

## Creación del Feature, StepDefinitions y Runner

### Tabla de Contenido

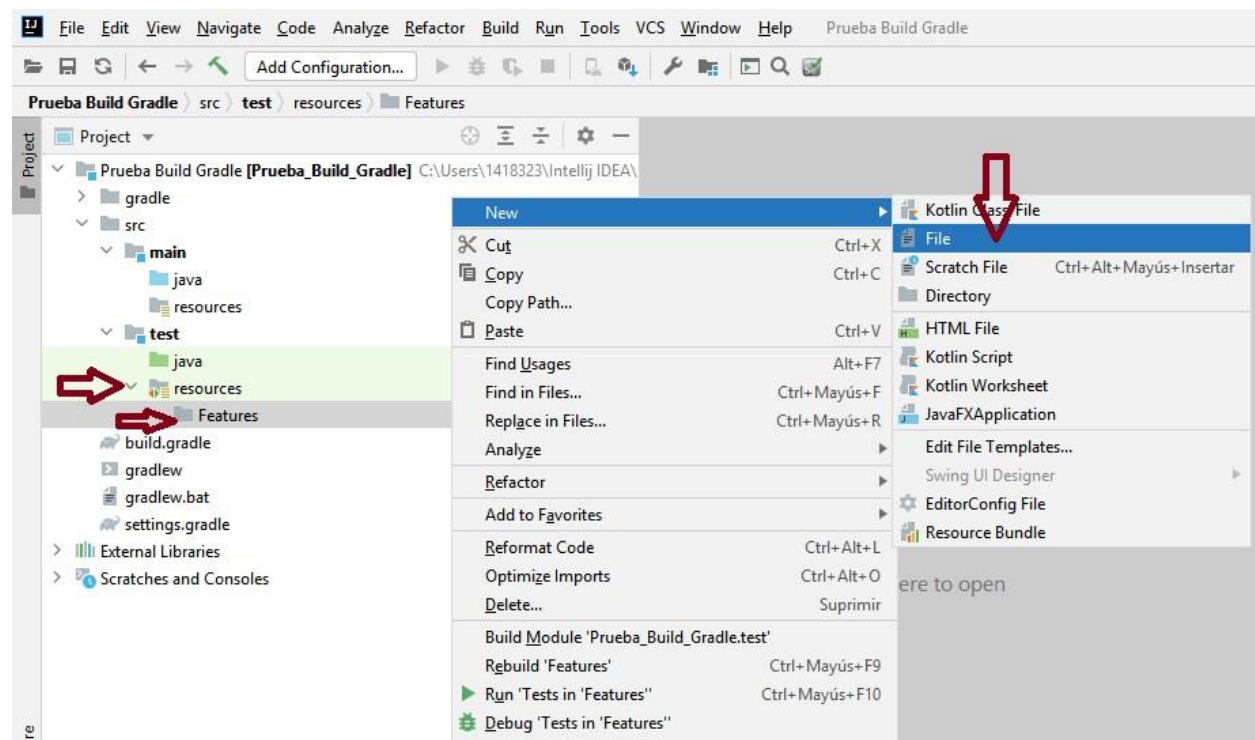
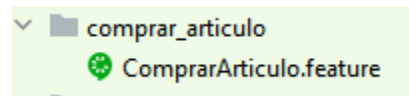
#### Contents

Creación de archivo feature.....	2
Creación del StepDefinitions.....	6
Creación del Runner .....	7
Configuración del Runner .....	8
Ejecución del Runner .....	9

