Automação de Teste

21/09/2019

hapvida
Faz bem pra você

INSTITUTO ATLÂNTICO
EXCELÊNCIA EM SOLUÇÕES TECNOLÓGICAS

AGENDA

- O que é o Cucumber?
- O que é BDD? (Desenvolvimento orientado por comportamento)
- Instalando o Cucumber;
- Estrutura do Projeto;
- Trabalhando com Cucumber;
- Revendo os cenários de teste;
- Um pouco de automatização.

O que é o Cucumber?

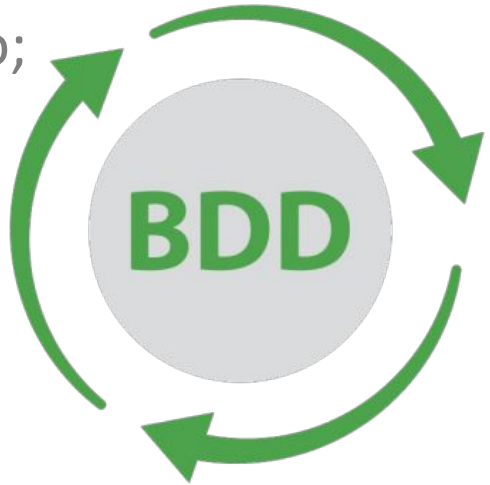
O Cucumber é uma ferramenta que suporta Behavior Driven Development (**BDD**), que consiste em descrever o comportamento de um usuário. Dessa forma, as necessidades reais do usuário são descritas.

O que é BDD?

BDD serve para criar testes e integrar regras de negócios com a linguagem de programação, focando no comportamento do software. Além disso, ainda melhora a comunicação entre as equipes de desenvolvimento e testes, aumentando o compartilhamento de conhecimento entre elas.

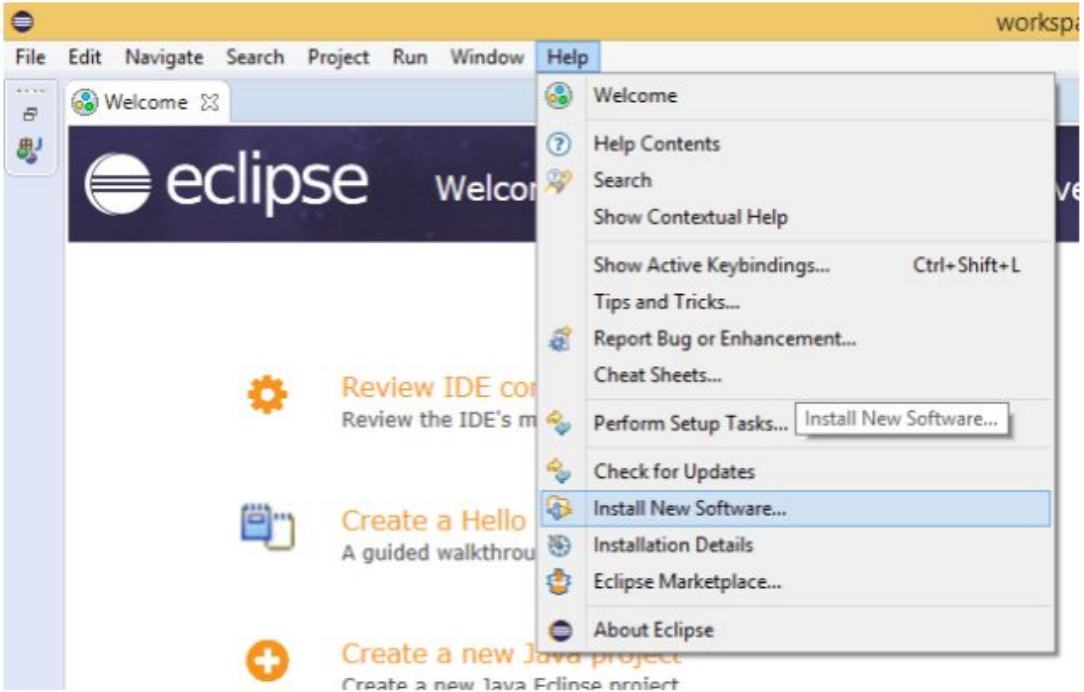
Vantagens em usar BDD?

- Comunicação entre equipes;
- Compartilhamento de conhecimento;
- Documentação dinâmica;
- Visão do todo.

