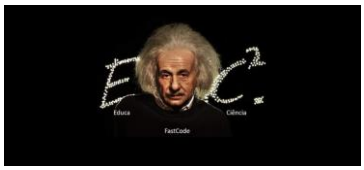


EducaCiência FastCode

Fala Galera,

Neste artigo, abordaremos um tema muito interessante.

- Artigo: 67/2022 Data: Fevereiro/2022
- Público-alvo: Desenvolvedores – Iniciantes ao Avançado
- Linguagem: Java
- Tema: Artigo 67 – ODM_Java Valida Promocao 3
- Link: https://github.com/perucello/Artigos-EducaCiencia_FastCode

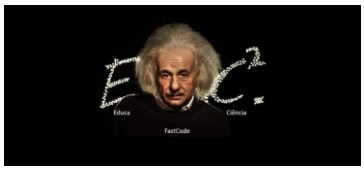
Desta vez , escolhi um tema interessante, vamos falar de ODM ou Operational Decision Management.
Antes de iniciarmos nosso projeto, vamos dar uma pequena explicação sobre o ODM, então vamos lá.

- *O IBM Operational Decision Manager é uma solução de automação de decisão abrangente no pacote de soluções Digital Business Automation , onde o ODM fornece recursos de automação de decisão abrangentes que ajudam a descobrir, capturar, analisar, automatizar e controlar decisões de negócios baseadas em regras*.

O ODM foi projetado para usuários de tecnologia da informação e de negócios, como exemplos de aplicações práticas, a solução pode autorizar um empréstimo, decidir sobre ofertas promocionais ou detectar uma oportunidade de venda cruzada com alta precisão.
A customização IBM Operational Decision Manager auxilia aplicativos críticos a permanecer atualizados e alinhados com os objetivos de negócios em constante mudança.

Como é " Com ODM ":

• **Arquitetura ODM** - desenha a solução para garantir a performance na execução das regras e usabilidade do Decision Management, supervisiona equipes de desenvolvimento.
• **Engenharia ODM Desenvolvedor de Integração** - atua na Integração das aplicações com ODM usando Console de Negócio, envolvido na validação das mudanças de contratos, desenvolve componentes que integram com outros sistemas (Web Services, Dados, HTTP etc).
• **Desenvolvedor da Regra** - desenha estrutura para suportar as regras e evolução dos Serviços de Decisão.
• **Administrador do Decision Center** - Cria e configura Projetos, delega acessos.
• **Gerente de Segurança Implementa** - segurança nos Serviços de Decisão, configura permissões e associa usuários aos grupos de acesso.
• **Administrador do Decision Server** - Administra o DS após subir para produção.
• **Owner de Publicação / Deploy** - define planos de publicação e implementação (Rule Designer, Regras de empacotamento / RulleApp, Regras de Versionamento RulleApp.



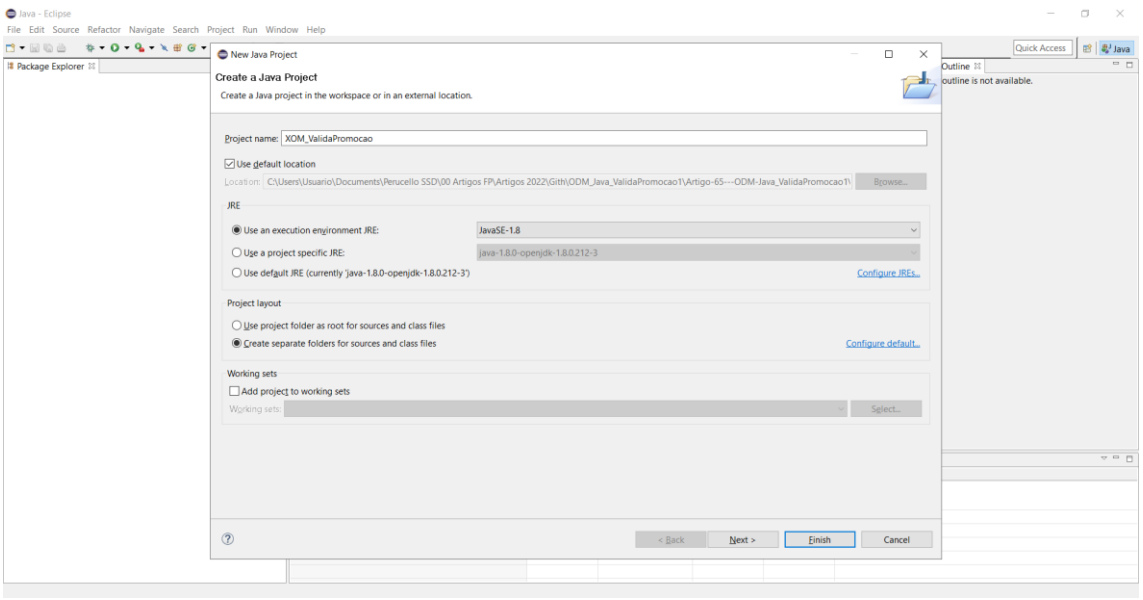
Dito isso, vamos iniciar nosso projeto.

No nosso cenário, iremos criar um ODM que simula uma promoção, ou seja, no cenário imaginemos que temos um comércio eletrônico, que possui nosso banco de dados com nossos produtos e como temos anualmente diversos eventos que faz com que coloquemos nossos produtos em promoção. Vamos simular um ODM onde terá como regra validar se nosso produto encontra em promoção ou não, em caso positivo ele irá informar o "novo" valor deste produto.

Lembrando que os fins são didáticos e iremos validar nosso ODM em seu ambiente de Desenvolvimento, ou seja, nosso Rule Designer ou Eclipse Luna que é a IDE que iremos usar no nosso desenvolvimento.

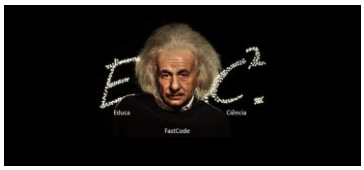
Nosso ODM receberá as regras de negócio em que validará se determinado produto está ou não em promoção. Se sim apresenta, o desconto, se não estiver, apresenta uma mensagem de que o produto não se encontra em promoção.

Então vamos iniciar nosso projeto, iremos iniciar criando nossa CAMADA XOM, ou seja, nosso código Java.

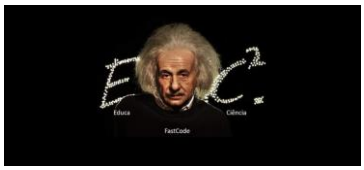


Iremos criar nesse momento duas classes, onde uma será nossa **Request** cujo qual receberá nossa requisição e a outra que será nossa **Response** , ou seja , que emitirá nossa resposta.

