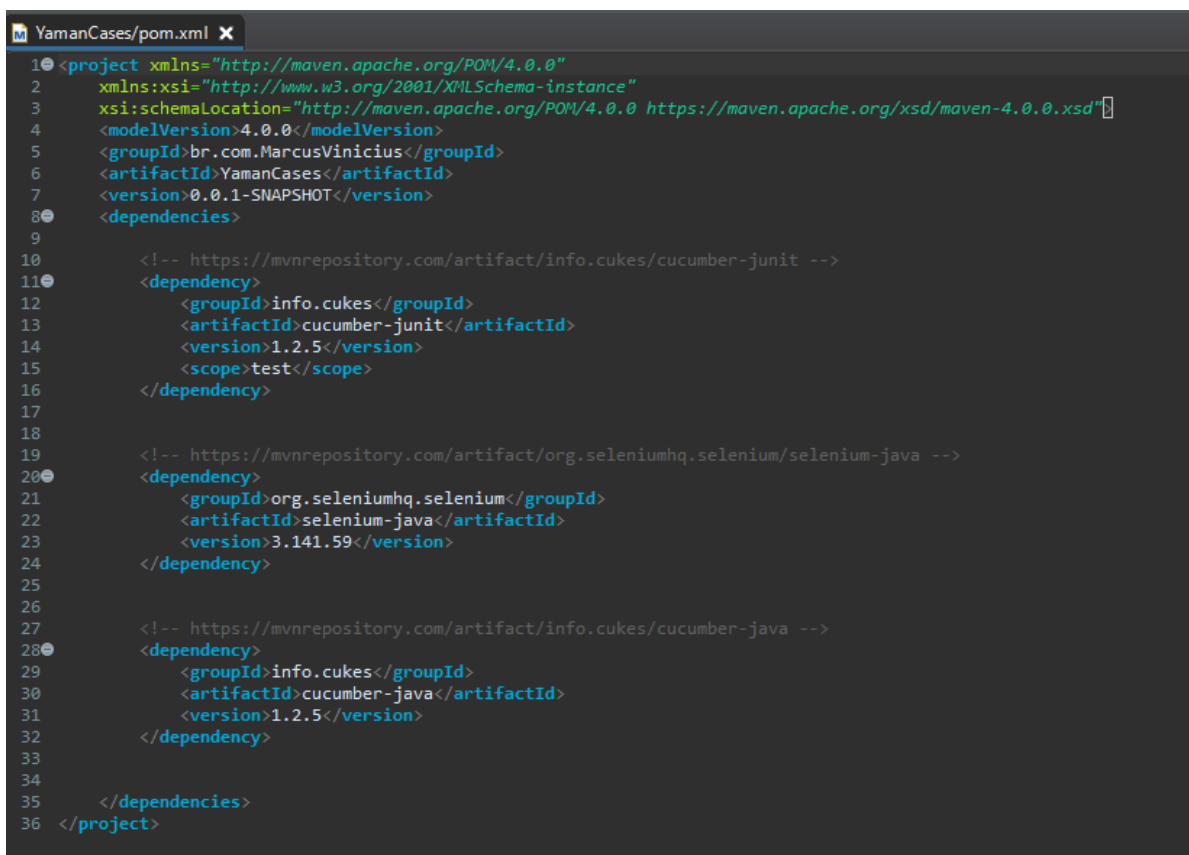
1 Scenarios (1 passed)
4 Steps (4 passed)
0m9,429s
```

## 3. COBERTURA DO CÓDIGO

Na cobertura de códigos será mostrado os prints das classes e quais suas funções no projeto.

### 3.1. TesteDeUpload.feature e POM.xml

O pom.xml é onde está todas as dependências deste projeto, no caso foi usado Cucumber-junit na versão 1.25, Cucumber-java na versão 1.2.5 e selenium-java na versão 3.141.59.



```
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.com.MarcusVinicius</groupId>
6   <artifactId>YamanCases</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <dependencies>
9
10     <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-junit -->
11     <dependency>
12       <groupId>info.cukes</groupId>
13       <artifactId>cucumber-junit</artifactId>
14       <version>1.2.5</version>
15       <scope>test</scope>
16     </dependency>
17
18     <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
19     <dependency>
20       <groupId>org.seleniumhq.selenium</groupId>
21       <artifactId>selenium-java</artifactId>
22       <version>3.141.59</version>
23     </dependency>
24
25     <!-- https://mvnrepository.com/artifact/info.cukes/cucumber-java -->
26     <dependency>
27       <groupId>info.cukes</groupId>
28       <artifactId>cucumber-java</artifactId>
29       <version>1.2.5</version>
30     </dependency>
31
32   </dependencies>
33
34 </project>
```

O “TesteDeUpload.feature” é o arquivo BDD (Behavior Driven Development) e nele está todos os passos a serem seguidos a história de usuário, transformada em contexto da funcionalidade, o cenário a ser testado e as tags @web @RealizarUpload e @Upload, são reconhecidas pelo Junit ao ler o BDD.