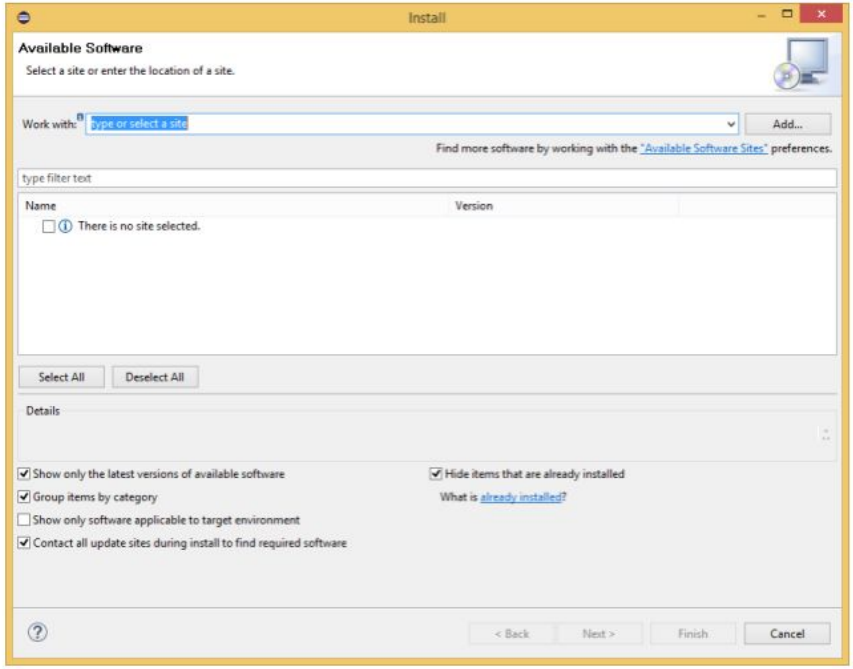


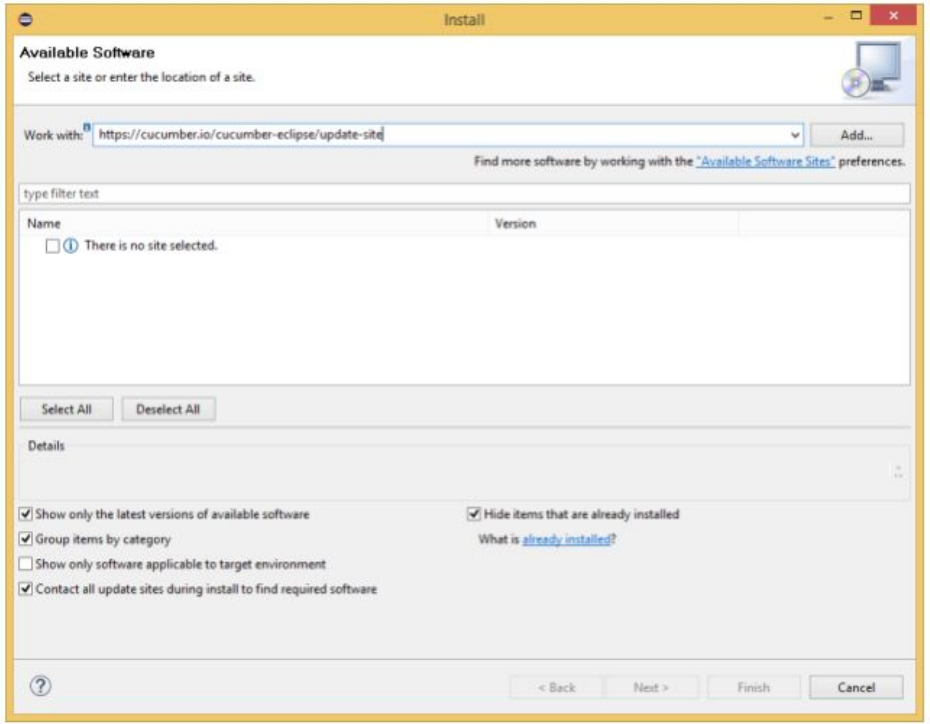
Instalando o Cucumber

Uma vez instalado o JDK e o Eclipse, é hora de instalar o plugin do Cucumber. Para isso, acesse o menu 'Help > Install New Software' no Eclipse:

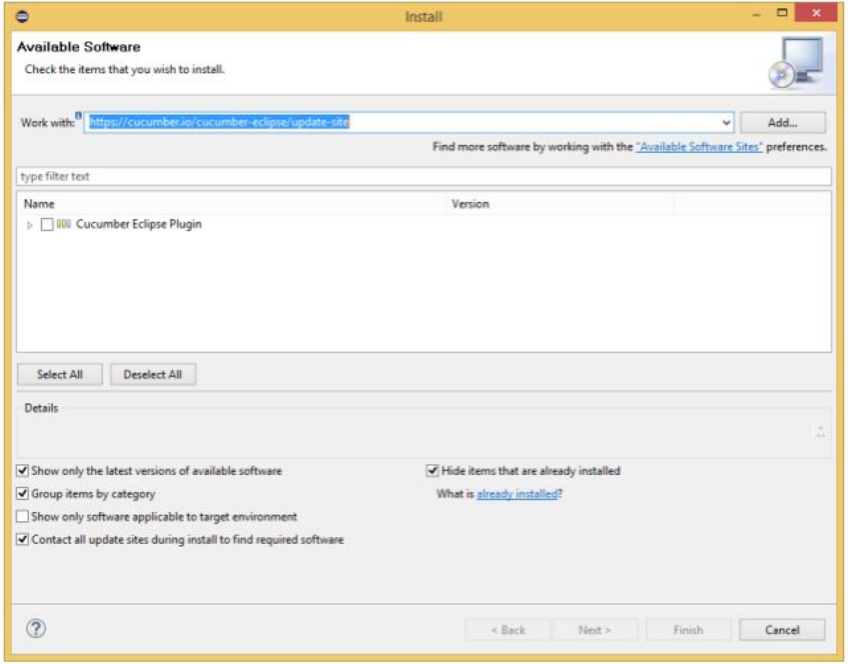


Na janela que abrir, digite a URL https://cucumber.io/cucumber-eclipse/update-site e tecle “ENTER”:

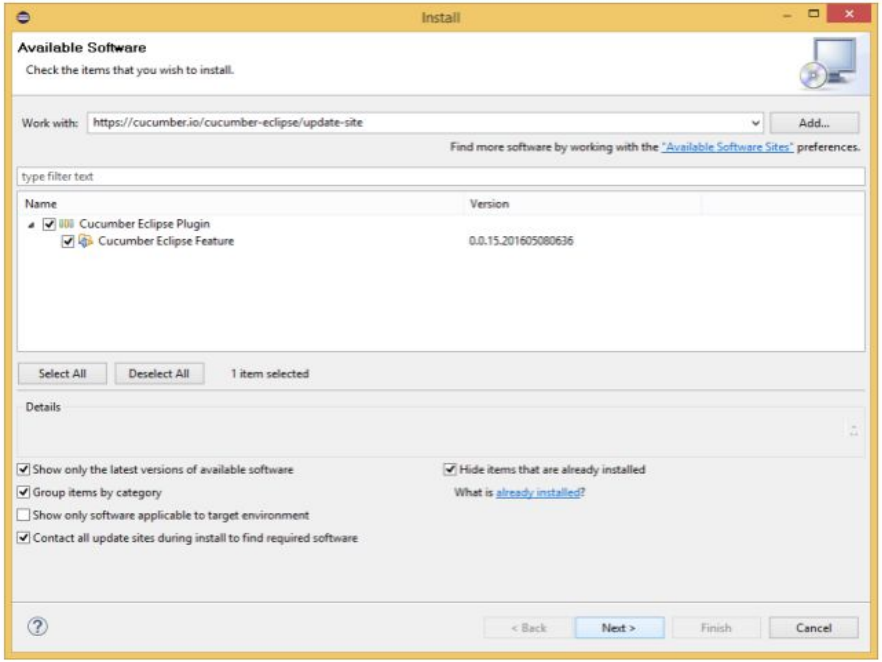




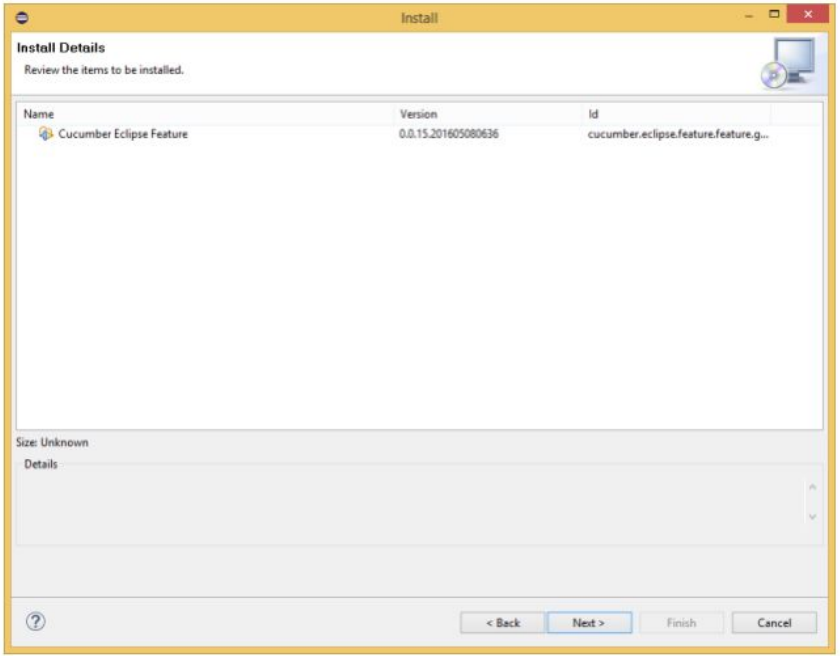
Será exibido o item “Cucumber Eclipse Plugin”:



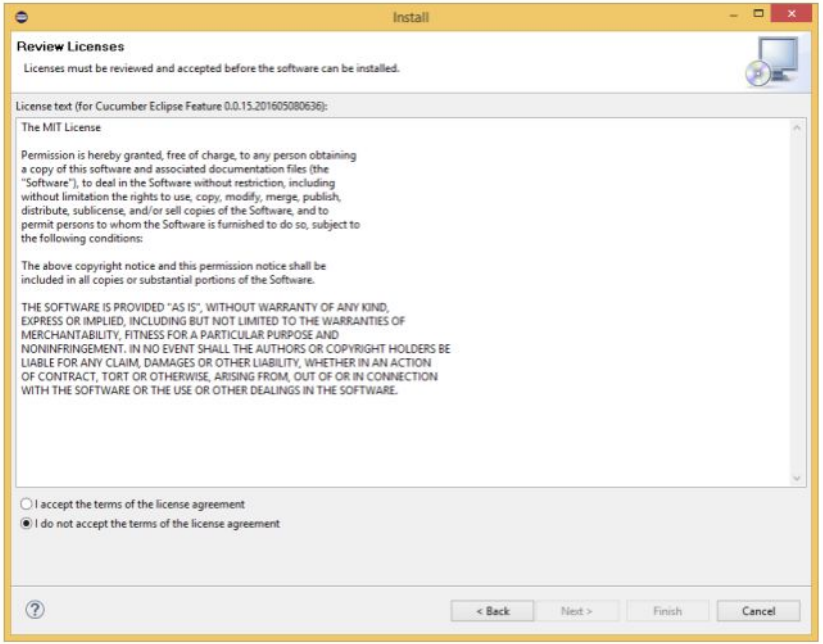
Marque o checkbox e clique em “Next”:



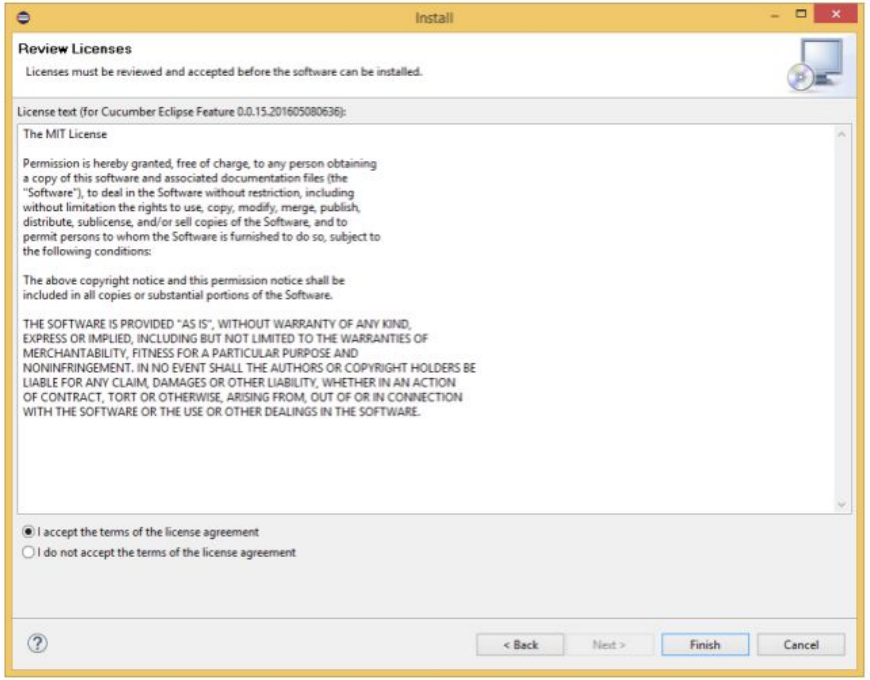
Será exibida a tela abaixo, mostrando detalhes da instalação:



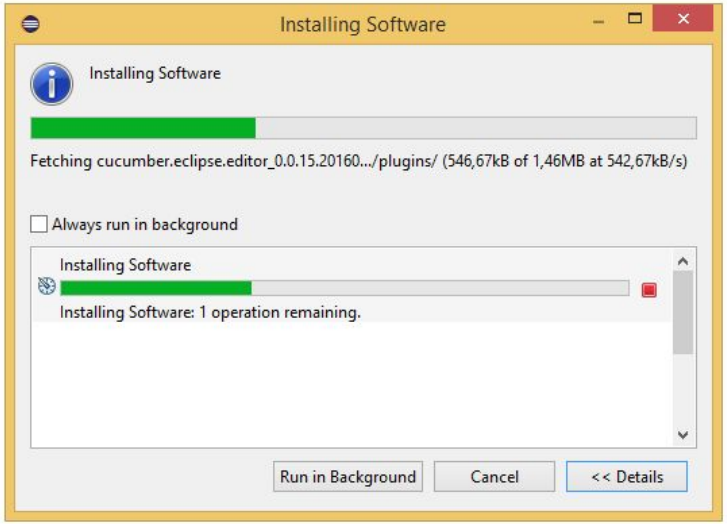
Ao clicar em “Next” novamente, será exibida a tela abaixo:



Clique em “I accept...” e então clique em “Finish”:

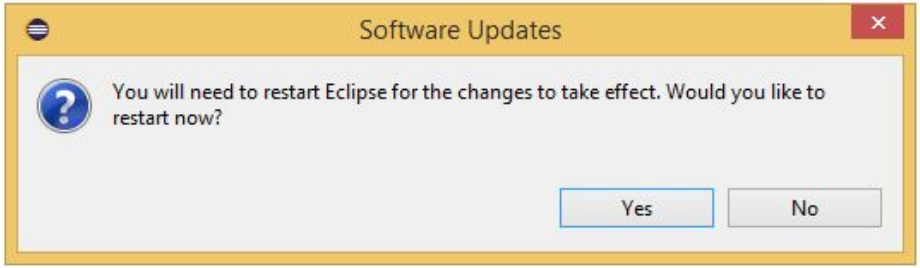


Aguarde o fim do download e da instalação (pode demorar um pouco):