- **RequestValidaPromocao** – nesta classe teremos 3 atributos sendo *nomeProduto* e *descricaoProduto* e *fabricanteProduto*.
- **ResponseValidaPromocao** – nessa classe, retornaremos o *valorProduto* caso esteja em promoção.
- **ValidaPromocao** – nessa classe validaremos se o produto se encontra em promoção ou não. Caso esteja em promoção, ele irá validar o valor do produto, em caso negativo, retornaremos que "false" na nossa chamada que significará que nosso produto não está em promoção.
  Teremos também a classe em que servira de validação para nossa Regra.

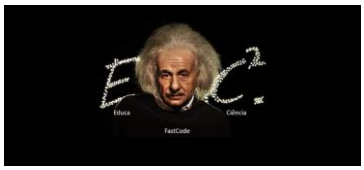


```
package com.educaciencia.odm.request;

public class RequestValidaPromocao {

	private String nomeProduto;
	private String descricaoProduto;
	private String fabricanteProduto;

	public RequestValidaPromocao() {
		super();
	}

	public RequestValidaPromocao(String nomeProduto, String
descricaoProduto,
			String fabricanteProduto) {
		super();
		this.nomeProduto = nomeProduto;
		this.descricaoProduto = descricaoProduto;
		this.fabricanteProduto = fabricanteProduto;
	}

	public String getNomeProduto() {
		return nomeProduto;
	}

	public void setNomeProduto(String nomeProduto) {
		this.nomeProduto = nomeProduto;
	}

	public String getDescricaoProduto() {
		return descricaoProduto;
	}

	public void setDescricaoProduto(String descricaoProduto) {
		this.descricaoProduto = descricaoProduto;
	}

	public String getFabricanteProduto() {
		return fabricanteProduto;
	}

	public void setFabricanteProduto(String fabricanteProduto) {
		this.fabricanteProduto = fabricanteProduto;
	}

}
```

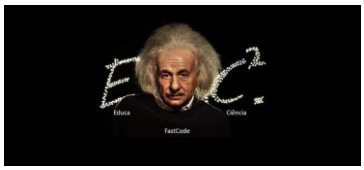


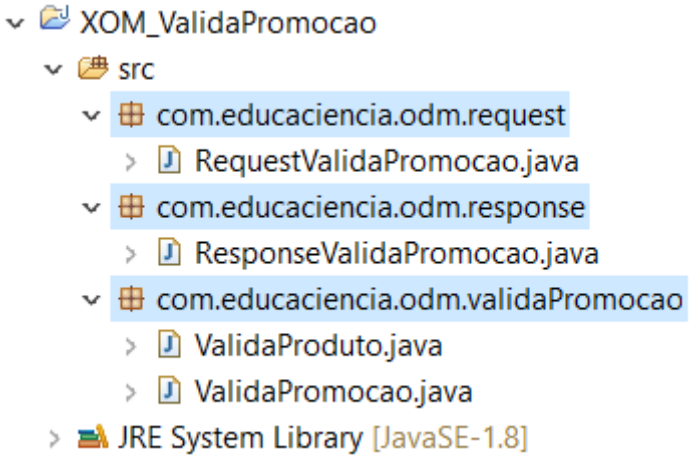
```java
package com.educaciencia.odm.response;

public class ResponseValidaPromocao {

	private String valorProduto;

	public ResponseValidaPromocao() {
		super();
	}

	public ResponseValidaPromocao(String valorProduto) {
		super();
		this.valorProduto = valorProduto;
	}

	public String getValorProduto() {
		return valorProduto;
	}

	public void setValorProduto(String valorProduto) {
		this.valorProduto = valorProduto;
	}

}
```

```java
package com.educaciencia.odm.validaPromocao;

public class ValidaPromocao {

	private boolean validaPromocao;

	public ValidaPromocao() {
		super();
	}

	public ValidaPromocao(boolean validaPromocao) {
		super();
		this.validaPromocao = validaPromocao;
	}

	public boolean isValidaPromocao() {
		return validaPromocao;
	}

	public void setValidaPromocao(boolean validaPromocao) {
		this.validaPromocao = validaPromocao;
	}

}
```

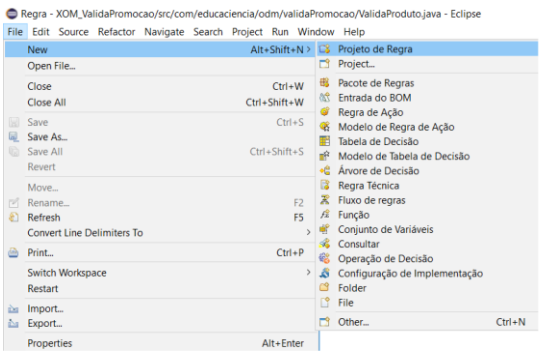


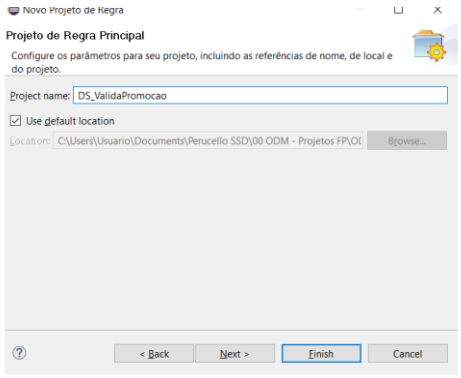
```
package com.educaciencia.odm.validaPromocao;

public class ValidaProduto {

	private String ValidaProdutonome;
	private String ValidaProdutodescricao;
	private String ValidaProdutofabricante;

	public ValidaProduto() {
		super();
	}

	public ValidaProduto(String validaProdutonome,
			String validaProdutodescricao, String
validaProdutofabricante) {
		super();
		ValidaProdutonome = validaProdutonome;
		ValidaProdutodescricao = validaProdutodescricao;
		ValidaProdutofabricante = validaProdutofabricante;
	}

	public String getValidaProdutonome() {
		return ValidaProdutonome;
	}

	public void setValidaProdutonome(String validaProdutonome) {
		ValidaProdutonome = validaProdutonome;
	}

	public String getValidaProdutodescricao() {
		return ValidaProdutodescricao;
	}

	public void setValidaProdutodescricao(String
validaProdutodescricao) {
		ValidaProdutodescricao = validaProdutodescricao;
	}

	public String getValidaProdutofabricante() {
		return ValidaProdutofabricante;
	}

	public void setValidaProdutofabricante(String
validaProdutofabricante) {
		ValidaProdutofabricante = validaProdutofabricante;
	}

}
```

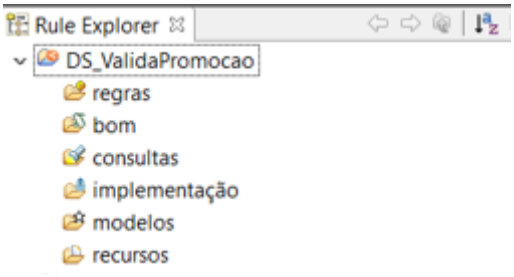




Com nossas classes criadas no nosso projeto, podemos criar nosso Projeto de Regra e nomeá-lo como "***DS_ValidaPromocao***" , então vamos nessa !

