**Treinamento SOA com IIB**

***Nível III***

**Março/2018**

CONTROLE DE VERSÕES

| **Versão** | **Data** | **Autor (es)** | **Descrição das Alterações** |
| --- | --- | --- | --- |
| 1.0 | 03/2018 | Sergio Fonseca da Silva | Criação |

Figura1 ÍNDICE

1 Apresentação 4

2 Modelo de serviço. 4

2.1 Entendendo os projetos e suas camadas. 4

3 Implementação. 5

3.1 Implementando o modelo canônico. 5

3.2 RS - Implementando a camada de exposição ou serviço corporativo. 8

3.3 RS - Implementando a camada de validação. 12

3.4 RS - Vinculando o canônico ao projeto de corporativo. 18

3.5 RS - Implementando a camada adapter – Json para Canônico. 19

3.6 RS - Implementando a camada de mediator. 22

3.7 CP - Implementando a camada de composição. 24

3.8 EN - Implementando a camada de mediator. 25

3.9 EN - Implementando a camada de enable – parte I. 26

3.10 EN - Vinculando o canônico ao projeto de enable. 31

3.11 EN - Implementando a camada adapter – Canônico para Json. 32

3.12 EN - Implementando a camada de enable – parte II. 34

3.13 EN - Implementando a camada adapter – JSON para Canônico. 35

3.14 RS - Implementando a camada adapter – Canônico para JSON. 38

3.15 Ligações finais. 43

3.16 Realizando deploy. 46

# Apresentação

Dando continuidade ao modelo de arquitetura corporativa apresenta no curso anterior “Nível II”, vamos manter, praticamente a mesma arquitetura de desenvolvimento, entretanto iremos adicionar os seguintes itens:

* Construção de uma Application.
* Elaboração de uma biblioteca compartilhada.
* Definição do modelo canônico.
* Orquestração e composição de Serviços.
* Invocação de projetos externos.

# Modelo de serviço.

Ainda tendo como referência os cursos anteriores, vamos construir o mesmo serviço de cotação de dólar, porem atendendo os requisitos apresentados acima.

# Entendendo os projetos e suas camadas.

Nessa modelo de arquitetura para desenvolvimento, nosso serviço de integração será composto conforme abaixo:

* A ***API-REST*** que será exposta no barramento com o nome: ***undbr\_busca\_cotacao\_rs\_v2.***
* O ***Application***, que será utilizado para fazer a orquestração e ou composição do serviço, com o nome: ***undbr\_busca\_cotacao\_cp\_v1.***
* O ***Application*** que será responsável pela invocação dos serviços expostos nos legados, que receberá o nome: ***undbr\_busca\_cotacao\_en\_v1.***
* A ***Shared-Library*** que será responsável em agrupar e organizar as estruturas dos dados, que receberá o nome: ***undbr\_commons\_canonico\_v1.***

# Implementação.

As próximas paginas serão sobre as implementações dos serviços.

# Implementando o modelo canônico.

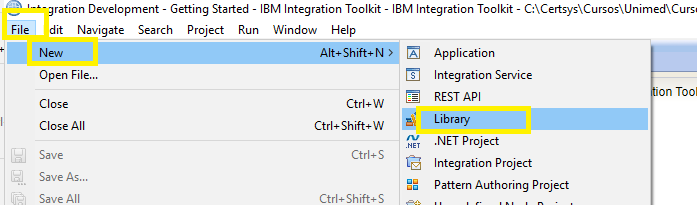
O modelo canônico de forma simplifica resume em agrupar diversos artefatos, que no nosso caso serão arquivos xsd, normalmente o modelo canônico padroniza a forma e define os padrões e estruturas de dados que devem ser compartilhados por todos os sistemas da companhia.

Entretanto esse raciocínio vem sendo polemico dentro da comunidade SOA, alguns defende o raciocínio que o modelo canônico deve estar presente apenas no barramento e deixando os demais sistemas legados autuarem conforme sua própria metodologia de desenvolvimento, não sofrendo influência da área de integração da empresa.

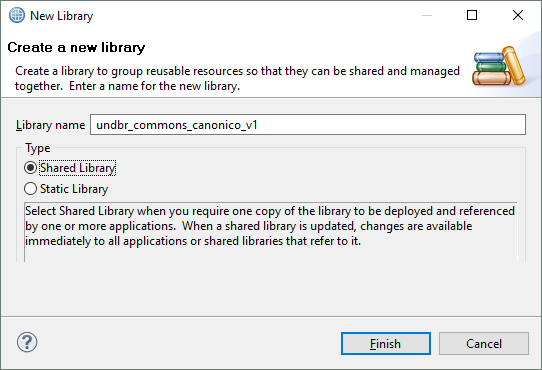
Segundo essa nova tendência e auxiliados pela facilidade que o IIB nos oferece vamos adotar esse novo modelo de implementação, para isso vamos construir uma ***Shared-Library***:

|  |
| --- |
| **Iniciando a construção** |

* + 1. Clique em ***File, New, Library***.



* + 1. Em Library name informe: ***undbr\_commons\_canonico\_v1.***
    2. Selecione Shared Library.

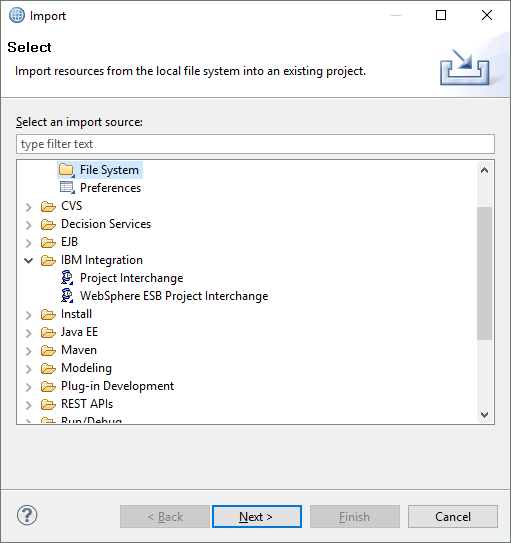


* + 1. Clique em ***Finish***.

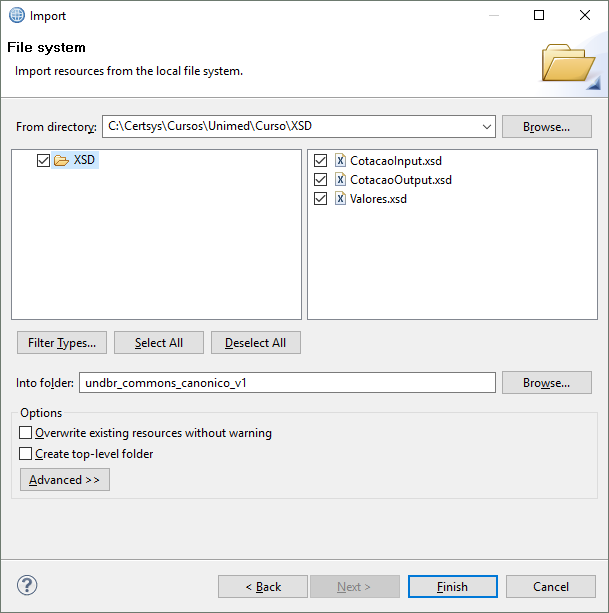
|  |
| --- |
| ***Observação:***  O xsd é um arquivo externo, então precisamos importar o mesmo para dentro do projeto canônico. |

|  |
| --- |
| **Importando o XSD** |

* + 1. Selecione o projeto de ***Shared Library*** do canônico.
    2. Clique em ***Import***.
    3. Selecione ***File System***.
    4. Clique em ***Next***.