Se aparecer a mensagem abaixo, clique em “OK”:

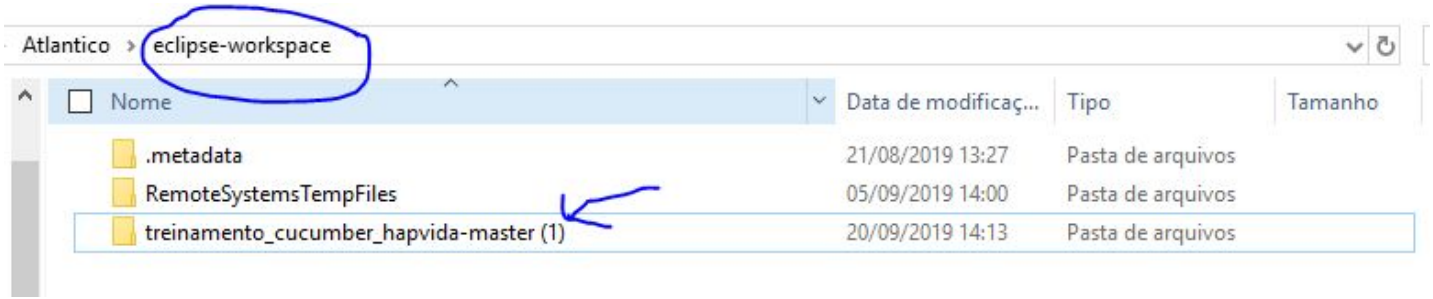
Quando aparecer a mensagem abaixo solicitando o reinicio do Eclipse, clique em “Yes”:



Importando projeto

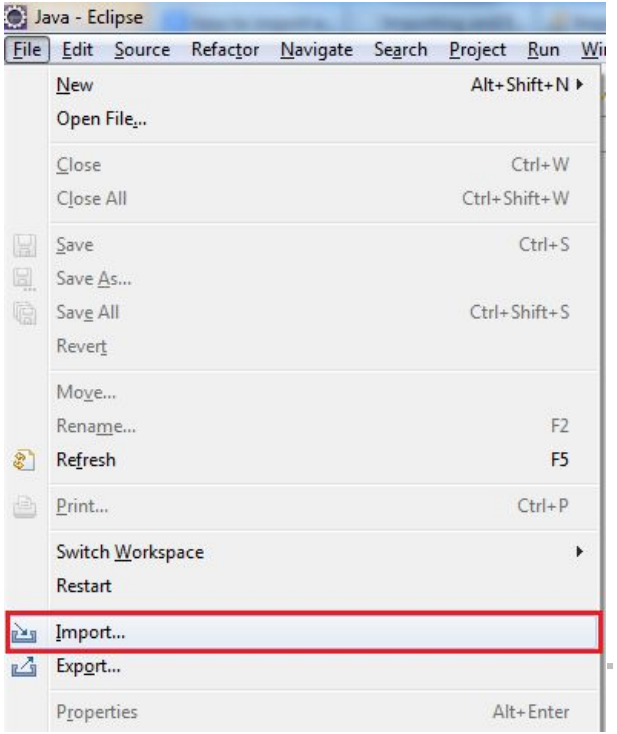
Github: https://github.com/mickhill/treinamento_cucumber_hapvida

Passo 1: Descompactar o projeto na Workspace que você irá usar.



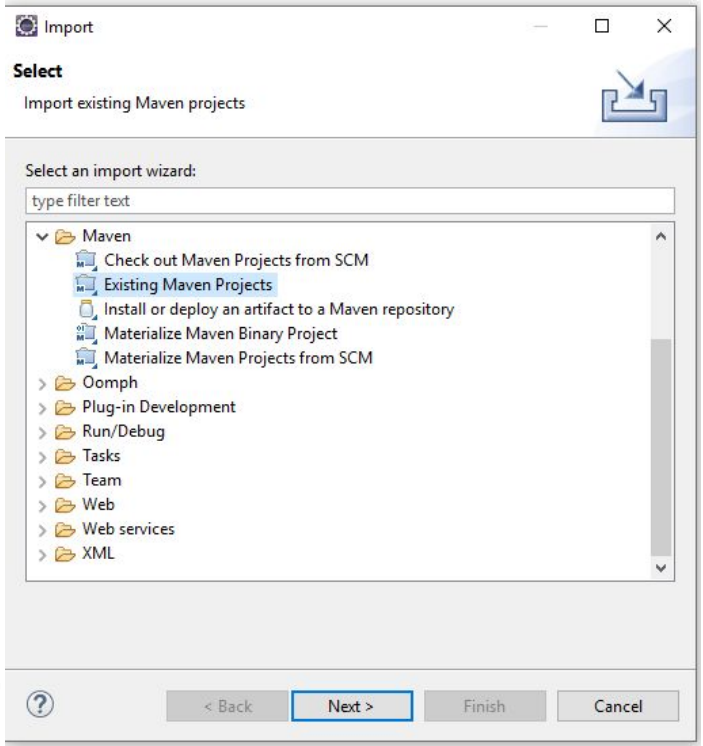
Importando projeto

Passo 2: Importar o projeto para o Eclipse.



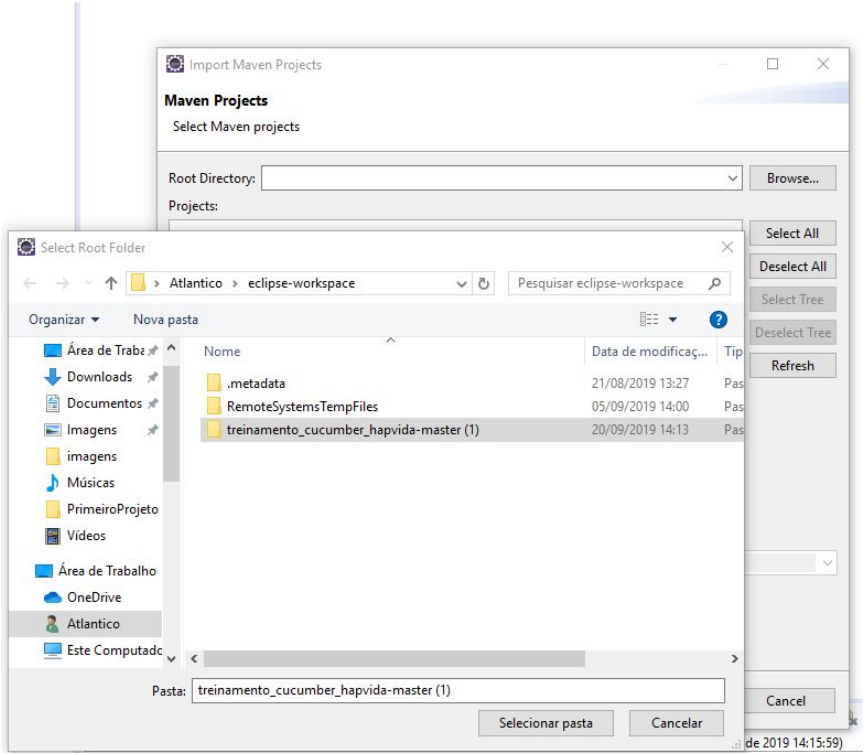
Importando projeto

Passo 3: Importar o projeto do tipo Maven existente.