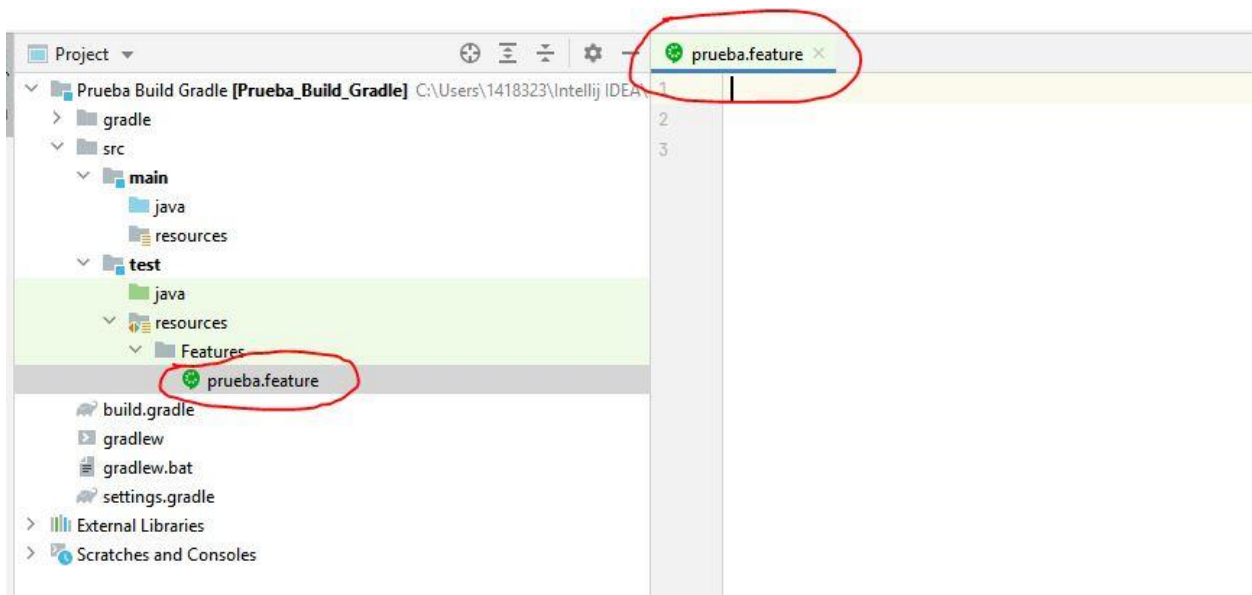
## Creación de archivo feature

1. Dirigirse a la segunda carpeta resources, crear una carpeta llamada Features dando click derecho y seleccionar Directorio o Directory
2. Dentro de la carpeta creada, crear el archivo feature dando click derecho, luego New y seleccionar File. Aca vamos a indicar el nombre del archivo adicionando la extensión .feature

Ejemplo:



Se abre un archivo en blanco para comenzar a configurar el Feature



3. Vamos a configurar el lenguaje en el que queremos describir el feature, por defecto viene con el idioma inglés. Se recomienda que sea en el mismo lenguaje que el Usuario Product Owner o grupo de trabajo lo entienda, es decir hacerlo todo en inglés o todo en español, etc.

641

Feature: Validar requisitos de solicitud de crédito de Libre Inversion

Yo como usuario requiero de una aplicacion que permita evaluar la solicitud de Crédito de Libre Inversión realizada por una persona Natural para verificar su aprobación o no de acuerdo a las condiciones del Banco y otorgar el crédito

Scenario: Validar Segmento Persona

Given que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion

When se consulta el segmento al que pertenece

Then se valida que sea de Segmento Personas

Scenario Outline: Validar ingresos

Given que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion

When se consulta el ingreso "<ingreso>"

Then se valida que sea superior a \$1000000

Examples: |

| ingreso |

| 2000000 |

Configuramos el lenguaje con **"# language: es"** por lo que ya la sentencias nos van a salir en español