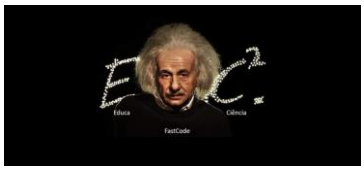




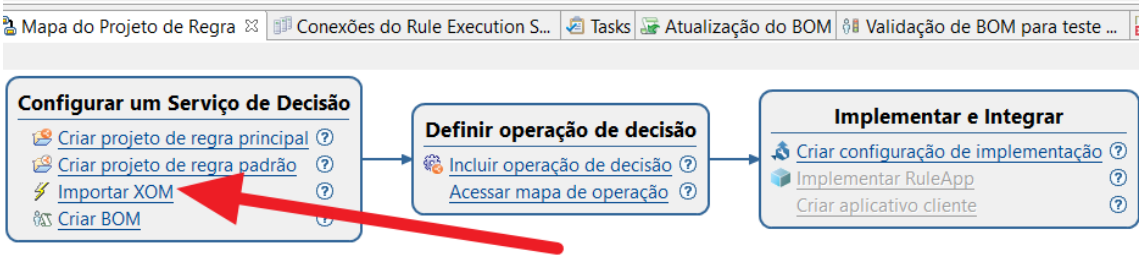


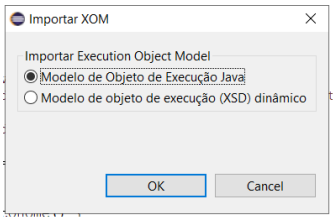
Note que temos uma arquitetura já criada de maneira automática em nosso projeto, então vamos explicar alguns tópicos importantes.

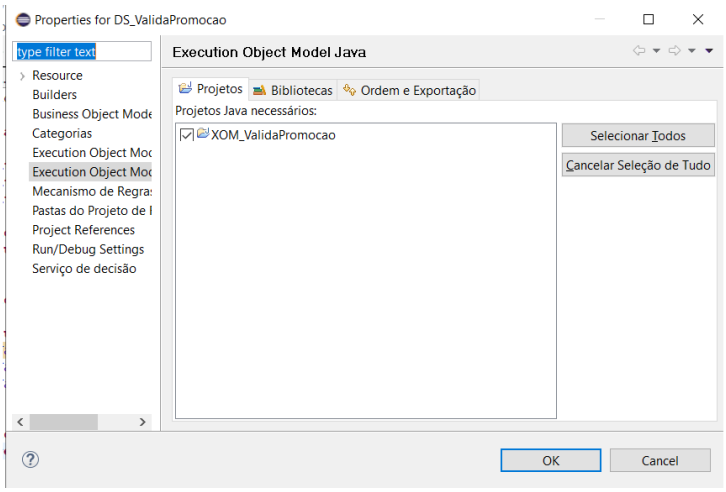
- **Regras** – onde criaremos nossa regra de ação
- **Bom** – onde receberemos nosso Model
- **Implementação** - onde criaremos nosso "run" para que seja implementado e executado nosso projeto



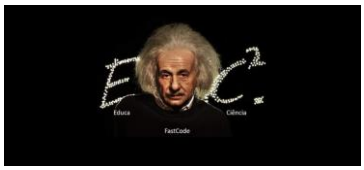
Agora que explicamos a pré arquitetura criada de maneira automática, vamos importar nosso XOM para o nosso Projeto de Regras que acabamos de criar.



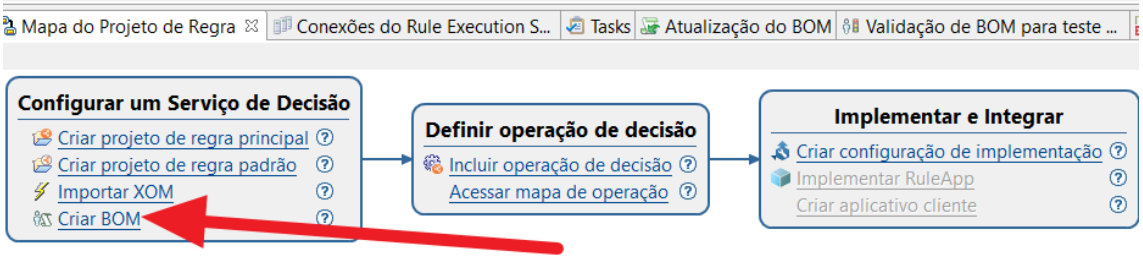


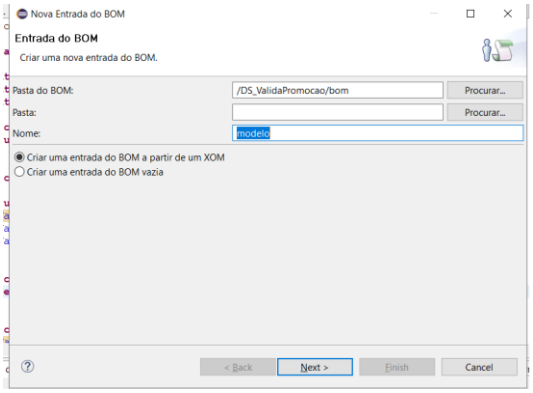


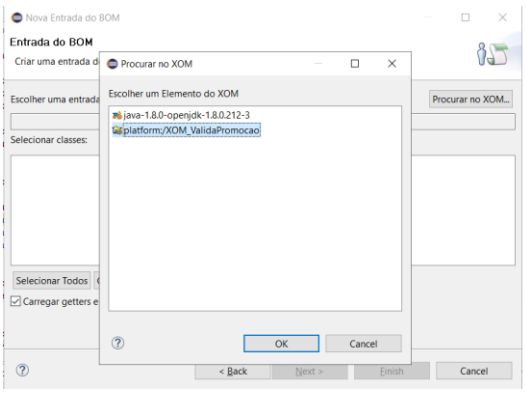
Agora que importamos o XOM , iremos criar o BOM ou Model.