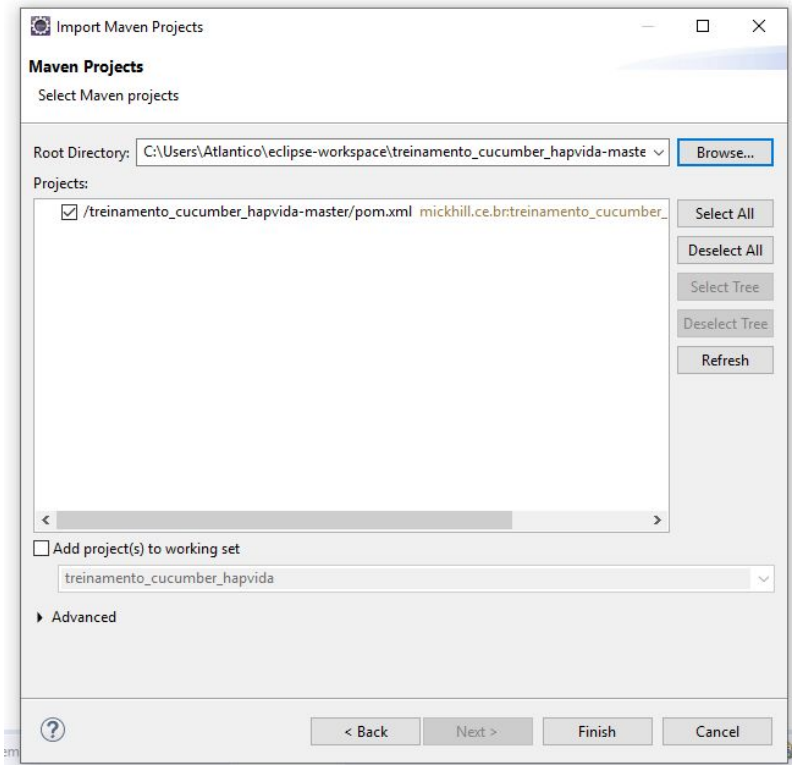
Importando projeto

Passo 4: Selecionar o local onde colocamos o projeto descompactado.

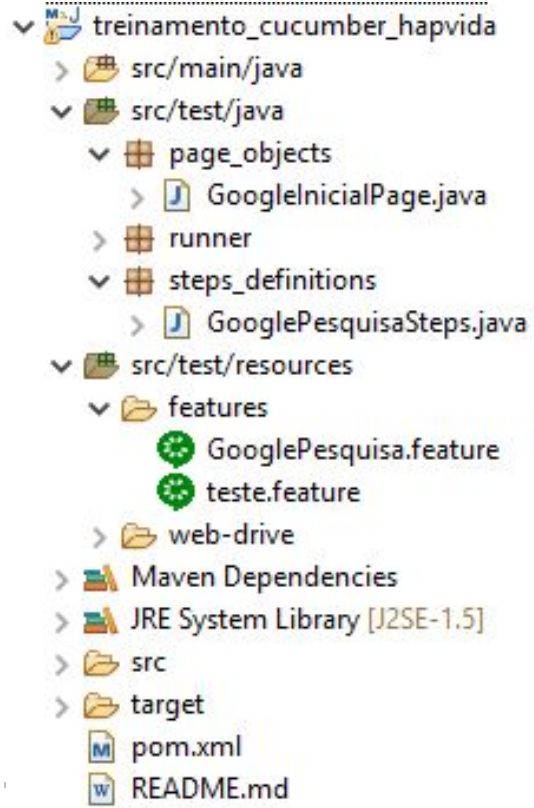


Importando projeto

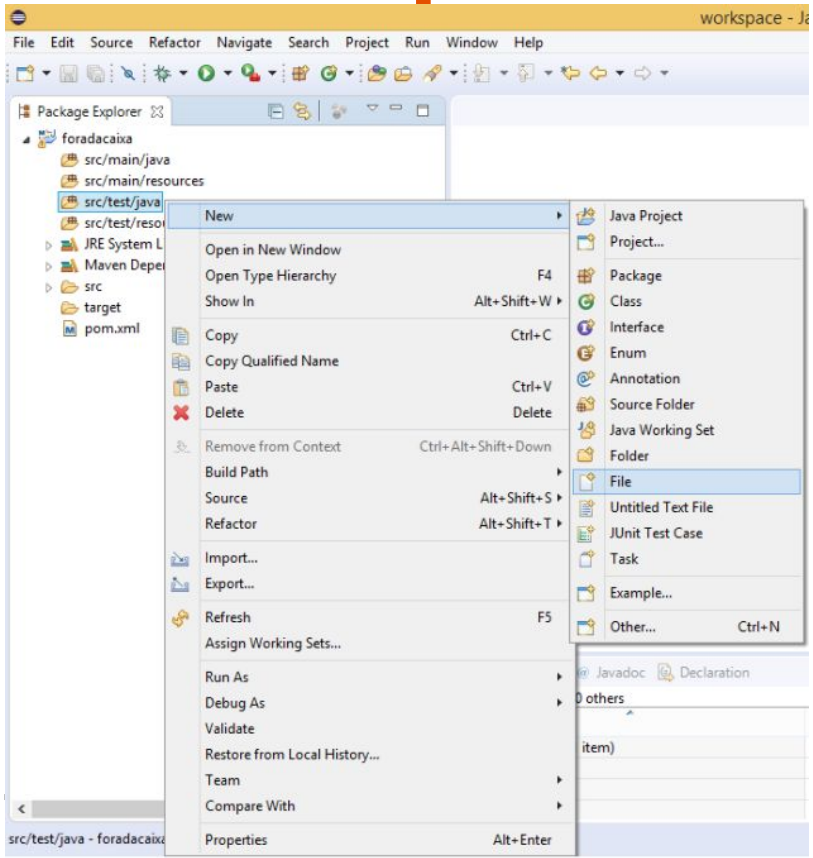
Passo 5: Verificar o endereço e finalizar a importação do projeto.