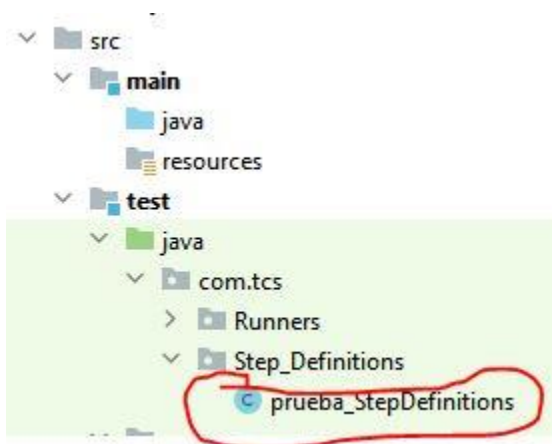
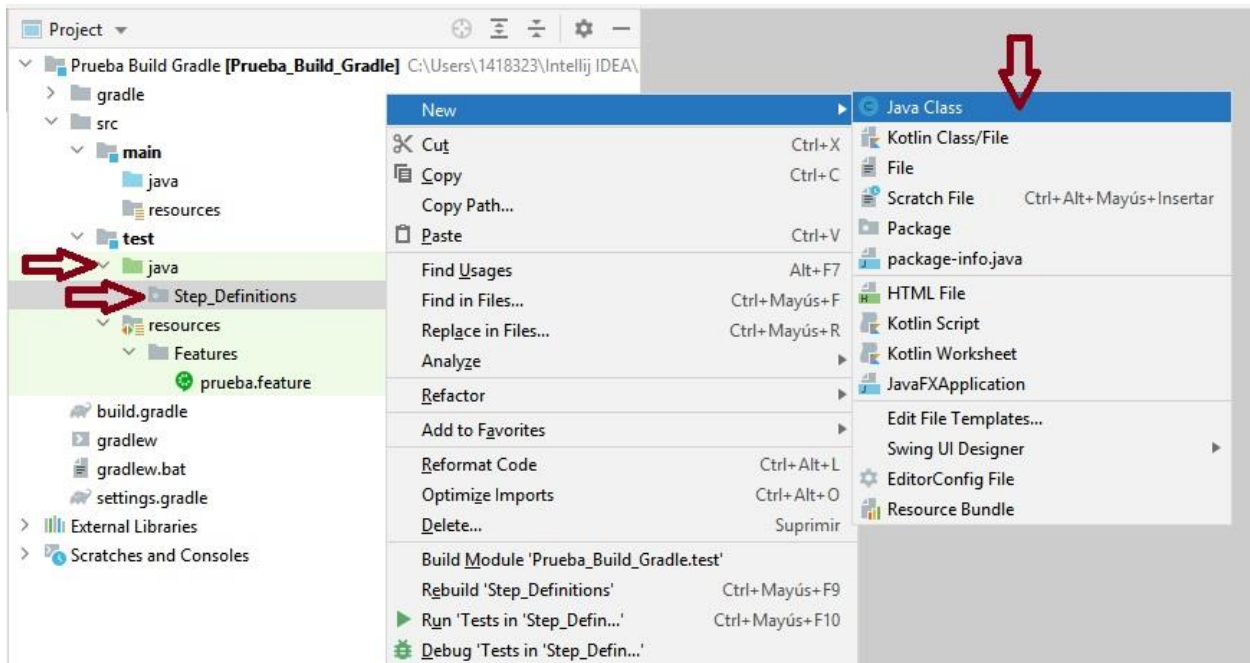
```
prueba.feature x
1  #language:es
2
3  >> Característica: Validar requisitos de solicitud de crédito de Libre Inversion
4
5      Yo como usuario requiero de una aplicacion que permita evaluar la solicitud de Crédito
6      de Libre Inversión realizada por una persona Natural para verificar su aprobación o no
7      de acuerdo a las condiciones del Banco y otorgar el crédito
8
9  >> Escenario: Validar Segmento Persona
10     Dado que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion
11     Cuando se consulta el segmento al que pertenece
12     Entonces se valida que sea de Segmento Personas
13
14  >> Esquema del escenario: Validar ingresos
15     Dado que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion
16     Cuando se consulta el ingreso "<ingreso>"
17     Entonces se valida que sea superior a $1000000
18
19     Ejemplos:
20     | ingreso |
21     | 2000000 |
22
```

**Nota:** las sentencias van a quedar resaltado de color naranja como se observa en la imagen, debido a que aún no se ha implementado los StepDefinitios para el feature creado.