```
TesteDeUpload.feature X
1 #language: pt
2 #encoding: UTF-8
3 #author: MarcusVinicius
4 #date: 27/05/2021
5 #version: 1.0
6 @Upload
7 ⑨ Funcionalidade: Upload de um documentos para o site Automacao com Batista
8
9 ⑨ Contexto: Acessar a Pagina de Treinamento Automacao Web
10 ④ Dado que acesso Automacao com Batista
11
12 @web @RealizarUpload
13 ⑨ Cenario: Realizar o Upload de um arquivo presente no Desktop
14 ④ E navego ate "Outros" e clico em "Upload de arquivos"
15 ④ E clico em "File"
16 ④ Entao devo fazer o Upload do arquivo "LEIAME.txt" que esta na pasta "ParaUpload"
17
```

## 3.2 BaseTest.java e hooks

Na BaseTest.java será estabelecido a inicialização, o fechamento do navegador e o tempo de espera para realizar as funções.

```
BaseTest.java X
1 package br.com.MarcusVinicius.configuracoes;
2
3
4
5 import org.openqa.selenium.WebDriver;
6
7
8
9
10 public abstract class BaseTest {
11
12     protected static WebDriver navegador;
13     protected static WebDriverWait espera;
14
15     protected void inicializarAplicacaoWeb(WebApplication webApplication, String url) {
16         if (navegador != null) {
17             navegador.quit();
18         }
19         navegador = webApplication.getWebDriver();
20         navegador.manage().window().maximize();
21         navegador.get(url);
22
23         espera = new WebDriverWait(navegador, 60);
24     }
25
26     protected static void fecharWeb() {
27
28         navegador.close();
29     }
30 }
31
32
```

Nos Hooks temos a SetUp e a tearDown, pois quando for iniciado o projeto, a setUp vai cuidar de disponibilizar a URL do site, realizar todos os passos do cucumber e quando tudo for realizado, o tearDown vai cuidar de fechar o navegador, assim finalizando os testes.

```
hooks.java X
1 package br.com.MarcusVinicius.configuracoes;
2
3 import br.com.MarcusVinicius.enums.Web;
4
5
6
7 public class hooks extends BaseTest {
8
9     @Before
10     public void setUp() {
11         System.out.println("iniciando automacao");
12         inicializarAplicacaoWeb(Web.CHROME, "https://automacaocombatista.herokuapp.com/");
13     }
14
15     @After
16     public void tearDown() {
17         System.out.println("terminando automacao");
18
19         fecharWeb();
20     }
21
22 }
23
```

## 3.3 Web

Na Web.java estamos colocando uma condição a ser realizada ao abrir o navegador, nesse caso essa condição é inibir notificações de aparecerem durante a execução do teste.

```
Web.java x
1 package br.com.MarcusVinicius.enums;
2
3 import org.openqa.selenium.WebDriver;
4
5 public enum Web implements WebApplication {
6     CHROME {
7         public WebDriver getWebDriver() {
8             System.setProperty("webdriver.chrome.driver",
9                 "C:\\\\Users\\Marcus Silva\\eclipse-workspace\\YamanCases\\src\\main\\resources\\drivers\\chromedriver.exe");
10             ChromeOptions opcoes = new ChromeOptions();
11             opcoes.addArguments("--disable-notifications");
12
13             return new ChromeDriver(opcoes);
14         }
15     }
16 }
17
18 }
```

## 3.4 WebApplication

Na WebApplication declaramos um WebDriver constante.

```
WebApplication.java x
1 package br.com.MarcusVinicius.interfaces;
2
3 import org.openqa.selenium.WebDriver;
4
5 public interface WebApplication {
6     WebDriver getWebDriver();
7 }
8
9 }
```

## 3.5 RobotSelenium

O Robot é o elemento de controle na hora dos testes, pois alguns botões só aparecem na tela depois de certo tempo, e para evitar que a automação efetue erros, esta classe só permite o clique de um certo botão, quando o mesmo aparecer na tela.