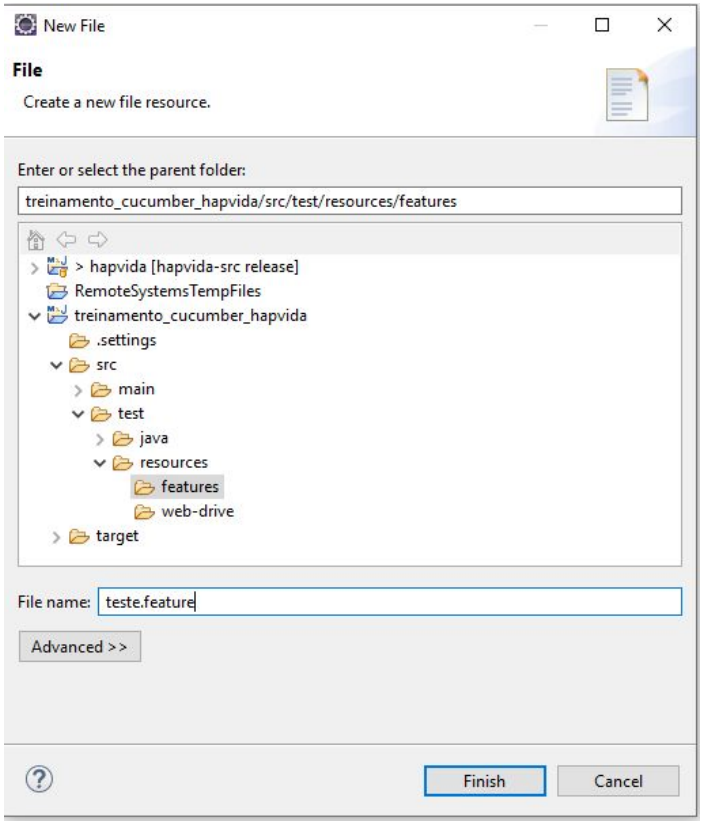
Estrutura do Projeto



Criando um arquivo .feature



Criando um arquivo .feature



Trabalhando com Cucumber

```
@tag
Feature: Title of your feature
  I want to use this template for my feature file

  @tag1
  Scenario: Title of your scenario
    Given I want to write a step with precondition
    And some other precondition
    When I complete action
    And some other action
    And yet another action
    Then I validate the outcomes
    And check more outcomes

  @tag2
  Scenario Outline: Title of your scenario outline
    Given I want to write a step with <name>
    When I check for the <value> in step
    Then I verify the <status> in step

    Examples:
      | name  | value | status  |
      | name1 |     5 | success |
      | name2 |     7 | Fail    |
```

```
#language: pt
Funcionalidade: Testar a Pesquisa do Google

  @pesquisaGoogle @pesquisaSuccess
  Cenario: Pesquisa Valida
    Dado que eu esteja na pagina inicial do google
    Quando eu pesquisar por um assunto
    Entao me retorna os resultados indexados

  @pesquisaGoogle @pesquisaUndefined
  Cenario: Pesquisa vazia
    Dado que eu esteja na pagina inicial do google
    Quando eu pesquisar sem preencher o assunto
    Entao continuarei na mesma pagian aguardando um assunto
```

Contexto (Gherkin)

Ao escrevermos usando um contexto, podemos adicionar passos que se repetem em todos os cenários, evitando acoplamento de responsabilidades e retrabalho caso ocorra alguma mudança na feature.

*teste.feature

```
#language: pt
Funcionalidade: Testando a ausencia da funcionalidade Contexto

  @CadastroClienteValido
  Cenario: Cadastrando um usuario valido
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos
    Quando informo os dados de um cliente valido para cadastro
    Entao o sistema deve cadastrar o cliente
    E exibir uma mensagem de confirmação.

      @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com CPF invalido
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos
    Quando informo os dados de um cliente com CPF invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.

    @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com Email invalido
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos
    Quando informo os dados de um cliente com Email invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.
```

*teste.feature

```
#language: pt
Funcionalidade: Testando a funcionalidade Contexto

Contexto:
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos


  @CadastroClienteValido
  Cenario: Cadastrando um usuario valido

    Quando informo os dados de um cliente valido para cadastro
    Entao o sistema deve cadastrar o cliente
    E exibir uma mensagem de confirmação.

      @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com CPF invalido
    Quando informo os dados de um cliente com CPF invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.

    @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com Email invalido
    Quando informo os dados de um cliente com Email invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.
```

Esquemas de Cenário (Scenario Outline)

Uma maneira de organizar esses cenários e mantê-los da melhor forma. Quando você usa, cria uma tabela de exemplos em que cada linha dela irá representar um cenário.

```
Cenario: Não usando esquema de cenario
Dado que o usuario tenha acesso o sistema
Quando informar meu usuario "sandra"
E informar a senha "123456"
Entao verificar se está "disponivel" no sistema

Cenario: Não usando esquema de cenario
Dado que o usuario tenha acesso o sistema
Quando informar meu usuario "teste"
E informar a senha "654321"
Entao verificar se está "indisponivel" no sistema
```

```
Esquema: Usando esquema de cenario
Dado que o usuario tenha acesso o sistema
Quando informar <meu usuario>
E informar a senha <minha senha>
Entao verificar o <status> no sistema

Exemplos:
| meu usuario | minha senha | status |
| sandra      | 123456      | disponivel |
| teste       | 654321      | indisponivel |
```

Faça uso de E (And) e Mas (But)

Os And e But que você venha a alterar terão o mesmo comportamento dos Given e When repetidos. Se você perceber quando for convertê-los para **steps-definitions**, eles terão a mesma nomenclatura, mas a descrição da sua funcionalidade será mais fácil de ler e entender.

```
Contexto:
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos


  @CadastroClienteValido
  Cenario: Cadastrando um usuario valido

    Quando informo os dados de um cliente valido para cadastro
    Entao o sistema deve cadastrar o cliente
    Entao exibir uma mensagem de confirmação.

    @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com CPF invalido
    Quando informo os dados de um cliente com CPF invalido
    Entao o sistema não deve cadastrra o cliente
    Entao exibir uma mensagem de erro.

    @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com Email invalido
    Quando informo os dados de um cliente com Email invalido
    Entao o sistema não deve cadastrra o cliente
    Entao exibir uma mensagem de erro.
```

*teste.feature

```
#language: pt
Funcionalidade: Testando a funcionalidade Contexto

Contexto:
    Dado que eu acesse a tela de login do sistema
    E informe meu login e senha de acesso validos


  @CadastroClienteValido
  Cenario: Cadastrando um usuario valido

    Quando informo os dados de um cliente valido para cadastro
    Entao o sistema deve cadastrar o cliente
    E exibir uma mensagem de confirmação.

      @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com CPF invalido
    Quando informo os dados de um cliente com CPF invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.

    @CadastroClienteCPFInvalido
    Cenario: Cadastrando um usuario com Email invalido
    Quando informo os dados de um cliente com Email invalido
    Entao o sistema não deve cadastrra o cliente
    E exibir uma mensagem de erro.
```