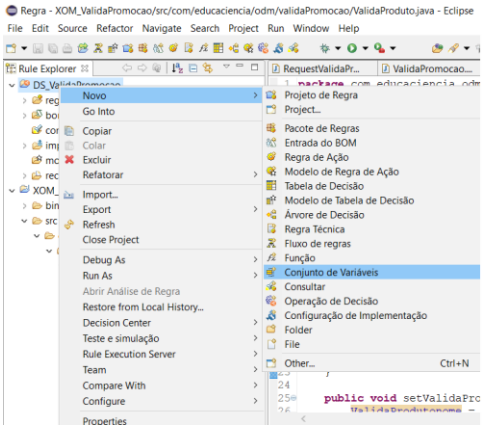
Siga os passos:

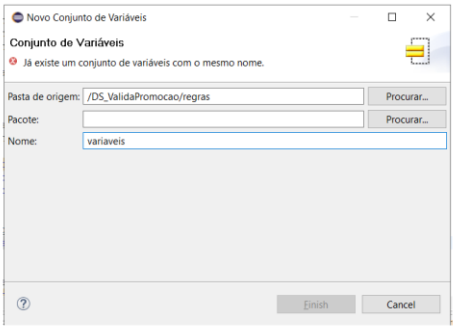


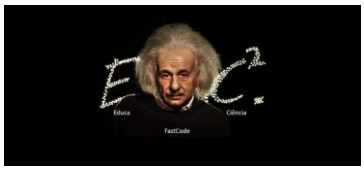


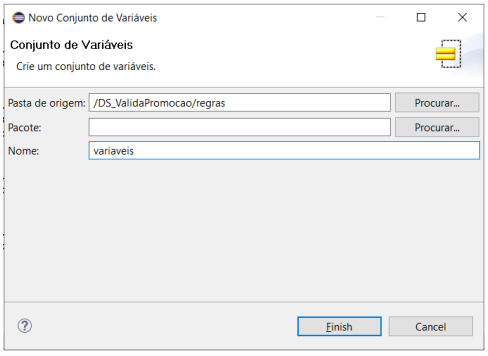


Com nosso XOM importado para o Projeto de Regra e criado nosso Model, vamos criar nossas variaveis e declará-las.

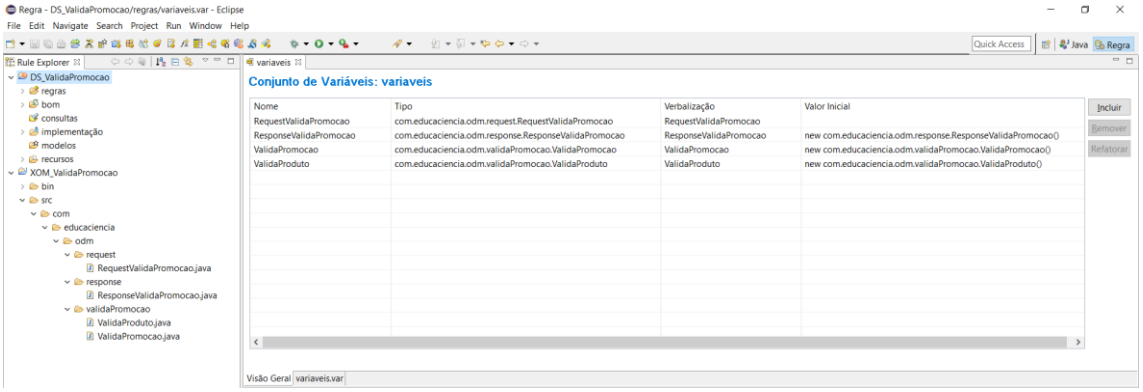




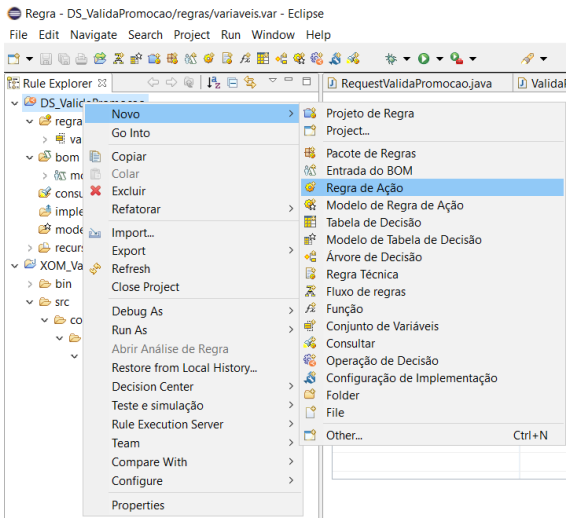


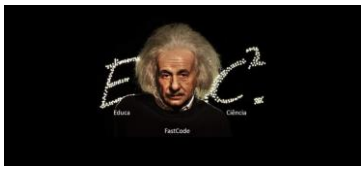


Agora devemos declarar nossas variáveis, portanto, declare as variáveis como abaixo:



Com nossas variáveis devidamente declaradas, podemos criar nossas Regras de Ação, no nosso cenário teremos duas regras onde uma irá validar se encontrou o produto para promoção e a outra para atribuirmos os valores da promoção de acordo com o que estiver declarado no nosso Projeto de Regras ou ODM.





Regra 1

```
se
        o nome produto de RequestValidaPromocao é o valida produtonome
de ValidaProduto
      e o descricao produto de RequestValidaPromocao é o valida
produtodescricao de ValidaProduto
      e o fabricante produto de RequestValidaPromocao é o valida
produtofabricante de ValidaProduto
então
Apresentar "encontrou produto na promoção.";
colocar à verdadeiro que ValidaPromocao é valida promocao ;
senão
Apresentar "produto nao esta na promocao!" ;
colocar à falso que ValidaPromocao é valida promocao ;
```

Regra 2 – validando nome do Produto

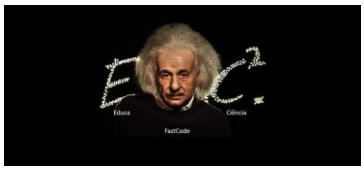
```
se
        o nome produto de RequestValidaPromocao é o valida produtonome
de ValidaProduto
então
Apresentar "Produto " + o nome produto de RequestValidaPromocao  ;
```