```
RobotSelenium.java X
1 package br.com.MarcusVinicius.robot;
2
3 import org.openqa.selenium.WebElement;
4
5
6
7
8 public class RobotSelenium extends BaseTest{
9     public void AguardeFicarVisivel(WebElement element) {
10         espera.until(ExpectedConditions.elementToBeClickable(element));
11         element.click();
12     }
13
14
15 }
16
```

## 3.6 Drivers

A versão do google chrome no período desta automação está na versão 92.0.4515.131, a versão mais recente do ChromeDriver para windows é a 92.0.4515.107.

```
Starting ChromeDriver 92.0.4515.107 (87a818b10553a07434ea9e2b6dccc3cbe7895134-refs/branch-heads/4515@{#1634}) on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

Sobre o Google Chrome



Google Chrome

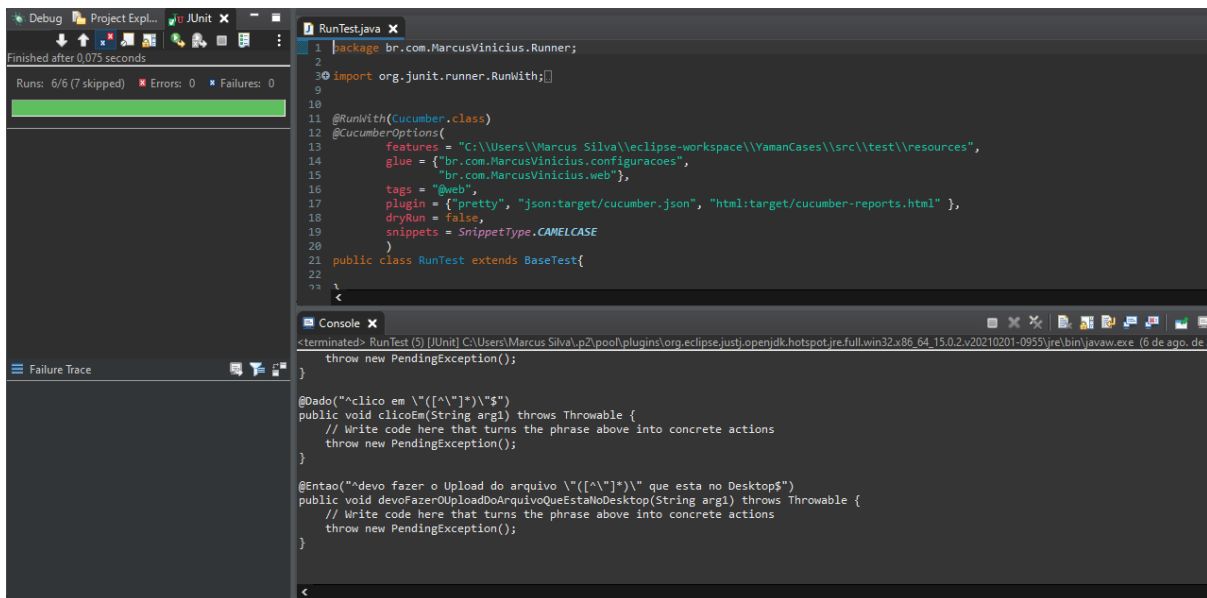


O Google Chrome está atualizado  
Versão 92.0.4515.131 (Versão oficial) 64 bits



## 3.7 Runner

A Runner está mapeando em sua configuração a pasta resource que é onde se encontra a nossa .feature, então os passos serão executados, e no console nos é devolvido às snippets formatadas no padrão CamelCase, e as mesmas serão implementadas nos arquivos com steps em seus nomes.