Features viram steps-definitions

src/test/resources/features/teste.feature - Eclipse IDE

Help

1 treinamento_cucumber_hapvida
2 hapvida
Run As
Run Configurations...
Organize Favorites...
1 Cucumber Feature

```
1  #langu
2  Funcio
3
4  Contex
5      Da                                  stema
6      E                                   lidos
7
8    @CadastroClienteValido
9    Cenario: Cadastrando um usuario valido
10
11     Quando informo os dados de um cliente valido para cadastro
12     Entao o sistema deve cadastrar o cliente
13     Entao exibir uma mensagem de confirmação.
14
15     @CadastroClienteCPFInvalido
16     Cenario: Cadastrando um usuario com CPF invalido
17     Quando informo os dados de um cliente com CPF invalido
18     Entao o sistema não deve cadastrra o cliente
19     Entao exibir uma mensagem de erro.
20
21     @CadastroClienteCPFInvalido
22     Cenario: Cadastrando um usuario com Email invalido
23     Quando informo os dados de um cliente com Email invalido
24     Entao o sistema não deve cadastrra o cliente
25     Entao exibir uma mensagem de erro.
26
```

teste.feature

```
1  #language: pt
2  Funcionalidade: Testando a funcionalidade Contexto
3
4  Contexto:
5      Dado que eu acesse a tela de login do sistema
6      E informe meu login e senha de acesso validos
7
8    @CadastroClienteValido
9    Cenario: Cadastrando um usuario valido
10
11     Quando informo os dados de um cliente valido para cadastro
```

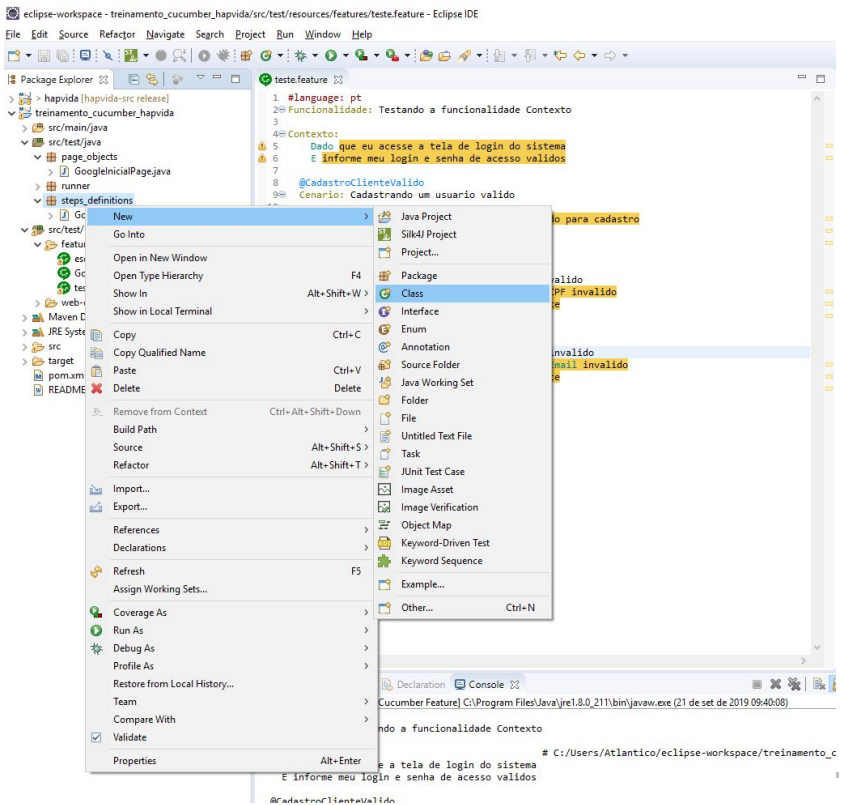
Problems Javadoc Declaration Console

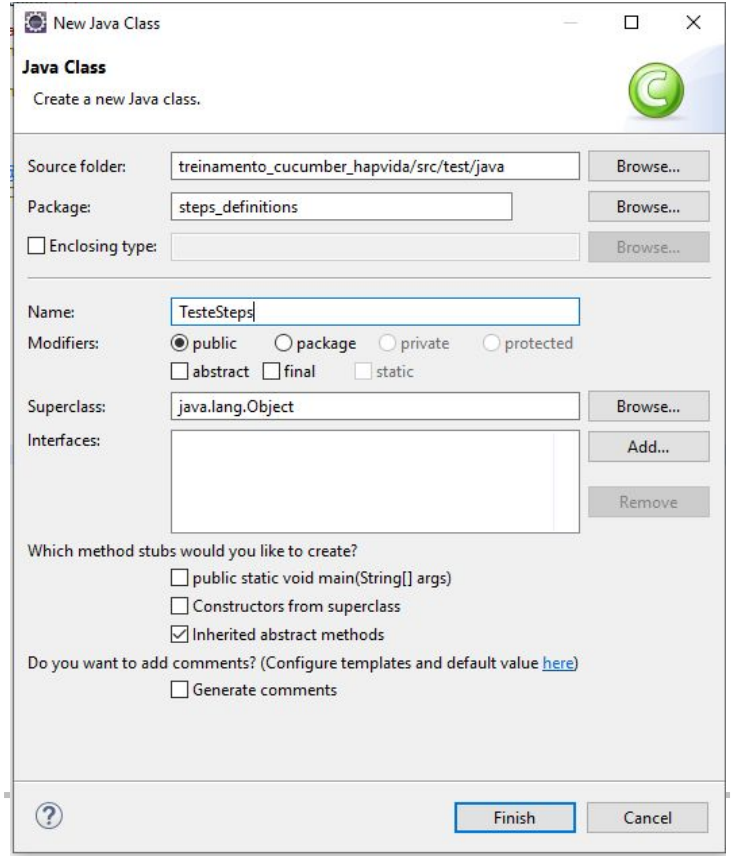
<terminated> teste.feature [Cucumber Feature] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (21 de set de 2019 09:40:08)

You can implement missing steps with the snippets below:

```
@Dado("^que eu acesse a tela de login do sistema$")
public void que_eu_acesse_a_tela_de_login_do_sistema() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Dado("^informe meu login e senha de acesso validos$")
public void informe_meu_login_e_senha_de_acesso_validos() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Quando("^informo os dados de um cliente valido para cadastro$")
public void informo_os_dados_de_um_cliente_valido_para_cadastro() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^o sistema deve cadastrar o cliente$")
public void o_sistema_deve_cadastrar_o_cliente() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^exibir uma mensagem de confirmação\\.$")
public void exibir_uma_mensagem_de_confirmação() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Quando("^informo os dados de um cliente com CPF invalido$")
public void informo_os_dados_de_um_cliente_com_CPF_invalido() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^o sistema não deve cadastrra o cliente$")
public void o_sistema_não_deve_cadastrra_o_cliente() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
```