* + 1. Clique no botão ***Browser***.
    2. Selecione a pasta onde o xsd se encontram.
    3. Selecione a direita ***XSD***, node que todos os artefatos estão selecionados.
    4. Cerifique-se que em Info folder está setado para o projeto.
    5. Clique em ***Finish***.



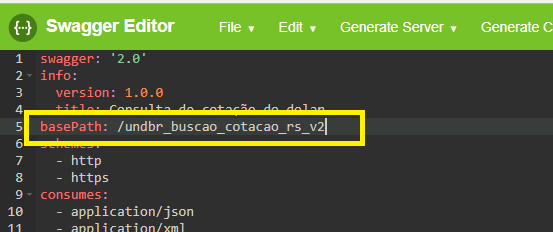
# RS - Implementando a camada de exposição ou serviço corporativo.

A primeira etapa em rumo ao novo padrão de desenvolvimento é reutilizarmos o Swagger adotado na apostila anterior, mudaremos a tag ***basePath*** do arquivo Swagger para suportar a nova versão do serviço.

Essa é uma boa prática para permitir dupla convivência de versões diferentes no mesmo projeto.

Nessa camada termos apenas os patterns: ***adapter, mediator*** e ***validate***, já comentados anteriormente, portanto temos que:

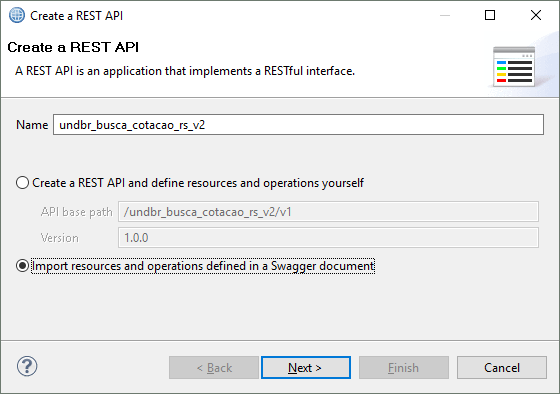
* + 1. No browser, abrir o arquivo ***Swagger*** no editor on-line.
    2. Altere o basePath para: ***/undbr\_busca\_cotacao\_rs\_v2***.
    3. Realize o download do mesmo.
    4. Disponibilize na pasta Swagger local da máquina.
    5. Renomeei o nome do arquivo json conforme sua convencia, mas que tenha sentido com relação ao projeto.



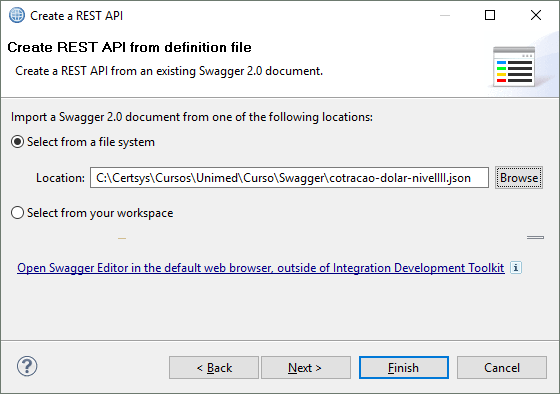
|  |
| --- |
| ***Observação:***  Nunca coloque a versão no nome do arquivo Swagger, o fato de trocar o nome do arquivo o IIB perde a referência nos mapeamentos dos fluxos já existentes. |

* + 1. No ***Workspace***, selecione ***File, New, REST API***.

Em Name digite: ***undbr\_busca\_cotacao\_rs\_v2***.



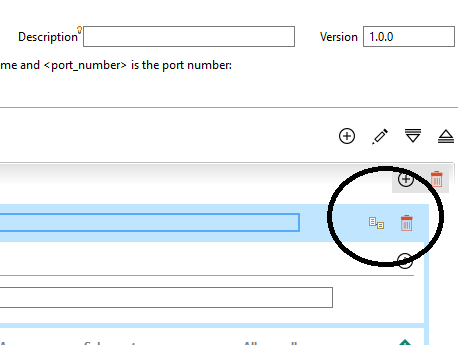
* + 1. Selecione ***Import Resources...***
    2. Selecione o arquivo JSON: ***cotacao-dolar-nivelIII.json*** na pasta de download.



* + 1. Clique em ***Finish***.

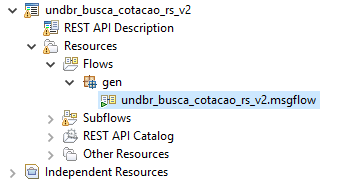
|  |
| --- |
| **Implementando o msgflow** |

* + 1. Clique no botão: ***Open the subflow for the operation***, ilustrado abaixo.

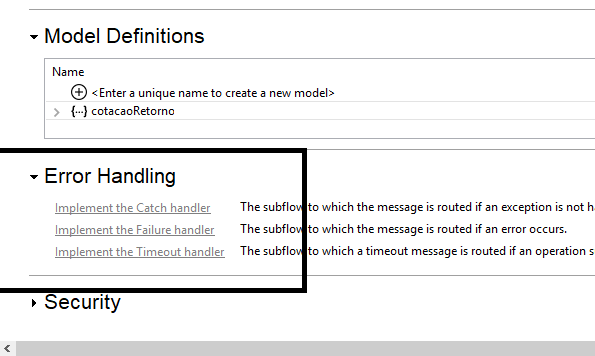
******

|  |
| --- |
| **Implementando os tratamentos para as exceções** |

* + 1. Duplo clique no flow ***undbr\_busca\_cotacao\_rs\_v2.***



* + 1. Role a página para baixo no arquivo json até o bloco ***Error Handling.***



* + 1. Execute duplo clique nos links abaixo, para criar os subflow de erros correspondentes:
       1. ***Implements the Cactch handler.***
       2. ***Implements the Failure handler.***
       3. ***Implements the Timeout handler.***

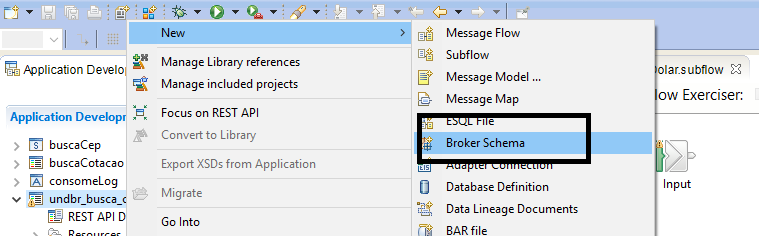
|  |
| --- |
| ***Observação***:  No momento oportuno vamos implementar os fluxos de mensagens para tratamento de erro. |

# RS - Implementando a camada de validação.

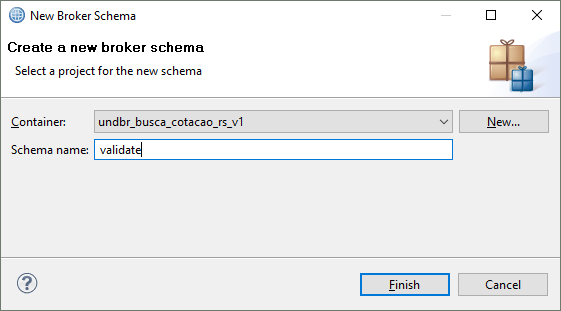
A primeira atividade que o serviço de integração deve realizar é a validação dos dados de entrada, dessa forma caso exista alguma inconformidade com os mesmos o serviço notifica o consumidor e suspende a execução do serviço, economizando processamento desnecessário.