Regra 3 – validando descrição do Produto

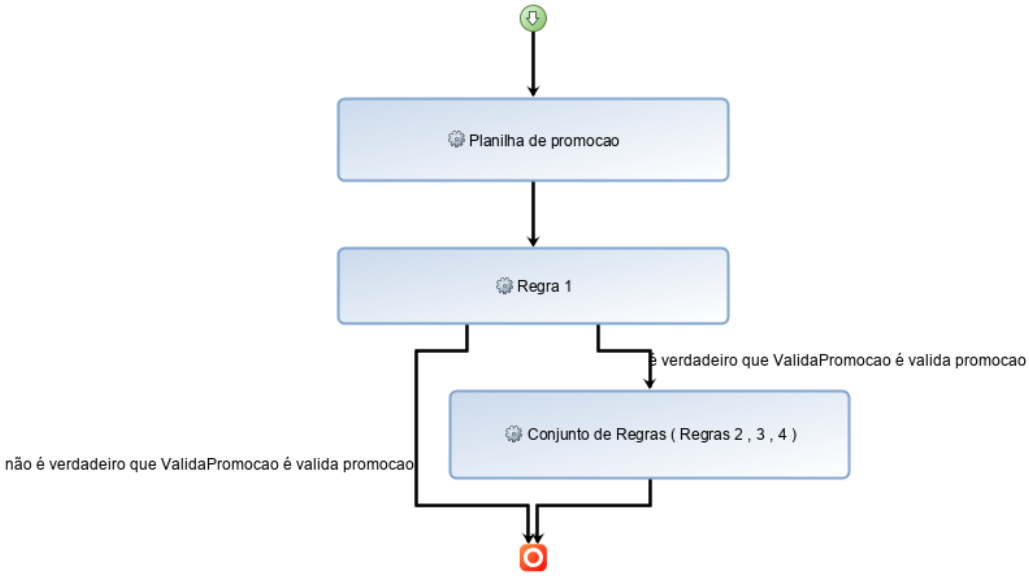
```
se
      o descricao produto de RequestValidaPromocao é o valida
produtodescricao de ValidaProduto
então
Apresentar "Descrição " + o descricao produto de
RequestValidaPromocao;
```

Regra 4 – validando fabricante do Produto

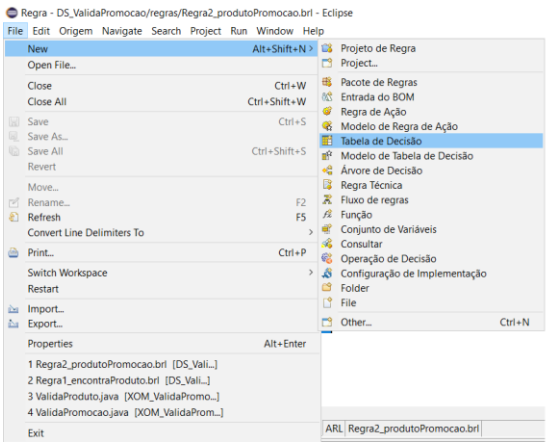
```
se
      o fabricante produto de RequestValidaPromocao é o valida
produtofabricante de ValidaProduto
então
Apresentar "Fabricante " + o fabricante produto de
RequestValidaPromocao;
Apresentar "Valor do Produto em promoção R$ " + o valor produto de
ResponseValidaPromocao ;
atribuir o valor produto de ResponseValidaPromocao a o valor produto
de ResponseValidaPromocao ;
```

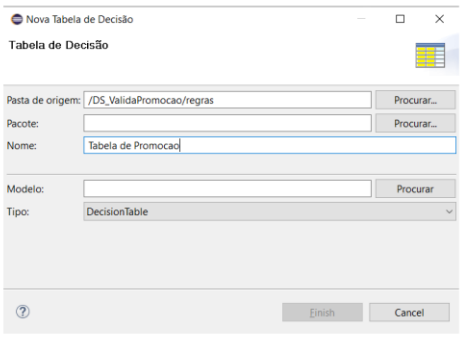


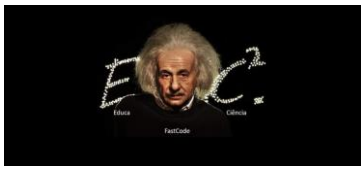
Note que criamos 3 regras de ação para validar em nosso fluxo, e atribuímos o valor apenas na última regra , isso ocorreu porque temos a regra 1 que valida a existência do produto em nossa Tabela de Decisão que criaremos logo mais.
No entanto, nesse momento , arquitetamos nosso desenho do fluxo da seguinte maneira veja:



Agora que criamos as regras, iremos criar nossa Tabela de Decisão, ou seja, a Tabela que iremos incluir os produtos em promoção para a validação da nossa Regra de Negócio.

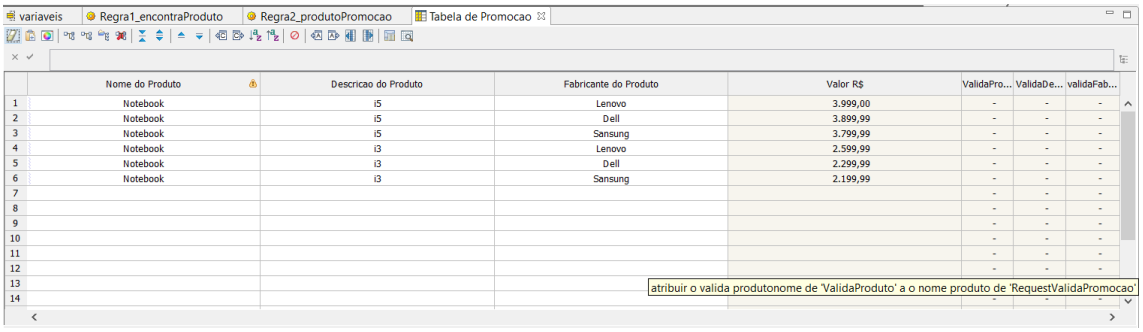






Agora com nossa tabela de Promoção criada, vamos declarar os campos da nossa tabela e sua regra.

