

Candidate: Valnei Filho

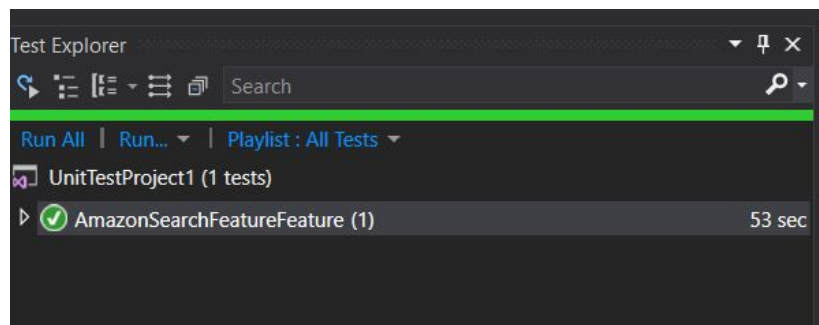
E-mail: v\_marinpietri@yahoo.com.br

## Session 2

### The Scenario

```
Feature: AmazonSearchFeature
@mytag1
Scenario: Search result of the books in amazon site
    Given I navigate to www.amazon.com
    When I select the option books in the dropdown next to the search text input criteria
        | keyword |
        | Books |
    Then I search for Test automation
        | keyword |
        | test automation |
    And I select the cheapset book of the page without using any sorting method available
    When I reach the detailed book page, I check if the name in the header is the same name of the book that I selected previously
```

### Evidence to pass test



## 1) Find the lowest price for a type paperback book

```
[Then(@"I select the cheapset book of the page without using any sorting method available")]
public void SelectTheCheapsetBook()
{
    IList<IWebElement> all = _currentDriver.FindElements(By.XPath("//a[@class='a-link-normal a-text-normal']"));

    var i = 1;
    //define a pattern for expression regular
    string pattern = @"Paperback \${[0-9]{2}} [0-9]{2}";

    //Iterator dynamically for elements for page
    foreach (IWebElement element in all)
    {
        //get childs of class sg_col_inner
        var getClassSgColInner = _currentDriver.FindElements(By.XPath($"//*[ @id=\\"search\\" ]/div[1]/div[2]/div/span[4]/div[1]/div[{i}]/div/span/div/div/div[2]/div[2]/div/div[2]/div/d"));
        foreach (var item in getClassSgColInner)
        {
            var regex = new Regex(pattern);
            var str = item.Text.Replace(oldValue: "\n", newValue: "").Replace(oldValue: "\r", newValue: " ");

            if (regex.IsMatch(str))
            {
                var strPaperBack = regex.Matches(str)[0].Value;
                var strPrince = strPaperBack.Split(params separator: '$')[1].Replace(oldChar: ' ', newChar: '.');

                _list.Add(new Documents
                {
                    Name = element.Text,
                    Link = element.GetAttribute(attributeName: "href"),
                    PriceToPaperback = decimal.Parse(strPrince)
                });
            }
        }
        i++;
    }
}
```

## 2) Show the cheapset

```
[20/11/2019 17:21:43 Informational] When I reach the detailed book page, I check if the name in the header is the same name of the book that I selected previously
[20/11/2019 17:22:07 Informational] =====
[20/11/2019 17:22:07 Informational] That's it the lowest price found in the first page
[20/11/2019 17:22:07 Informational] Book Name: The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations
[20/11/2019 17:22:07 Informational] Link: https://www.amazon.com/DevOps-Handbook-World-Class-Reliability-Organizations/dp/1942788002/ref=sr\_1\_4?keywords=test+automation&qid=1574281292&s=books&sr=1-4
[20/11/2019 17:22:07 Informational] Lowest price to Paperback: 18.69
[20/11/2019 17:22:07 Informational] By Valnei Filho v_marinpietri@yahoo.com.br 71 99946-7636, thks so much!
[20/11/2019 17:22:07 Informational] =====
[20/11/2019 17:22:11 Informational] -> done: AmazonSearchFeatureSteps.WhenIReachTheDetailedBookPageICheckIfTheNameInTheHeaderIsTheSameNameOfTheBookThatISelectedPreviously() (27,6s)
[20/11/2019 17:22:11 Informational] NUnit VS Adapter 2.2.0.0 executing tests is finished
[20/11/2019 17:22:11 Informational] ===== Run test finished: 1 run (0:00:54,708531) =====
```