Dando ênfase no nosso modelo de desenvolvimento vamos criar um Broker Schema que é utilizado para definirmos a organização estrutural do projeto.

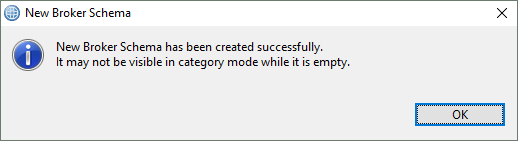
* + 1. Selecione o projeto.
    2. Clique em ***New***



* + 1. Clique em ***Broker Schema.***
    2. Certifique que o ***Container*** selecionado é o serviço que estamos definindo.
    3. Em ***Schema name***, digite o nome de nosso pacote: ***validate***.

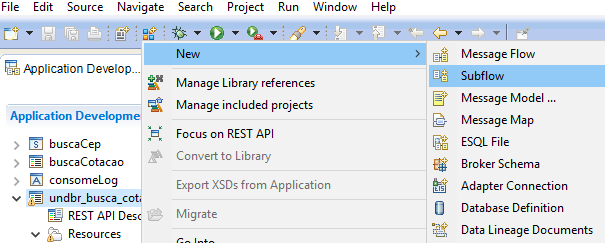


* + 1. Ao clicar em ***Finish***, será apresentada a seguinte mensagem.

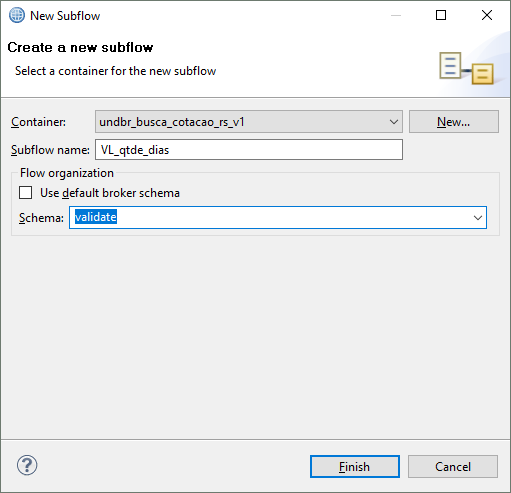


Agora vamos construir nosso subflow de validação, essa validação consiste em verificar se o consumidor do serviço enviou o parâmetro de entrada correspondente a quantidade de dias, caso negativo vamos interromper o fluxo e notificar o consumidor. Para isso:

* + 1. Selecione o projeto.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

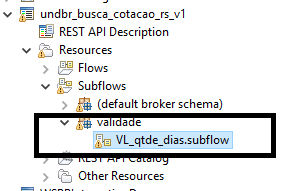
****

* + 1. Em Name digite: ***VL\_qtde\_dias***, sendo:
    2. Certifique que o Container selecionado é o serviço em implementação.
    3. Digite o seguinte nome para o subflow: ***VL\_qtadeDias***
    4. Deselecione a opção: ***Use default broker schema***.
    5. Em schema selecione o broker schema que criamos, nesse caso: ***validate.***

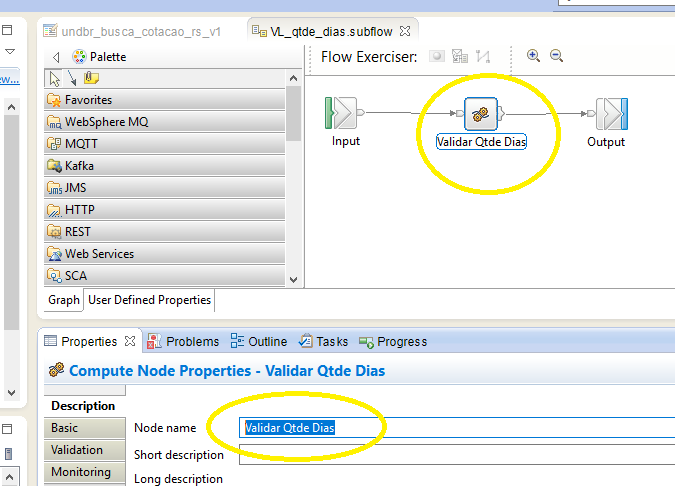


* + 1. Clique em ***Finish***.

|  |
| --- |
| **Ao concluir as etapas teremos** |



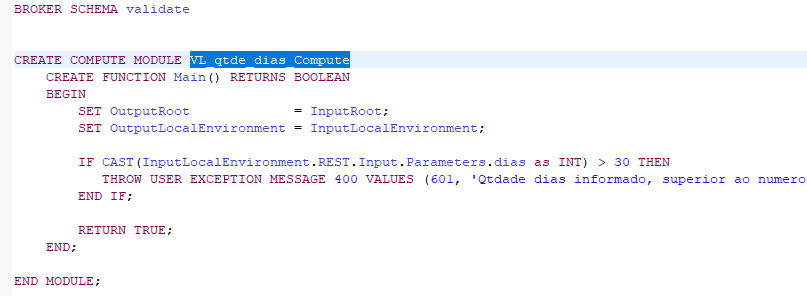
* + 1. Duplo clique no subflow ***VL\_qtde\_dias***, para abrir a área de trabalho.
    2. Selecione o node de compute e arraste para área de trabalho.
    3. Selecione aba de ***description*** e em node ***name*** informe: ***Validar Qtde Dias.***



* + 1. Duplo clique no node de ***compute node***, será aberto a área de edição dos comandos ***ESQL***.
    2. Entre o ***begin*** e o ***end***, implemente o seguinte código ***ESQL*** para validação.

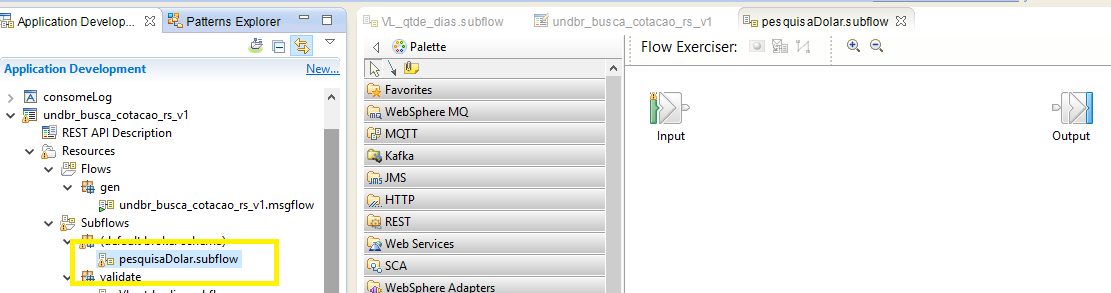
|  |
| --- |
| SET OutputRoot = InputRoot;  SET OutputLocalEnvironment = InputLocalEnvironment;  IF CAST(InputLocalEnvironment.REST.Input.Parameters.dias as INT) > 30 THEN  THROW USER EXCEPTION MESSAGE 400 VALUES (601, 'Qtdade dias informado, superior ao numero máximo permitido!');  END IF;    RETURN TRUE; |

|  |
| --- |
| **Abaixo código final do ESQL no node de compute.** |

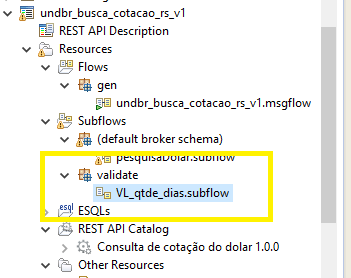


Com as etapas anterior, concluímos a construção do subflow de validação, porem precisamos disponibilizar o mesmo no flow principal da aplicação, então:

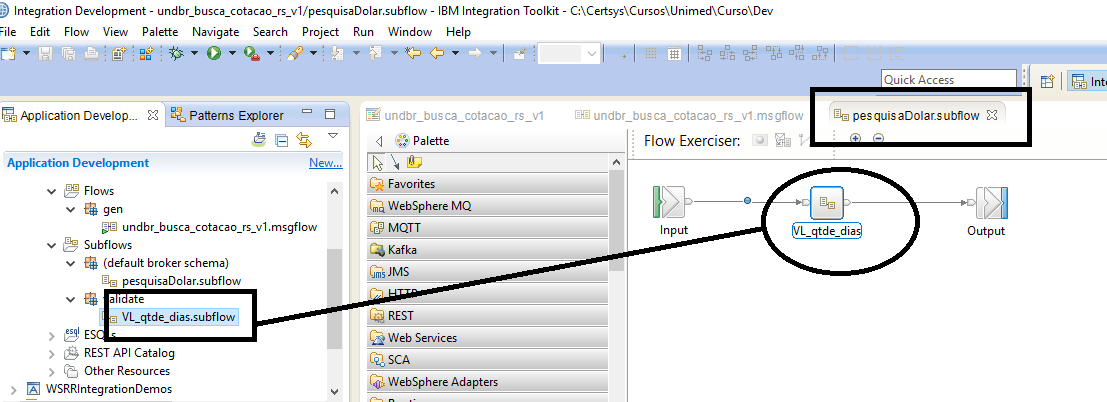
* + 1. Na lateral esquerda do toolkit, navegue até encontrar o subflow “***pesquisaDolar.subflow***”, clique duas vezes para abrir a área de desenvolvimento.



* + 1. Ainda na lateral esquerda do toolkit, navegue até encontrar o subflow : ***VL\_qtde\_dias***.



* + 1. Selecione a estrutura Subflows “***VL\_qtde\_dias***” e araste para a área de trabalho entre os nodes ***input*** e o de ***output*** do flow ***pesquisaDolar***.

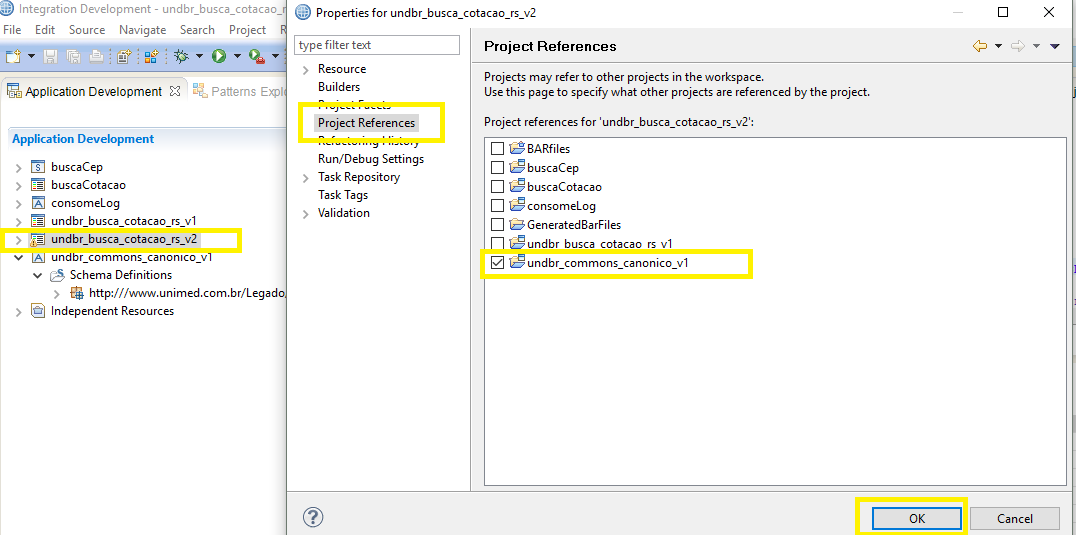


* + 1. Realize as ligações entre os notes.

# RS - Vinculando o canônico ao projeto de corporativo.

Para que aplicação possa ter acesso a estrutura de dados, precisamos criar um vínculo do canônico com o projeto, para isso:

* + 1. Selecione o projeto que receberá o vínculo.
    2. Clique em ***Properties***.
    3. Selecione Project ***References***.
    4. Selecione o projeto que será vinculado.
    5. Clique em ***OK***.

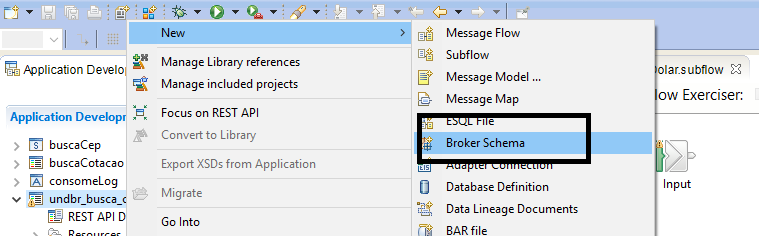


# RS - Implementando a camada adapter – Json para Canônico.

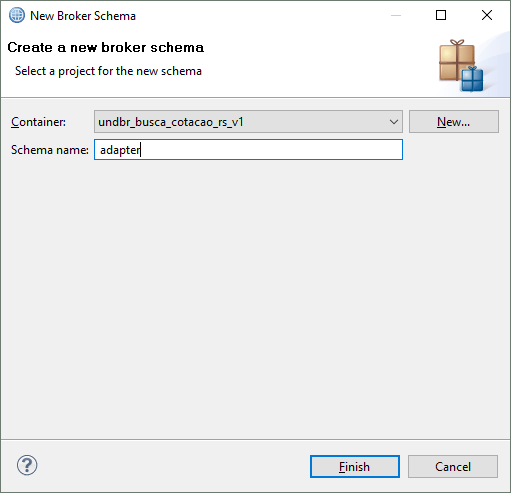
Nesse fluxo de adaptação, terá a responsabilidade de receber uma requisição ***JSON*** e converter para o modelo ***canônico***, disponibilizando nesse formato para as demais camadas internas do barramento.

|  |
| --- |
| **Criando Broker Schema** |

* + 1. Selecione o projeto.
    2. Clique em ***New***
    3. Clique em ***Broker Schema.***



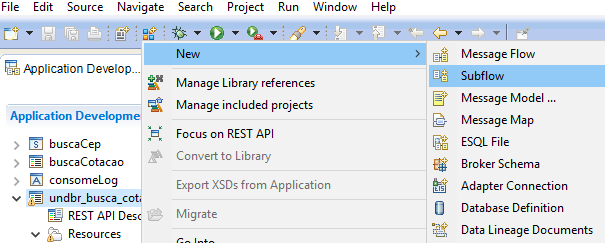
* + 1. Certifique que o ***container*** selecionado é o serviço que estamos definindo.
    2. Em ***Schema name***, digite o nome de nosso pacote: ***adapter***.