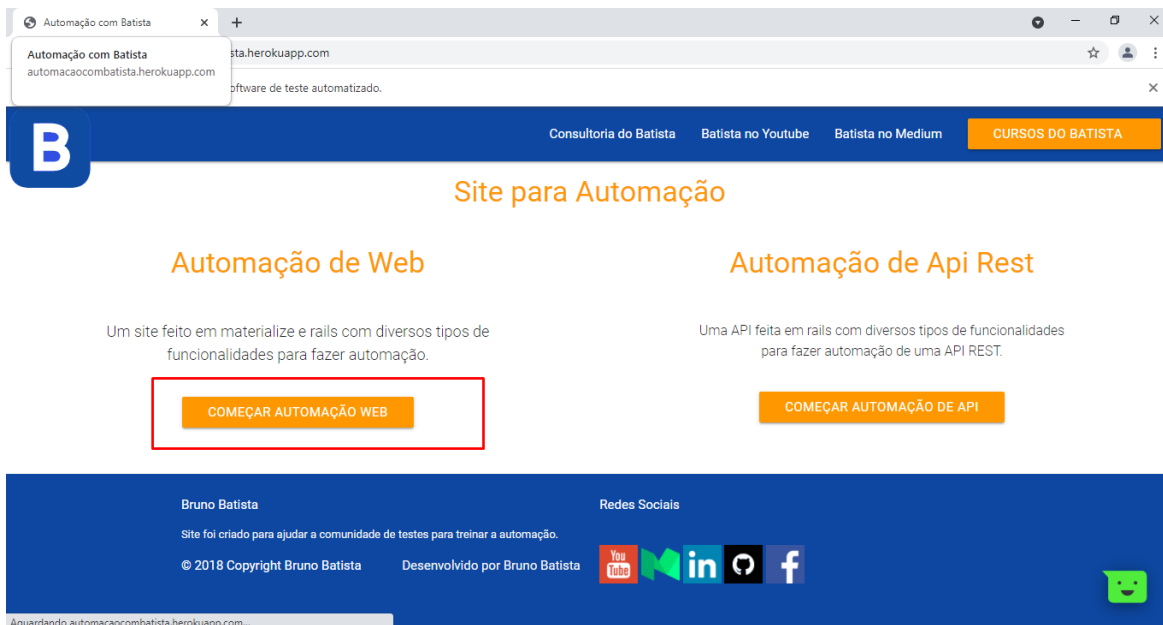
The screenshot shows an IDE with two main panels. The top panel displays the source code of a Java class named `RunTest.java`. The code is as follows:

```
1 package br.com.MarcusVinicius.Runner;
2
3 import org.junit.runner.RunWith;
4
5
6
7
8
9
10
11 @RunWith(Cucumber.class)
12 @CucumberOptions(
13     features = "C:\\Users\\Marcos Silva\\eclipse-workspace\\YamanCases\\src\\test\\resources",
14     glue = {"br.com.MarcusVinicius.configuracoes",
15           "br.com.MarcusVinicius.web"},
16     tags = "@web",
17     plugin = {"pretty", "json:target/cucumber.json", "html:target/cucumber-reports.html"},
18     dryRun = false,
19     snippets = SnippetType.CAMEL_CASE
20 )
21 public class RunTest extends BaseTest{
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The bottom panel shows the console output, which is currently empty, indicating that the test has not yet been executed or that the output is being filtered.

## 3.8 Inicial

Os arquivos de nome Inicial representam a funcionalidade da página principal, onde após o navegador ser aberto, a url ser pesquisada, na pagina que nos é apresentada, devemos clicar no botão “Começar Automação Web”.



Na “InicialPage.java” encontramos o botão “Começar Automação Web” pelo Xpath e então retornando o “WebElement” “getBtnIniciarAutomacao”.