## Creación del StepDefinitions

1. Debajo de la carpeta Java contenida en la carpeta Test, crear una paquete llamado **com.tcs.Step\_Definitios**, luego dando click derecho en el paquete creado, seleccionar opción new . Java Class y poner el nombre del StepDefinitios de la siguiente manera:

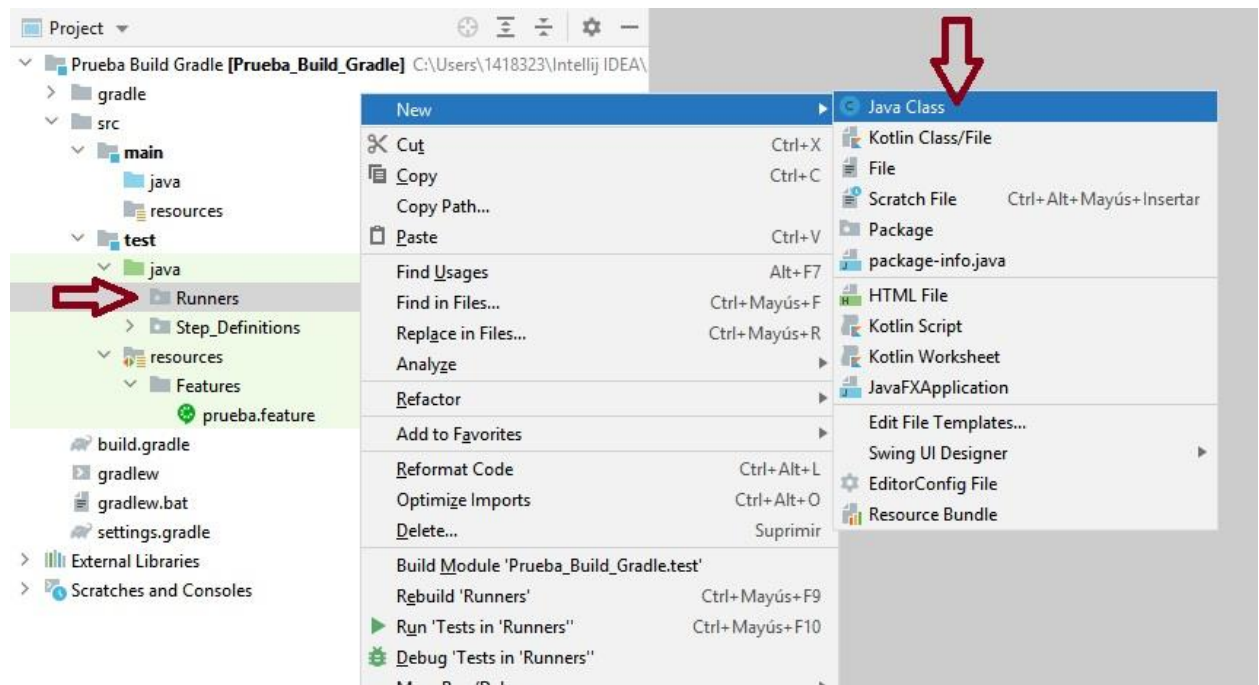
**nombre\_StepDefinitions**



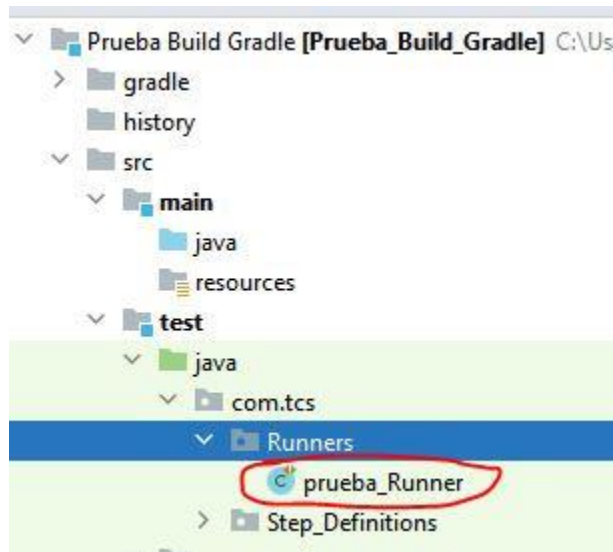
## Creación del Runner

1. Debajo de la carpeta Java contenida en la carpeta Test, crear un paquete llamado **com.tcs.Runners**, luego dando click derecho en el paquete creado, seleccionar opción new, Java Class y poner el nombre del Runner de la siguiente manera:

**nombre\_Runner**

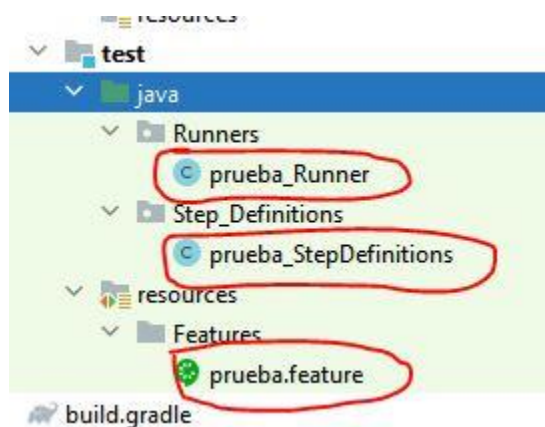






**Nota:** Recuerde que debe de haber por cada archivo .feature , 1 clase de StepDefinitions y 1 Runner, todos con el mismo nombre.

Ejemplo:



## Configuración del Runner

En la clase creada del runner (en este caso prueba\_Runner) , ingresar lo siguiente encima de public class:

```
@RunWith(CucumberWithSerenity.class)
@CucumberOptions(features = "src/test/resources/Features/prueba.feature",
//Aca se indica la ruta donde se encuentra el Feature a ejecutar
snippets = SnippetType.CAMELCASE, // se indica que ejecute en formato
Mayuscula - Minuscula ej: ClasePrueba
// tags="@Prueba", /// se indica el tag del escenario que se quiere
probar especificamente
```

Elaborado por: Lina Marcela Franco Montes  
 Analista de Certificación  
[Lina.francomontes@tcs.com](mailto:Lina.francomontes@tcs.com)



```
glue = "com.tcs.stepDefinitions") // aca se indica la ruta donde se encuentra la carpeta de StepFefinitios.
```

**Nota:** va a aparecer sentencias en Rojo como por ejemplo RunWith, por lo tanto, debe ubicar el mouse ahí para que le indique importar clase.