* + 1. Clique em ***Finish***.

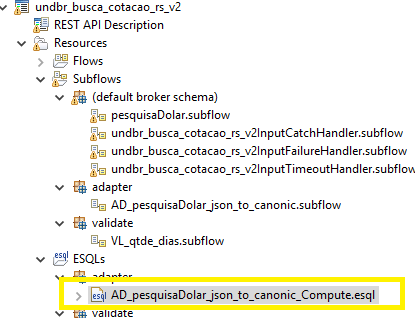
|  |
| --- |
| **Vamos construir o subflow de adaptação.** |

* + 1. Selecione o projeto.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

****

* + 1. Em Name digite: ***AD\_pesquisaDolar\_json\_to\_canonic***.
    2. Clique em ***Finish***.

|  |
| --- |
| **Teremos a imagem abaixo:** |



|  |
| --- |
| **Editando o código ESQL.** |

* + 1. Selecione o subflow: AD\_pesquisaDolar\_json\_to\_canonic.

e realize duplo clique, cole o código ***ESQL*** abaixo entre a tag ***begin*** e ***end***:

|  |
| --- |
| SET OutputRoot = InputRoot;  SET OutputLocalEnvironment = InputLocalEnvironment;  DECLARE nsDias NAMESPACE 'http:///www.unimed.com.br/Legado/CotacaoInput';  SET OutputRoot.XMLNSC.nsDias:Cotacao.qtdeDias = InputLocalEnvironment.REST.Input.Parameters.dias;  RETURN TRUE; |

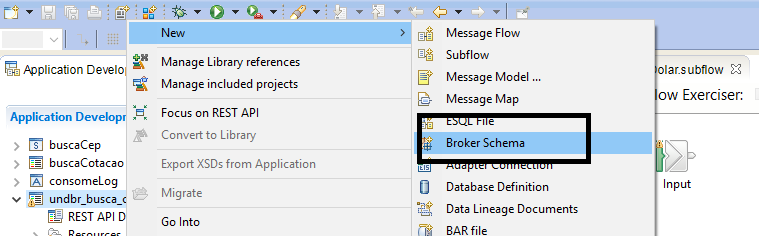
|  |
| --- |
| ***Observação***:  Não esqueça de setar o compute mode para ***LocalEnvironment*** and ***Message***. |

# RS - Implementando a camada de mediator.

Ainda no ***API-Rest***, implementados a camada mediator, responsável em separar a requisição que chega no barramento com as demais camadas ou subflows do projeto.

Dando ênfase ao modelo de desenvolvimento vamos criar um Broker Schema que é utilizado para definirmos a organização estrutural do serviço.

* + 1. Selecione o projeto.
    2. Clique em ***New***
    3. Clique em ***Broker Schema.***

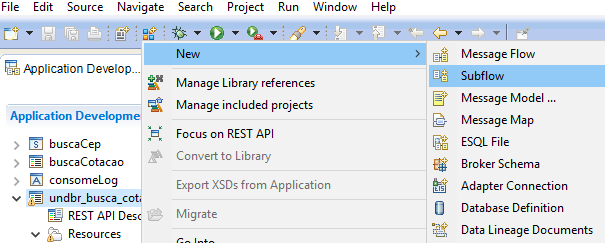


|  |
| --- |
| **Certifique que o container selecionado e o serviço que estamos definindo.** |

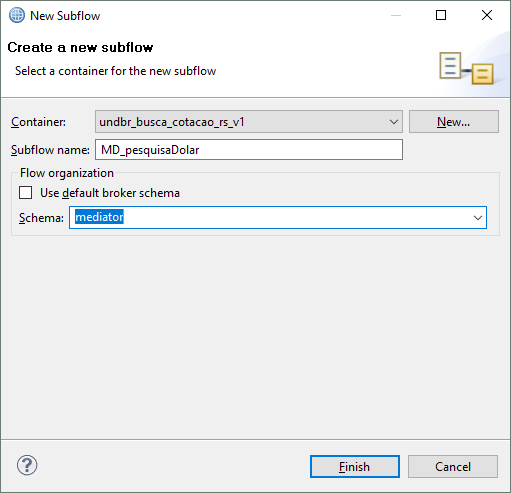
* + 1. Em ***Schema name***, digite o nome de nosso pacote: ***mediator***.

|  |
| --- |
| **Vamos construir nosso subflow de mediação. Para isso:** |

* + 1. Selecione o projeto.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

****

* + 1. Em Name digite: ***MD\_pesquisaDolar.***
    2. Certifique que o Container selecionado é o serviço em implementação.
    3. Digite o seguinte nome para o subflow: ***MD\_pesquisaDolar***.
    4. Deselecione a opção: ***Use default broker schema***.
    5. Em schema selecione o broker schema que criamos, nesse caso: ***mediator***.
    6. Clique em ***Finish***.



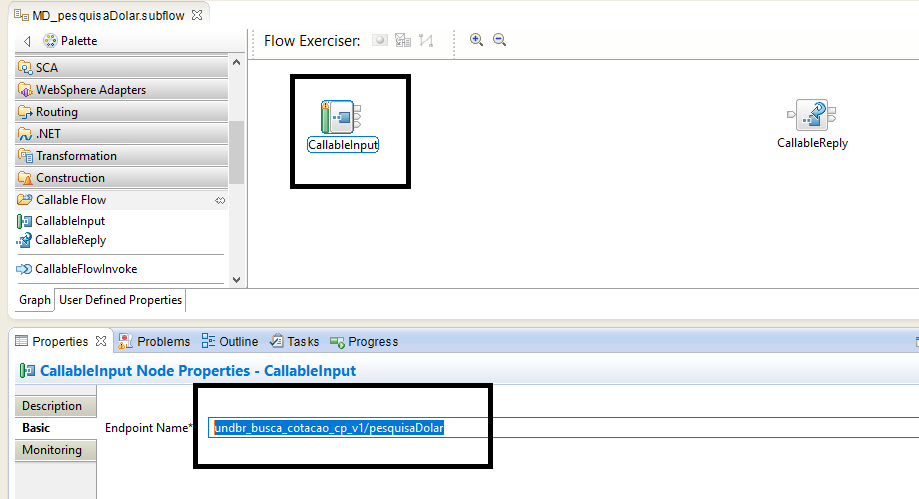
|  |
| --- |
| ***Observação***:  Por hora vamos parar por aqui, antes de continuamos, precisamos primeiro criar o projeto de composição de serviços, após a conclusão do mesmo voltaremos a implementação dessa camada no item: Implementando a camada de mediator – parte II. |

# CP - Implementando a camada de composição.

Essa camada poderá ser substituída a medida que novas composição são solicitadas e perfeitamente possível a existência de dupla convivência, sua responsabilidade é realizar a orquestração dos serviços ou implementar os serviços de negócio.