```
1 package br.com.MarcusVinicius.pages;
2
3 import org.openqa.selenium.WebDriver;
4
5
6
7
8 public class InicialPage {
9     public InicialPage(WebDriver navegador) {
10
11         PageFactory.initElements(navegador, this);
12     }
13
14     @FindBy(xpath = "//a[text()='Começar Automação Web']")
15     private WebElement btnIniciarAutomacao;
16
17     public WebElement getBtnIniciarAutomacao() {
18         return btnIniciarAutomacao;
19     }
20
21
22 }
23
```

Na “InicialFuncionalidade.java” é feita a ação na página, nesse caso, ao declararmos um “WebElement” de nome “botao” que recebe o “getBtnIniciarAutomacao” da inicial page, é aguardado o botão ficar visível e então é simulado o “click”.

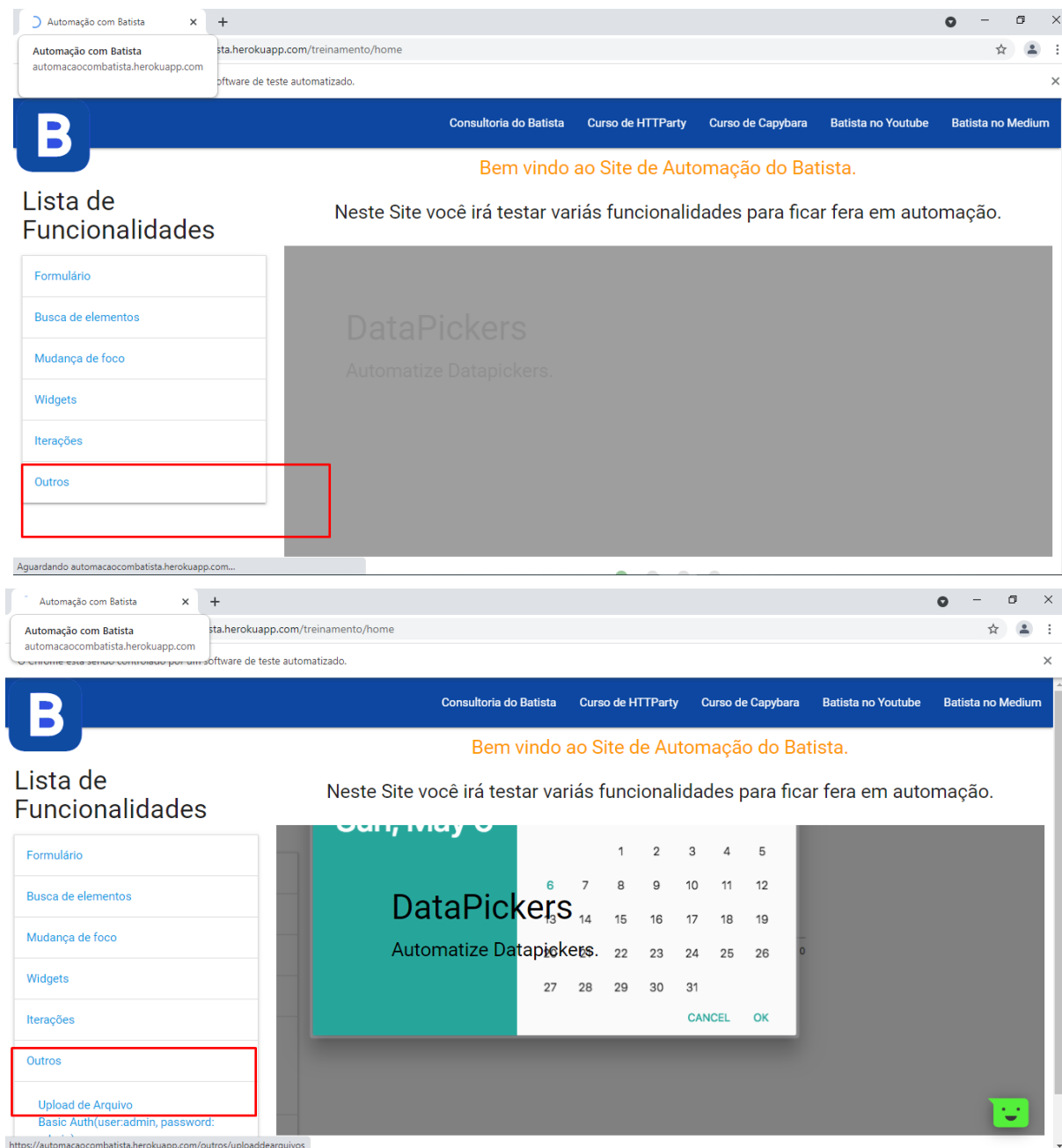
```
InicialFuncionalidades.java X
1 package br.com.MarcusVinicius.funcionalidades;
2
3 import org.openqa.selenium.WebElement;
4
5
6
7
8
9
10 public class InicialFuncionalidades extends BaseTest{
11     private InicialPage inicialPage;
12
13     public InicialFuncionalidades() {
14         this.inicialPage = new InicialPage(navegador);
15     }
16
17     public void clicaNoBotaoComecarAutomacao() {
18         WebElement botao = this.inicialPage.getBtnIniciarAutomacao();
19         espera.until(ExpectedConditions.visibilityOf(botao));
20         botao.click();
21     }
22
23 }
24
```

Na “InicialSteps.java” vai o nosso primeiro snippet que o Runner nos devolveu, e esta classe vai efetuar o “clicaNoBotaoComecarAutomacao”.

```
InicialPage.java X
1 package br.com.MarcusVinicius.pages;
2
3 import org.openqa.selenium.WebDriver;
4
5
6
7
8 public class InicialPage {
9     public InicialPage(WebDriver navegador) {
10
11         PageFactory.initElements(navegador, this);
12     }
13
14     @FindBy(xpath = "//a[text()='Começar Automação Web']")
15     private WebElement btnIniciarAutomacao;
16
17     public WebElement getBtnIniciarAutomacao() {
18         return btnIniciarAutomacao;
19     }
20
21
22 }
23
```

## 3.9 Treinamento

Passando da Página inicial, entramos na área de treinamento e nos é apresentado uma lista de opções, sabendo que o objetivo deste teste é fazer o Upload de um arquivo que, então devemos clicar em “Outros”, aguardar a sub-opções ficarem visíveis e clicar em “Upload de arquivo”.



Na “TreinamentoPage.java” estamos declarando o caminho do botões “Outros” e “Upload de arquivos” por seus xpaths e retornando seus WebElements “getLinkUpload” e “getLinkUploadDeArquivo”.