Quedaría el runner así:

```
package com.tcs.Runners;

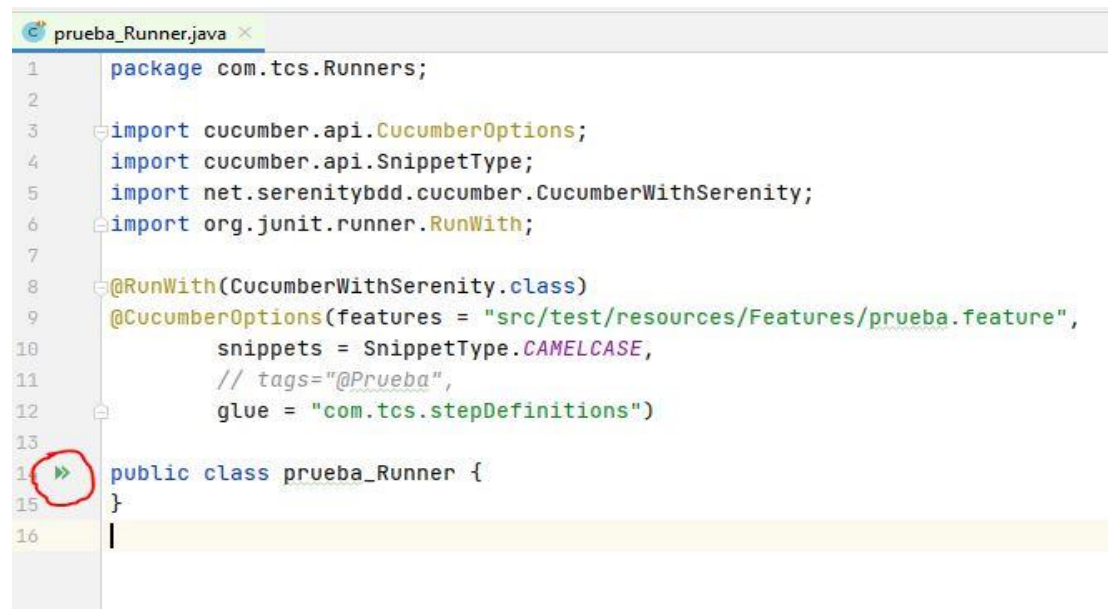
import cucumber.api.CucumberOptions;
import cucumber.api.SnippetType;
import net.serenitybdd.cucumber.CucumberWithSerenity;
import org.junit.runner.RunWith;

@RunWith(CucumberWithSerenity.class)
@CucumberOptions(features = "src/test/resources/Features/prueba.feature",
    snippets = SnippetType.CAMEL_CASE,
    // tags="@Prueba",
    glue = "com.tcs.stepDefinitions")

public class prueba_Runner {
}
```

## Ejecución del Runner

1. Para ejecutar el Runner, se presiona de la flecha ubicada al lado izquierdo



2. Como aun el Stepdefinitions está vacío, por lo tanto, al ejecutar el Runner nos debe de aparecer los test pendientes que se encuentran resaltados de color Naranja en el feature.

```
prueba.feature x
1  #language:es
2
3  >> Característica: Validar requisitos de solicitud de crédito de Libre Inversion
4
5  Yo como usuario requiero de una aplicacion que permita evaluar la solicitud de Crédito
6  de Libre Inversión realizada por una persona Natural para verificar su aprobación o no
7  de acuerdo a las condiciones del Banco y otorgar el crédito
8
9  >> Escenario: Validar Segmento Persona
10  Dado que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion
11  Cuando se consulta el segmento al que pertenece
12  Entonces se valida que sea de Segmento Personas
13
14  >> Esquema del escenario: Validar ingresos
15  Dado que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion
16  Cuando se consulta el ingreso "<ingreso>"
17  Entonces se valida que sea superior a $1000000
18
19  Ejemplos:
20  | ingreso |
21  | 2000000 |
22
```

Nos debe de aparecer lo siguiente:

TEST PENDING: Validar ingresos

---

2 Scenarios (2 undefined)

6 Steps (6 undefined)

0m0,516s

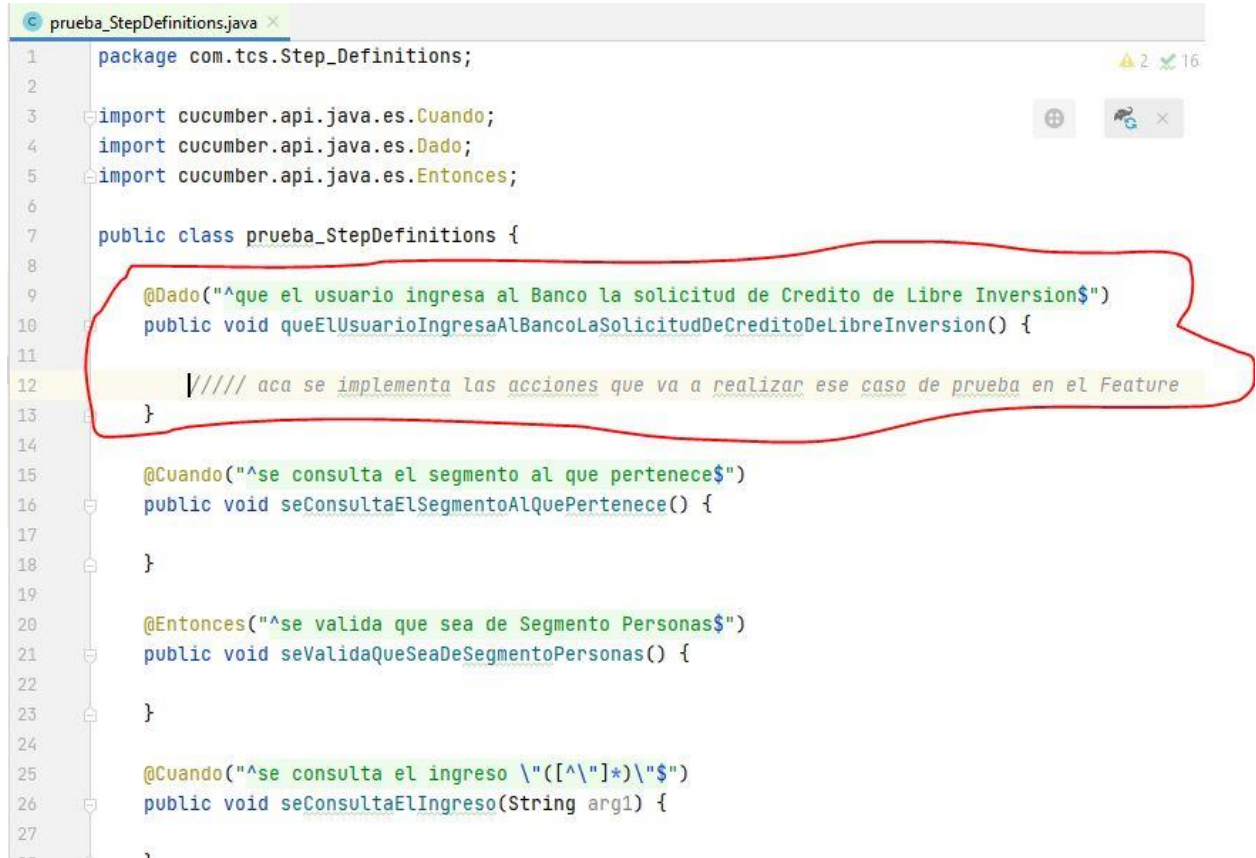
You can implement missing steps with the snippets below:

```
@Dado("^que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion$")
public void queElUsuarioIngresaAlBancoLaSolicitudDeCreditoDeLibreInversion() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
```

```
@Cuando("^se consulta el segmento al que pertenece$")
public void seConsultaElSegmentoAlQuePertenece() {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
```

```
@Entonces("^se valida que sea de Segmento Personas$")
public void seValidaQueSeaDeSegmentoPersonas() {
```

3. Copiar todos los escenarios que aún están pendientes y pegarlo en el stepDefinitions.



```
1 package com.tcs.Step_Definitions;
2
3 import cucumber.api.java.es.Cuando;
4 import cucumber.api.java.es.Dado;
5 import cucumber.api.java.es.Entonces;
6
7 public class prueba_StepDefinitions {
8     @Dado("^que el usuario ingresa al Banco la solicitud de Credito de Libre Inversion$")
9     public void queElUsuarioIngresaAlBancoLaSolicitudDeCreditoDeLibreInversion() {
10
11         ///// aca se implementa las acciones que va a realizar ese caso de prueba en el Feature
12     }
13
14     @Cuando("^se consulta el segmento al que pertenece$")
15     public void seConsultaElSegmentoAlQuePertenece() {
16
17     }
18
19     @Entonces("^se valida que sea de Segmento Personas$")
20     public void seValidaQueSeaDeSegmentoPersonas() {
21
22     }
23
24     @Cuando("^se consulta el ingreso \"([^\"]*)\"$")
25     public void seConsultaElIngreso(String arg1) {
26
27     }
28 }
```

**Nota:** va a aparecer sentencias en Rojo como por ejemplo @Dado, por lo tanto, debe ubicar el mouse ahí para que le indique importar clase.