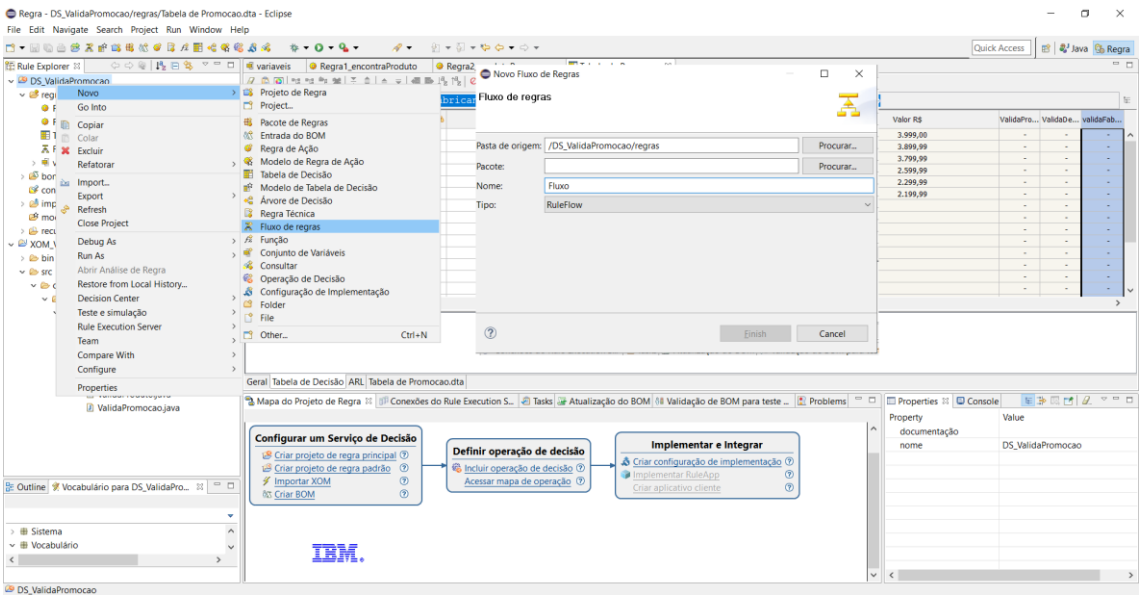
Portanto, vamos preencher nossa tabela com os produtos que entrarão em promoção conforme print abaixo e sua regra respectiva.

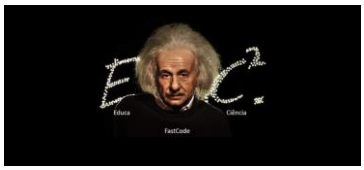


Regra:

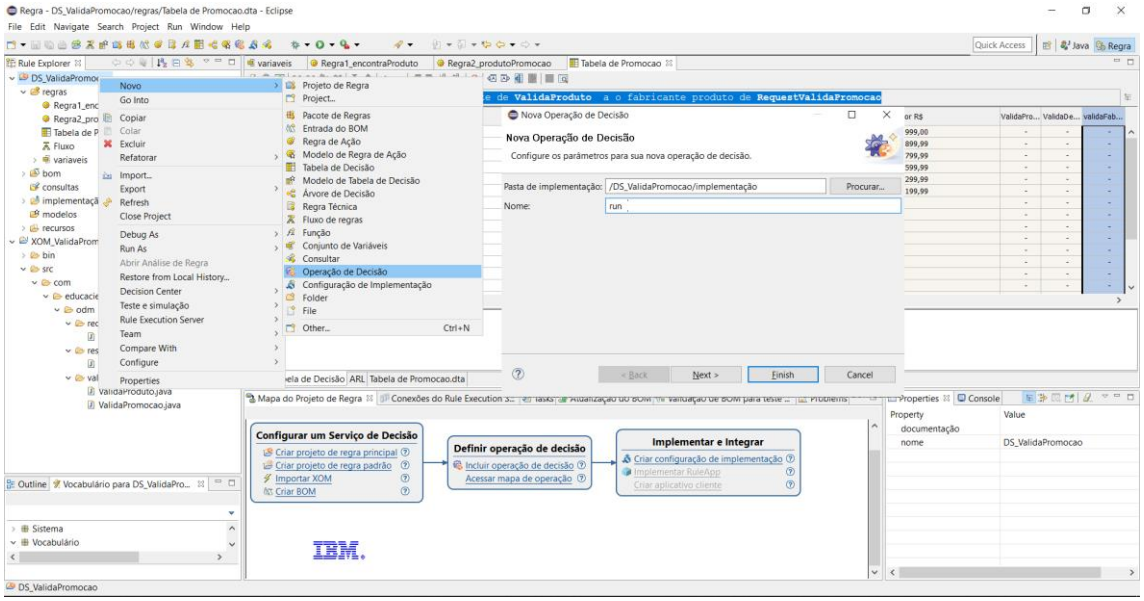
- **Nome do Produto**: o nome produto de RequestValidaPromocao é <um objeto>
- **Descrição do Produto**: o descrição produto de RequestValidaPromocao é <um objeto>
- **Fabricante do Produto**: o fabricante produto de RequestValidaPromocao é <um objeto>
- **Valor R$**: atribuir o valor produto de ResponseValidaPromocao a <uma cadeia>
- **validaProduto**: atribuir o valida produtonome de ValidaProduto a o nome produto de RequestValidaPromocao
- **validaDescricao**: atribuir o valida produtodescricao de ValidaProduto a o descricao produto de RequestValidaPromocao
- **validaFabricante**: atribuir o valida produtofabricante de ValidaProduto a o fabricante produto de RequestValidaPromocao

Pronto, com a regra da nossa Tabela de Decisão criada, iremos “desenhar” o nosso fluxo. Dessa forma, criaremos o nosso “MAIN” que será responsável por direcionar o fluxo da nossa regra de negócio.

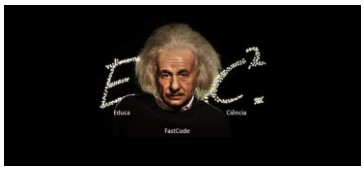




Podemos agora termos como mente , que nossa regra está criada e que nossas regras foram aplicadas de acordo com o nosso fluxo, porém precisamos agora executar, e para isso precisamos criar nossa “Operação de Decisão”. Portanto, vamos nessa !