```
TreinamentoPage.java X
1 package br.com.MarcusVinicius.pages;
2
3 import org.openqa.selenium.WebDriver;
4
5 public class TreinamentoPage {
6     public TreinamentoPage(WebDriver webDriver) {
7         PageFactory.initElements(webDriver, this);
8     }
9     @FindBy(xpath = "//a[text()='Outros']")
10    private WebElement LinkUpload;
11
12    public WebElement getLinkUpload(){
13        return LinkUpload;
14    }
15    @FindBy(xpath = "//a[text()='Upload de Arquivo']")
16    private WebElement LinkUploadDeArquivo;
17
18    public WebElement getLinkUploadDeArquivo(){
19        return LinkUploadDeArquivo;
20    }
21
22 }
23
24
25
26
27
```

Juntamente com a nossa classe “RobotSelenium.java” a função dessa classe fará com que após clicar em outros, o selenium vai esperar o “getLinkUploadDeArquivo” ficar visível para poder efetuar o “click”.

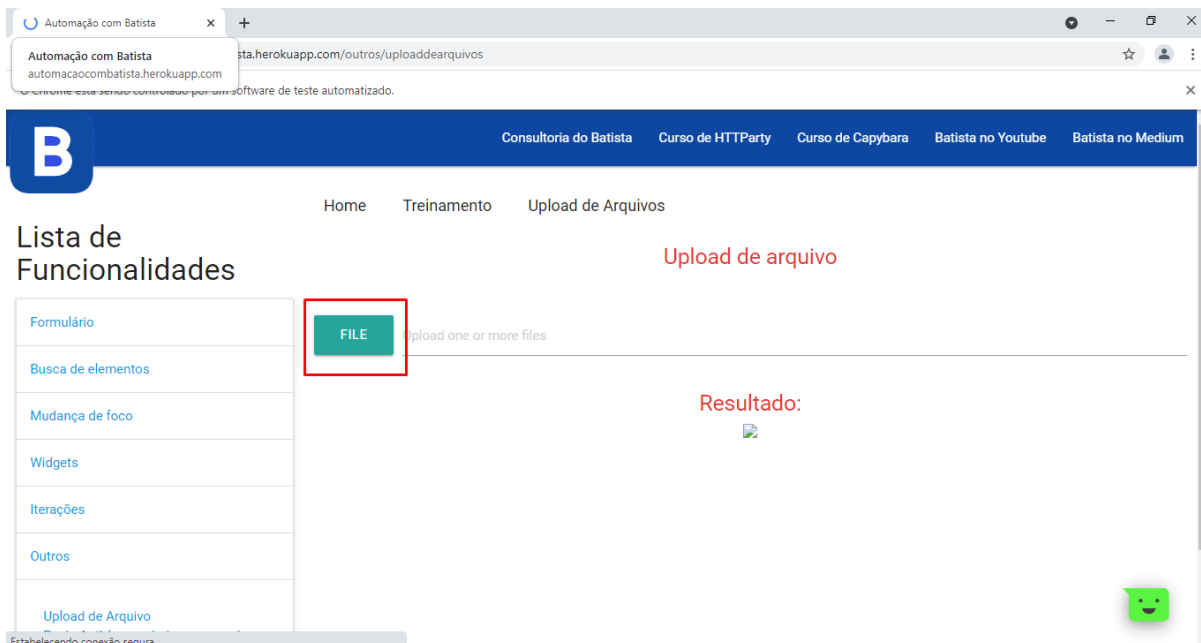
```
TreinamentoFuncionalidades.java X
1 package br.com.MarcusVinicius.funcionalidades;
2
3
4
5 import br.com.MarcusVinicius.configuracoes.BaseTest;
6
7 public class TreinamentoFuncionalidades extends BaseTest {
8     private TreinamentoPage treinamentoPage;
9     private RobotSelenium robotSelenium;
10
11    public TreinamentoFuncionalidades() {
12        this.treinamentoPage = new TreinamentoPage(navegador);
13        this.robotSelenium = new RobotSelenium();
14    }
15    public void clicaNoLinkOutros() {
16        this.robotSelenium.AguardeFicarVisivel(this.treinamentoPage.getLinkUpload());
17    }
18
19    public void clicaNoLinkUploadDeArquivos() {
20        this.robotSelenium.AguardeFicarVisivel(this.treinamentoPage.getLinkUploadDeArquivo());
21    }
22
23 }
24
25
26
27
28
```

Na “TreinamentoSteps.java”, primeiro será efetuado o “clicaNoLinkOutros” e depois “clicaNoLinkUploadDeArquivos”.