|  |
| --- |
| **Vamos a implementação** |

* + 1. Clique em ***File, New, Application***.
    2. Em name informe: ***undbr\_busca\_cotacao\_cp\_v1.***
    3. Selecione e remova o nodes de ***Input*** e ***Output***.
    4. Na paleta de nodes, clique em ***Callable Flow***.
    5. Arraste e solte na área de trabalho os node ***Callable Input*** e ***Callable Reply***.
    6. Selecione o node ***Callable Input*** em ***Endpoint Name***, informe: ***undbr\_busca\_cotacao\_cp\_v1/pesquisaDolar.***

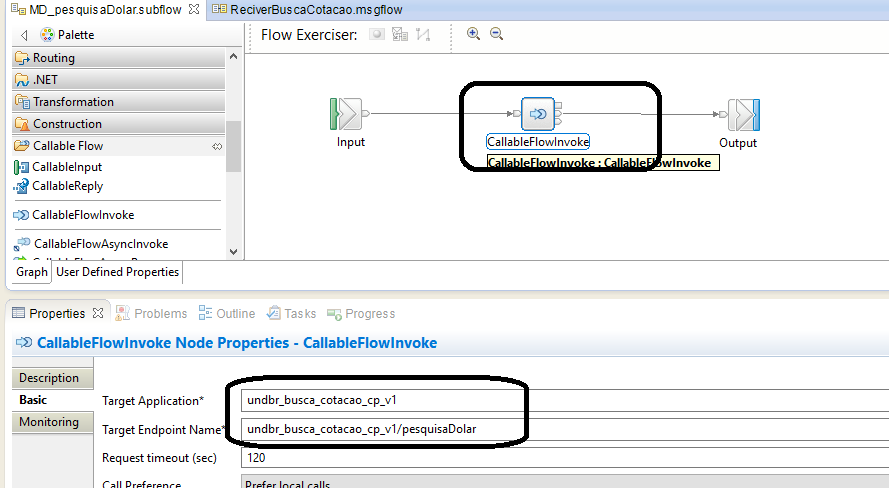


|  |
| --- |
| ***Observação***:    Por hora vamos parar por aqui, antes de continuamos, precisamos primeiro criar o projeto de chamadas atômicas/enable nos legados, após a conclusão do mesmo voltaremos a implementação dessa camada. |

# EN - Implementando a camada de mediator.

Voltamos novamente a nossa ***API-REST***, que está com fluxo de composição parcialmente construída faltando implementar a chamada remota do mediator ao projeto composite, que criamos anteriormente, para isso:

* + 1. Selecione o projeto: ***undbr\_busca\_cotacao\_rs\_v2***.
    2. Localize e realize um duplo clique no subflow: ***MD\_pesquisaDolarw***.
    3. Na paleta de nodes, clique em ***Callable Flow*** e arraste e solte na área de trabalho o node ***CallableFlowInvoke***.
    4. Com ***CallableFlowInvoke*** selecionado, clique em ***Basic*** e preencha as seguintes propriedades:
       1. Target Application: ***undbr\_busca\_cotacao\_cp\_v1.***
       2. Target EndPoint Name: ***undbr\_busca\_cotacao\_cp\_v1/pesquisaDolar***.



# EN - Implementando a camada de enable – parte I.

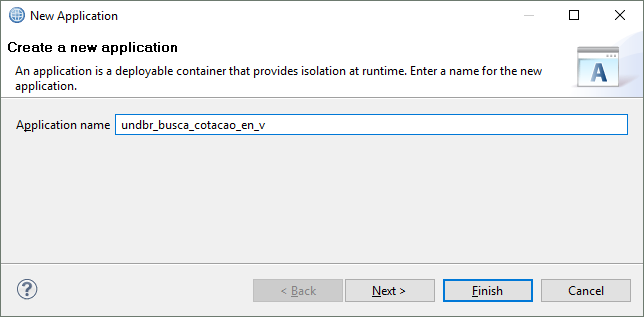
Até esse ponto implementamos parcialmente o serviço coorporativo que é a camada ***REST-API*** , também desenvolvemos o serviço de negócio que é a camada ***Composite***.

Portanto nos resta implementar a ultima camada, conhecido como camada de ***enable*** ou camada de chamada a atômica.

Tal camada tem a responsabilidade de realizar as chamadas aos serviços dos legados, que são os provedores de dados ao barramento.

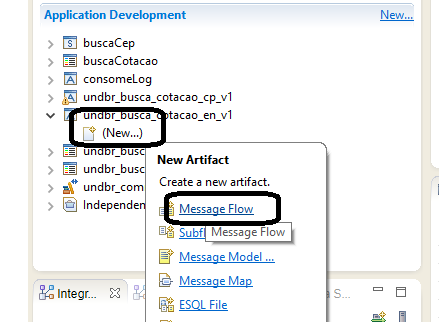
Para isso:

* + 1. Clique em ***File, New, Application***.
    2. Em name informe: ***undbr\_busca\_cotacao\_en\_v1.***
    3. Clique em ***Finish***.

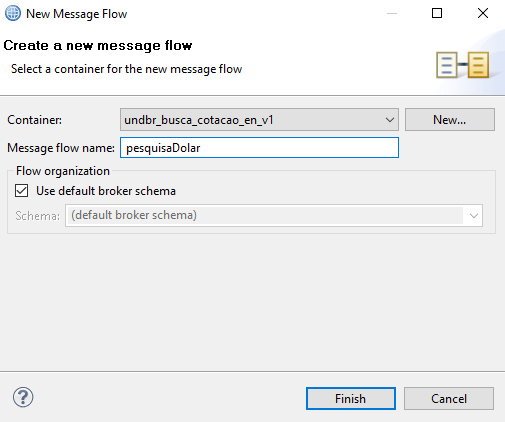


|  |
| --- |
| **Construindo o flow principal.** |

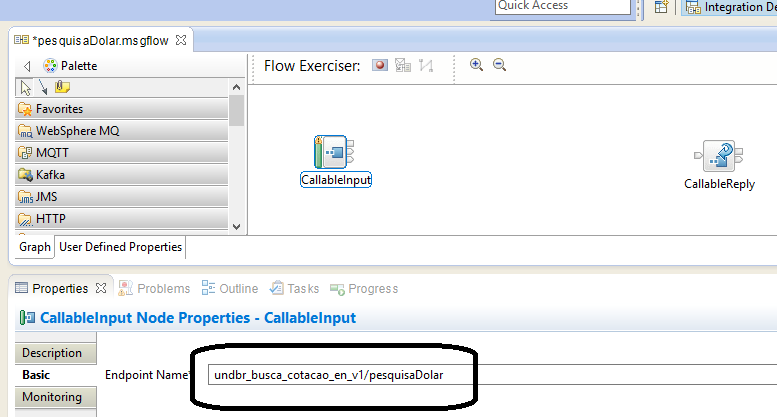
* + 1. Selecione o projeto.
    2. Clique em ***New.***
    3. Clique em ***Message Flow***.



* + 1. Em ***message flow name*** informe: ***pesquisaDolar***.