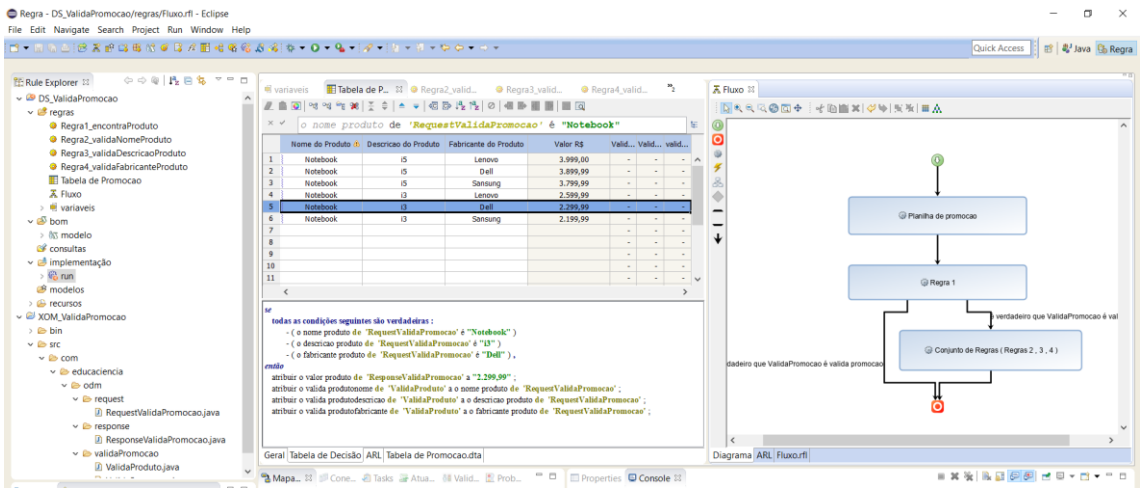


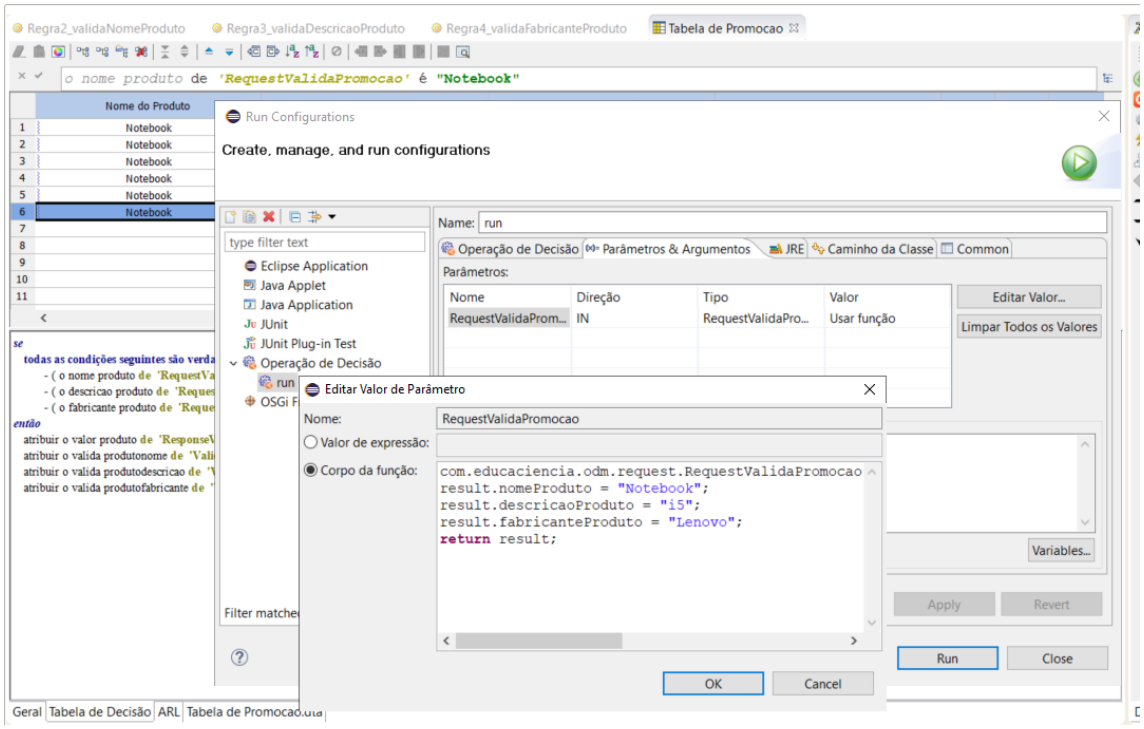
Após criar nossa Operação de Decisão e definirmos nossos parâmetros de entrada e saída, podemos testar.



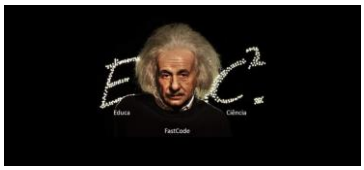
Para isso , vamos executar , lembrando que precisamos passar os objetos para que nossa regra funcione, para simularmos o funcionamento da nossa regra de negócio.

Sendo assim, vamos nessa !