Criar classe para os Steps





Classe steps com estória de usuário.

teste.feature | TesteSteps.java

```java
package steps_definitions;

import cucumber.api.java.es.Dado;
import cucumber.api.java.it.Quando;
import cucumber.api.java.pt.Entao;

public class TesteSteps {

@Dado("^que eu acesse a tela de login do sistema$")
public void que_eu_acesse_a_tela_de_login_do_sistema() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Dado("^informe meu login e senha de acesso validos$")
public void informe_meu_login_e_senha_de_acesso_validos() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Quando("^informo os dados de um cliente valido para cadastro$")
public void informo_os_dados_de_um_cliente_valido_para_cadastro() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^o sistema deve cadastrar o cliente$")
public void o_sistema_deve_cadastrar_o_cliente() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^exibir uma mensagem de confirmação\\.$")
public void exibir_uma_mensagem_de_confirmação() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Quando("^informo os dados de um cliente com CPF invalido$")
public void informo_os_dados_de_um_cliente_com_CPF_invalido() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Entao("^o sistema não deve cadastrra o cliente$")
public void o_sistema_não_deve_cadastrra_o_cliente() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
```

HORA DE:

Criação de Cenários de Teste

RF 3: Exclusão do Cadastro de Usuário Descrição: O moderador do site poderá excluir o cadastro dos usuários. Entrada: Nome de usuário Processo: O sistema verifica se o usuário é cadastrado, se for o usuário é excluído. Saída: Mensagem de confirmação bem sucedido da exclusão do cadastro caso tenha sido efetuado com sucesso, senão, mensagem de erro.

RF. 4: Inserção de Documentos Descrição: Os usuários cadastrados podem inserir documentos com suas descrições. Entrada: Autor(es), título, palavras-chaves, resumo, local de aplicação, upload dos documentos. Processo: O sistema insere todos esses dados no no banco de dados.

Criação de Cenários de Teste

RF.5: Modificação de Documento Descrição: O usuário pode fazer alguma alteração na descrição do documento. Entrada: Campo desejado e o novo dado. Processo: Atualização da descrição do documento no banco de dados. Saída: Mensagem de confirmação bem sucedido da modificação caso tenha sido efetuado com sucesso, senão, mensagem de erro.

RF. 6: Exclusão de Documento Descrição: O moderador pode efetuar a exclusão de documentos. Entrada: Título Processo: O sistema busca o título no banco de dados, caso ele encontre ele exclui o documento. Saída: Mensagem de confirmação bem sucedido da exclusão do documento caso tenha sido efetuado com sucesso, senão, mensagem de erro.

Um pouco de automação.

GooglePesquisa.feature

```
1  #language: pt
2  Funcionalidade: Testar a Pesquisa do Google
3
4    @pesquisaGoogle @pesquisaSuccess
5    Cenario: Pesquisa Valida
6      Dado que eu esteja na pagina inicial do google
7      Quando eu pesquisar por um assunto
8      Entao me retorna os resultados indexados
9
10   @pesquisaGoogle @pesquisaUndefined
11   Cenario: Pesquisa vazia
12     Dado que eu esteja na pagina inicial do google
13     Quando eu pesquisar sem preencher o assunto
14     Entao continuarei na mesma pagian aguardando um assunto
```

Um pouco de automação.

GoogleInicialPage.java

```
6  import org.openqa.selenium.support.ui.WebDriverWait;
7
8  public class GoogleInicialPage
9  {
10     private WebDriver browser;
11     private WebDriverWait wait;
12     private String url          = "https://www.google.com.br/";
13     private By inputPesrquisar  = By.xpath("//input[@name='q']");
14     private By btnPesquisar     = By.xpath("(//input[@name='btnK'])[2]");
15     private By txtResultado     = By.id("resultStats");
16
17     public GoogleInicialPage()
18     {
19         super();
20
21         System.setProperty("webdriver.chrome.driver", "src\\test\\resources\\web-drive\\chromedriver.exe");
22         browser = new ChromeDriver();
23         wait    = new WebDriverWait(browser, 9999);
24     }
25
26     public void abrirPagina()
27     {
28         browser.get(url);
29     }
30
31     public void preencherFormPesquisa(String pesquisa)
32     {
33         wait.until(ExpectedConditions.visibilityOfElementLocated(inputPesrquisar));
34         browser.findElement(inputPesrquisar).sendKeys(pesquisa);
35     }
36
37     public void pesquisar()
38     {
39         wait.until(ExpectedConditions.visibilityOfElementLocated(btnPesquisar));
40         browser.findElement(btnPesquisar).click();
41     }
42
43     public String verResultadoPesquisa()
44     {
45         wait.until(ExpectedConditions.visibilityOfElementLocated(txtResultado));
46         return browser.findElement(txtResultado).getText().substring(0,15);
47     }
48
49     public Boolean conferirSeEstouNaPaginaInicial()
50     {
51         String paginaAtual = browser.getCurrentUrl();
52         return url.equals(paginaAtual);
53     }
54
55     public void fecharPagina()
56     {
57         browser.quit();
```

Um pouco de automação.

*GooglePesquisaSteps.java

```java
package steps_definitions;
import static org.junit.Assert.assertEquals;

public class GooglePesquisaSteps
{
    GoogleInicialPage pagina = new GoogleInicialPage();

    @Dado("^que eu esteja na pagina inicial do google$")
    public void queEuEstejaNaPaginaInicialDoGoogle()
    {
        pagina.abrirPagina();
    }
    /**
     * pesquisaSuccess
     **/
    @Quando("^eu pesquisar por um assunto$")
    public void euPesquisarPorUmAssunto()
    {
        pagina.preencherFormPesquisa("Teste Automatizado");
        pagina.pesquisar();
    }

    @Entao("^me retorna os resultados indexados$")
    public void meRetornaOsResultadosIndexados()
    {
        String txtAssertPage = pagina.verResultadoPesquisa();
        pagina.fecharPagina();

        assertEquals("Aproximadamente", txtAssertPage);
    }

    /**
     * pesquisaUndefined
     **/
    @Quando("^eu pesquisar sem preencher o assunto$")
    public void euPesquisarSemPreencherOAssunto()
    {
        pagina.pesquisar();
    }

    @Entao("^continuarei na mesma pagian aguardando um assunto$")
    public void continuareiNaMesmaPagianAguardandoUmAssunto()
    {
        Boolean mesmaPagina = pagina.conferirSeEstouNaPaginaInicial();
        pagina.fecharPagina();

        assertEquals(true, mesmaPagina);
    }
}
```

OBRIGADO!

TIME Hapvida!

hapvida
Faz bem pra você