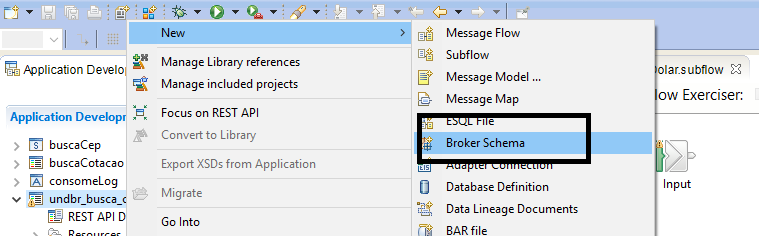


* + 1. Clique em ***Finish***.
    2. Selecione o flow criado e realize duplo clique para abrir a área de trabalho.
    3. Na paleta de nodes, clique em ***Callable Flow***.
    4. Arraste e solte na área de trabalho os node ***Callable Input*** e ***Callable Reply***.
    5. Selecione o node ***Callable Input*** em ***Endpoint Name***, informe: ***undbr\_busca\_cotacao\_en\_v1/pesquisaDolar.***



|  |
| --- |
| ***Observação***:  Precisamos construir dois brokers schemas, o de enable e o de adapter, prosseguindo: |

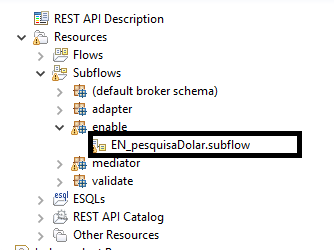
* + 1. Clique em ***Broker Schema.***



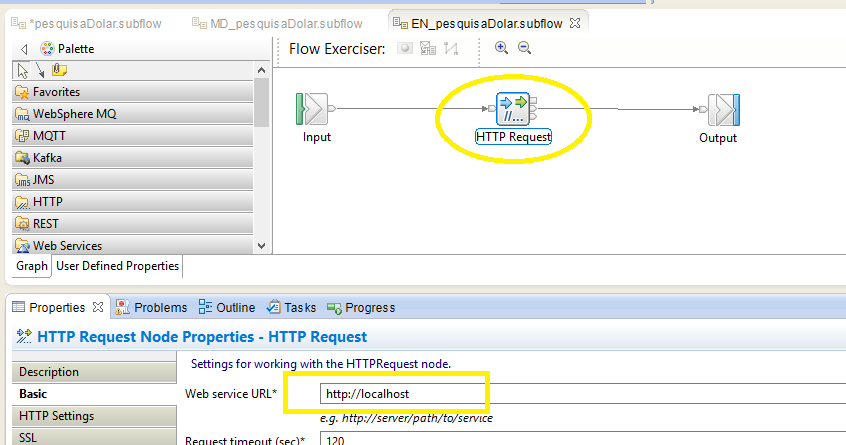
* + 1. Certifique que o ***container*** selecionado e o serviço que estamos definindo.
    2. Em ***Schema name***, digite o nome de nosso pacote: ***enable***.
    3. Clique em Finish.

|  |
| --- |
| ***Observação:***  Repita o procedimento acima e crie o broker schema com o nome ***adapter***. |

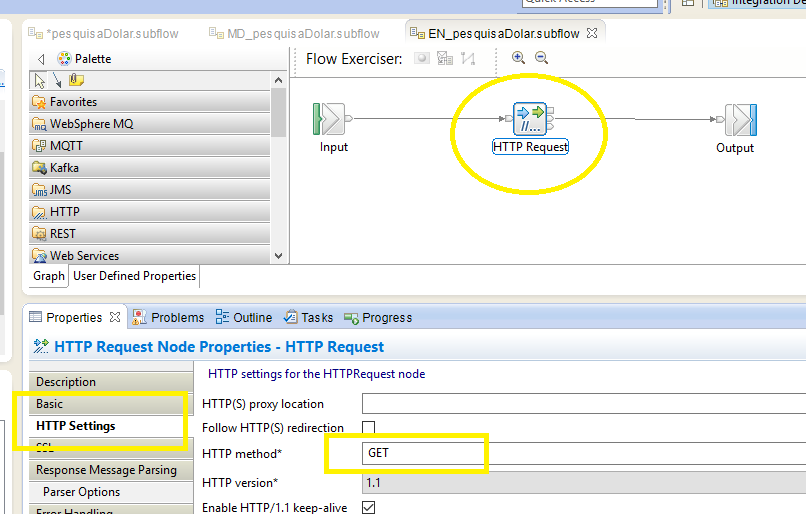
* + 1. Dublo clique no subflow ***EN\_pesquisaDolar.subflow***.



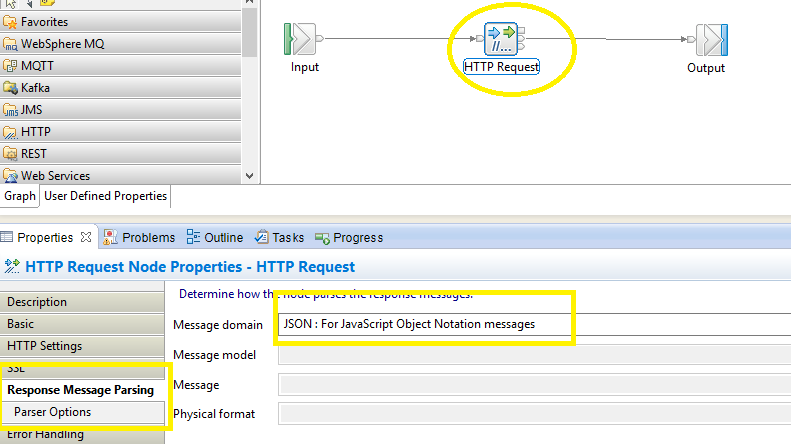
* + 1. Em ***Properties***, selecione ***Basic*** e preencha o campo ***Web Service URL*** com o valor ***http://localhost***.



* + 1. Em ***Properties***, selecione ***HTTP Settings*** e selecione o campo ***HTTP Method*** selecione ***GET***.



* + 1. Em ***Properties***, selecione ***Response Message Parsing*** e preencha o campo ***Message Domain*** selecione ***JSON: For JavaScript...***.



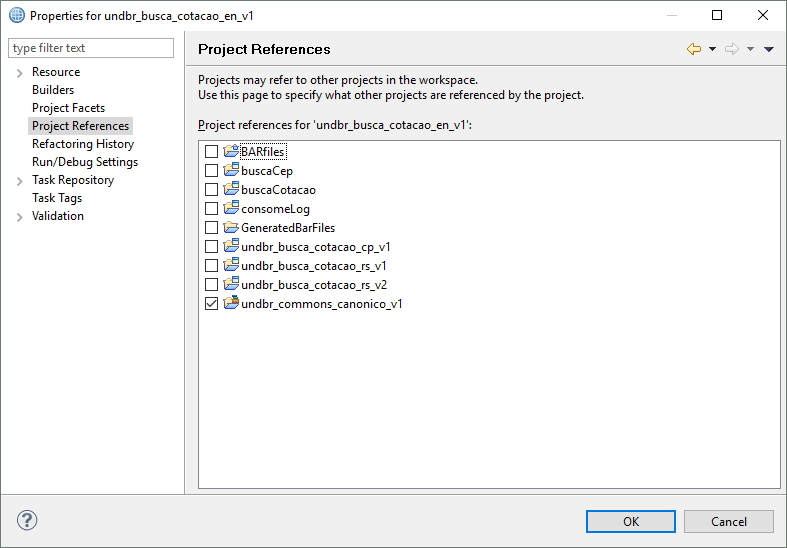
# EN - Vinculando o canônico ao projeto de enable.