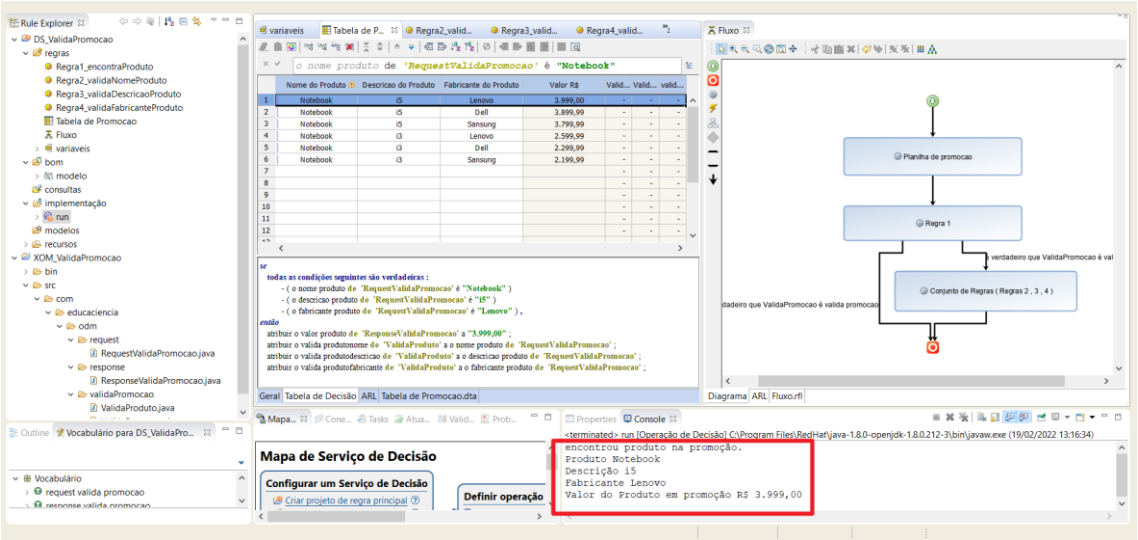




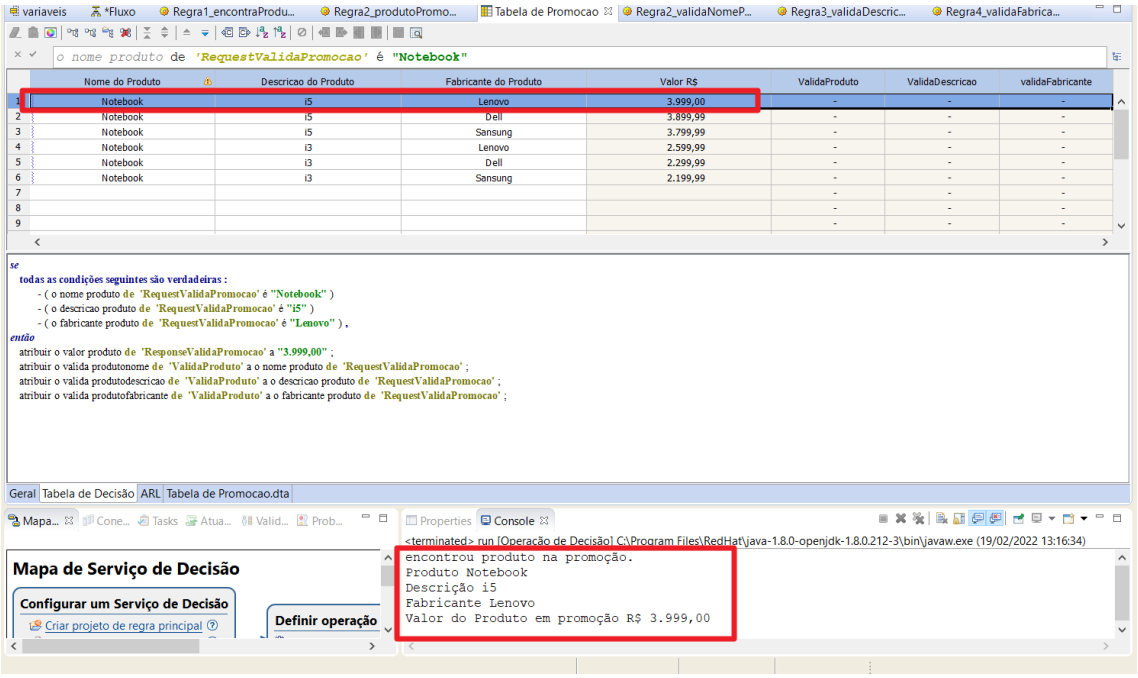
Criaremos um objeto para ver se nossa regra funciona. Nesse cenário, iremos criar um produto que já existe , para sabermos se a promoção será aplicada

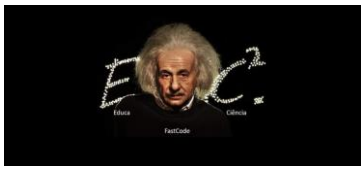


```
com.educaciencia.odm.request.RequestValidaPromocao result = new
com.educaciencia.odm.request.RequestValidaPromocao();
result.nomeProduto = "Notebook";
result.descricaoProduto = "i5";
result.fabricanteProduto = "Lenovo";
return result;
```



Note que o produto existe na nossa Tabela de Decisão, passou pela Regra 1 que valida que o produto está inscrito em uma promoção e acessou uma cadeia de regras até que se valida o valor do nosso Produto em Promoção.

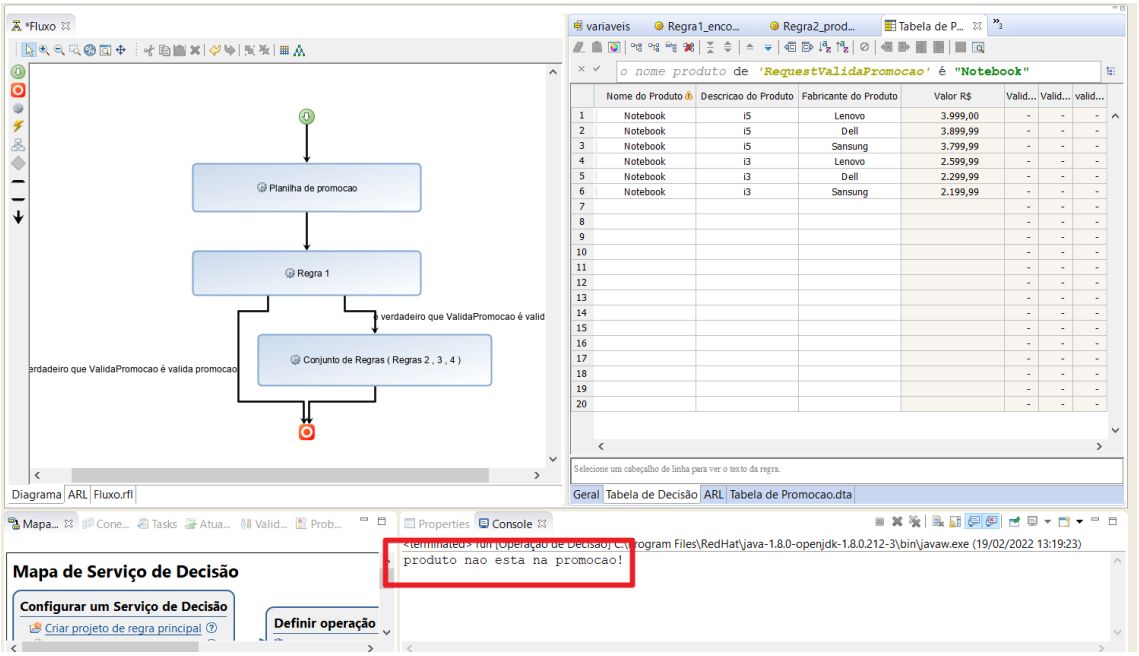




Agora, vamos passar um produto que não está cadastrado e vamos ver o comportamento:

```
com.educaciencia.odm.request.RequestValidaPromocao result = new
com.educaciencia.odm.request.RequestValidaPromocao();
result.nomeProduto = "Monitor";
result.descricaoProduto = "14polegadas";
result.fabricanteProduto = "Sansung";
return result;
```

Vamos ver o que ocorre, portando clique OK e Run.



Note que o produto não foi encontrado, como fizemos na Regra 1 , se o produto não participasse da nossa promoção, informaríamos que o produto não pertence a uma promoção.

Portanto, podemos concluir que nossa regra funcionou perfeitamente, lembrando que nesse artigo apenas estamos trabalhando com o Rule Designer, e não estamos publicando no Decision Center e no Decision Server, já que esses passos serão abordados em outro artigo.

Pelo Canal do ExplorandoTI , fiz uma palestra onde abordei o poder do ODM (Operational Decision Manager). Se você não viu , sugiro que visite o link: https://www.youtube.com/watch?v=YlqkO_RmwuY&t=2431s e confira !

Recentemente abordamos também o poder do ODM em HyperAutomation, se você não viu , acesse o link – https://www.youtube.com/watch?v=BPI9j_pOuE0

Temos uma playlist do EducaCiência FastCode disponível para você: https://www.youtube.com/playlist?list=PLG_driR73c6OGrAJxXQD0xj677LU_zfGM

Espero ter colaborado de alguma maneira !

Abraços e até mais !