```
TreinamentoSteps.java X
1 package br.com.MarcusVinicius.steps;
2
3 import br.com.MarcusVinicius.funcionalidades.TreinamentoFuncionalidades;
4
5
6 public class TreinamentoSteps {
7     private TreinamentoFuncionalidades treinamentoFuncionalidade;
8
9     public TreinamentoSteps() {
10         this.treinamentoFuncionalidade = new TreinamentoFuncionalidades();
11     }
12
13     @Dado("^navego ate \"([^\"]*)\" e clico em \"([^\"]*)\"$")
14     public void navegoAteEClicoEm(String arg1, String arg2) throws Throwable {
15         this.treinamentoFuncionalidade.clicaNoLinkOutros();
16         this.treinamentoFuncionalidade.clicaNoLinkUploadDeArquivos();
17     }
18 }
19
```

## 3.10 Upload

Agora que conseguimos entrar na página desejada, vamos clicar no botão “FILE” e então realizar o Upload do arquivo “LEIAME.txt” presente na pasta “ParaUpload”.



Em "UploadPage.java" é feita a declaração do xpath de "FILE" e nos é retornado o "WebElement" "getLinkFile".

```
UploadPage.java X
1 package br.com.MarcusVinicius.pages;
2
3 import org.openqa.selenium.WebDriver;
4
5
6
7
8 public class UploadPage {
9     public UploadPage(WebDriver webDriver) {
10         PageFactory.initElements(webDriver, this);
11     }
12
13     @FindBy(xpath = "//input[@id='upload']")
14     private WebElement LinkFile;
15
16     public WebElement getLinkFile() {
17         return LinkFile;
18     }
19
20
21 }
22
```

Em "UploadFuncionalidades.java" a "getLinkFile" será clicada.

```
UploadFuncionalidades.java X
1 package br.com.MarcusVinicius.funcionalidades;
2
3 import br.com.MarcusVinicius.configuracoes.BaseTest;
4
5
6
7 public class UploadFuncionalidades extends BaseTest {
8     private UploadPage uploadPage;
9
10
11     public UploadFuncionalidades() {
12         this.uploadPage = new UploadPage(navegador);
13     }
14
15
16     public void clicaEmFile() {
17         this.uploadPage.getLinkFile();
18     }
19
20 }
21
```

A “UploadSteps.java” realiza as funcionalidades de “UploadFuncionalidades.java”

```
UploadSteps.java x
1 package br.com.MarcusVinicius.steps;
2
3 import br.com.MarcusVinicius.funcionalidades.UploadFuncionalidades;
4
5
6
7 public class UploadSteps {
8     private UploadFuncionalidades uploadFuncionalidades;
9
10    public UploadSteps() {
11        this.uploadFuncionalidades = new UploadFuncionalidades();
12    }
13
14    @Dado("^clico em \"([^\"]*)\"$")
15    public void clicoEm(String arg1) throws Throwable {
16
17        this.uploadFuncionalidades.clicaEmFile();
18    }
19
20 }
21
```

### 3.11 RealizaUpload

Clicando em “FILE” o próximo e último passo será realizar o Upload do arquivo “LEIAME.txt” que se encontra na pasta “ParaUpload” no site.

Este Computador > Windows (C:) > Usuários > Marcus Silva > eclipse-workspace > ProjetoUpload > ParaUpload				
	Nome	Data de modificação	Tipo	Tamanho
do abalho	LEIAME	02/08/2021 11:42	Documento de Te...	1 KB



A “RealizaUploadPage.java” está utilizando o mesmo xpath de “FILE” porém sua função não será um clique, mas inserir o caminho em que o arquivo “LEIAME.txt” se encontra.

```
RealizaUploadPage.java X
1 package br.com.MarcusVinicius.pages;
2
3 import org.openqa.selenium.WebDriver;
4
5
6
7
8 public class RealizaUploadPage {
9
10     public RealizaUploadPage(WebDriver webDriver) {
11         PageFactory.initElements(webDriver, this);
12     }
13
14     public WebElement getCampoRealizaUpload() {
15         return campoRealizaUpload;
16     }
17
18     @FindBy(xpath = "//input[@id='upload']")
19     private WebElement campoRealizaUpload;
20
21
22
23
24 }
25
```

Em “RealizaUploadFuncionalidades.java” é enviado a coordenadas de onde está o arquivo para upload, através do “sendKeys”.

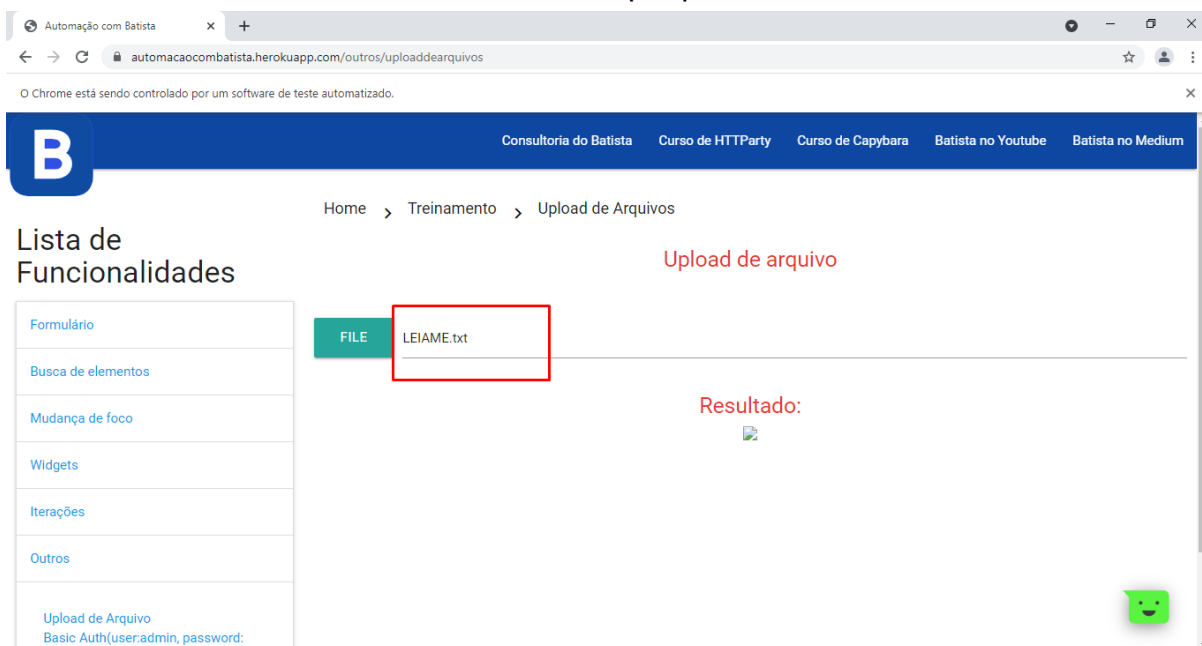
```
RealizaUploadFuncionalidades.java X
1 package br.com.MarcusVinicius.funcionalidades;
2
3 import br.com.MarcusVinicius.configuracoes.BaseTest;
4
5
6 public class RealizaUploadFuncionalidades extends BaseTest{
7     private RealizaUploadPage realizaUploadPage;
8
9
10     public RealizaUploadFuncionalidades() {
11         this.realizaUploadPage = new RealizaUploadPage(navegador);
12     }
13
14     public void preencherCampoDePesquisa() {
15         this.realizaUploadPage.getCampoRealizaUpload().sendKeys
16             ("C:\\Users\\Marcus Silva\\eclipse-workspace\\ProjetoUpload\\ParaUpload\\LEIAME.txt");
17     }
18
19
20 }
21
```

O último passo realizado é o Upload do Arquivo “LEIAME.txt” que se encontra em “ParaUpload”.

```
RealizaUploadSteps.java x
1 package br.com.MarcusVinicius.steps;
2
3 import br.com.MarcusVinicius.funcionalidades.RealizaUploadFuncionalidades;
4
5
6
7 public class RealizaUploadSteps {
8     private RealizaUploadFuncionalidades realizaUploadFuncionalidades;
9
10    public RealizaUploadSteps() {
11        this.realizaUploadFuncionalidades = new RealizaUploadFuncionalidades();
12    }
13
14
15    @Entao("^devo fazer o Upload do arquivo \"([^\"]*)\" que esta na pasta \"([^\"]*)\"")
16    public void devoFazerOUploadDoArquivoQueEstaNaPasta(String arg1, String arg2) throws Throwable {
17        this.realizaUploadFuncionalidades.preencherCampoDePesquisa();
18    }
19
20 }
21
```

## 4. Resultados

Os resultados foram um sucesso e as 4 “steps” passaram.



## 5. Tendências de defeitos

Para estes testes serem feitos é recomendado que o Chromedriver do projeto seja da mesma versão que o seu navegador Chrome para que não ocorra erros na hora de rodar a automação.

## 6. Ações Sugeridas

Estar sempre atento a versão do seu navegador e baixar a versão mais atualizado do site

<https://chromedriver.chromium.org/downloads>.