Após nos certificamos através do teste anterior, que o serviço de integração está conseguindo realizar a invocação do serviço do legado e o respectivo legado retornando os dados ao barramento, podemos implementar os dois últimos mapeamentos de dados que consiste em:

Capturar o response do legado e converter para o canônico no projeto de ***EN*** e depois pegar do ***canônico*** e converter para o formato ***JSON*** que o consumidor está esperando, isso no projeto ***RS***.

Entretanto precisamos criar um vínculo desse projeto com o projeto canônico, uma vez que o xsd para estrutura de dados está contida no mesmo, para isso:

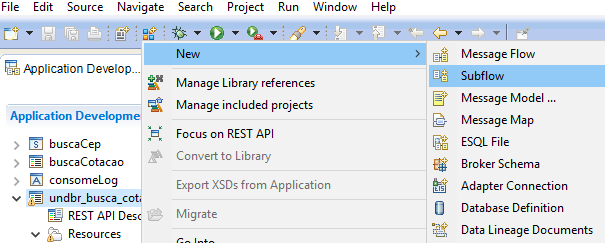
* + 1. Selecione o projeto que receberá o ***vínculo***.
    2. Clique em ***Properties***.
    3. Selecione ***Project References***.
    4. Selecione o ***projeto*** que será ***vinculado***.
    5. Clique em ***OK***.



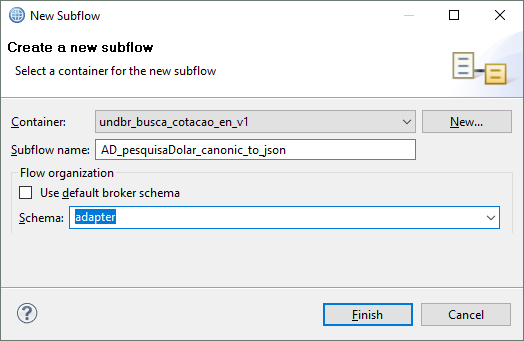
# EN - Implementando a camada adapter – Canônico para Json.

Na sequência precisamos construir mais um adapter, que ira receber os dados no modelo canônico e transformar o para a estrutura de dados que o legado está esperando, para isso:

* + 1. Selecione o ***projeto***.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

****

* + 1. Em Name digite: ***AD\_pesquisaDolar\_canonic\_to\_json.***



* + 1. Clique em ***Finish***.

|  |
| --- |
| **Implementando o código ESQL** |

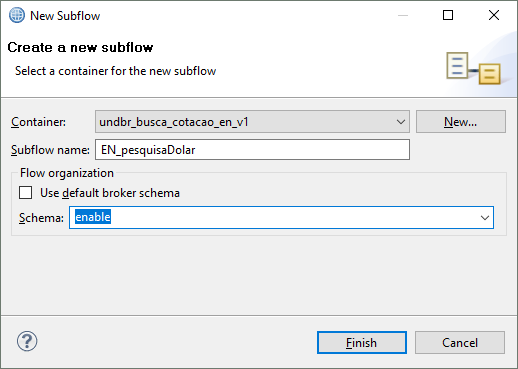
* + 1. Selecione o compute node e arraste para a área de trabalho.
    2. Em description digite: ***pesquisaDolar\_canonic\_to\_json***.
    3. Duplo clique o node para abrir o editor ***ESQL***.
    4. Cole o código abaixo entre ***begin*** e ***end***.

|  |
| --- |
| SET OutputRoot = InputRoot;  SET OutputLocalEnvironment = InputLocalEnvironment;    DECLARE nsDias NAMESPACE 'http:///www.unimed.com.br/Legado/CotacaoInput';  -------------------------------------------------------------------------------  -- Endpoint do provedor que teremos que acessar  -- http://api.bcb.gov.br/dados/serie/bcdata.sgs.1/dados/ultimos/10?formato=json  -------------------------------------------------------------------------------  -- Declarando variaveis  -------------------------------------------------------------------------------  DECLARE uri\_parte1 CHAR;  DECLARE dias CHAR;  DECLARE uri\_parte2 CHAR;  DECLARE endpoint CHAR;  -------------------------------------------------------------------------------  -- Concatenando a URL para compor o endpoint  -------------------------------------------------------------------------------  SET uri\_parte1 = 'http://api.bcb.gov.br/dados/serie/bcdata.sgs.1/dados/ultimos/';  SET dias = InputRoot.XMLNSC.nsDias:Cotacao.qtdeDias;  SET uri\_parte2 = '?formato=json';  SET endpoint = uri\_parte1 || dias || uri\_parte2;    -------------------------------------------------------------------------------  -- Setando o endpoint no node de HTTP Request  -------------------------------------------------------------------------------  SET OutputLocalEnvironment.Destination.HTTP.RequestURL = endpoint;  RETURN TRUE; |

# EN - Implementando a camada de enable – parte II.

Na sequência precisamos construir mais um adapter, que irá receber os dados de retorno do legado e transformar para o modelo canônico, para isso:

* + 1. Selecione o ***projeto***.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.
    4. Em Subflow Name: ***EN\_pesquisaDolar***.
    5. Selecione o schema: ***enable***.

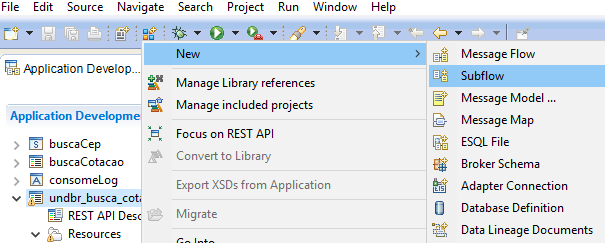


# EN - Implementando a camada adapter – JSON para Canônico.

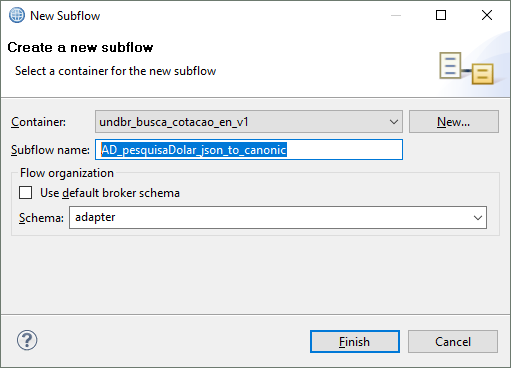
Temos agora que retornar os dados da requisição que o barramento realizou no legado, assim será necessário realizar o mapeamento de resposta, entenda que é a resposta do legado que devemos responder.

|  |
| --- |
| **Agora vamos construir nosso subflow de adaptação para a resposta.** |

* + 1. Selecione o projeto: ***undbr\_busca\_cotacao\_en\_v1***.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

****

* + 1. Em Name digite: ***AD\_pesquisaDolar\_json\_to\_canonic.***
    2. *Certifique que o Container selecionado é o serviço em implementação.*
    3. Deselecione a opção: ***Use default broker schema***.
    4. Em schema selecione o broker schema que criamos, nesse caso: ***adapter.***



* + 1. Clique em ***Finish***.

|  |
| --- |
| Vamos implementar o código ***ESQL***. |

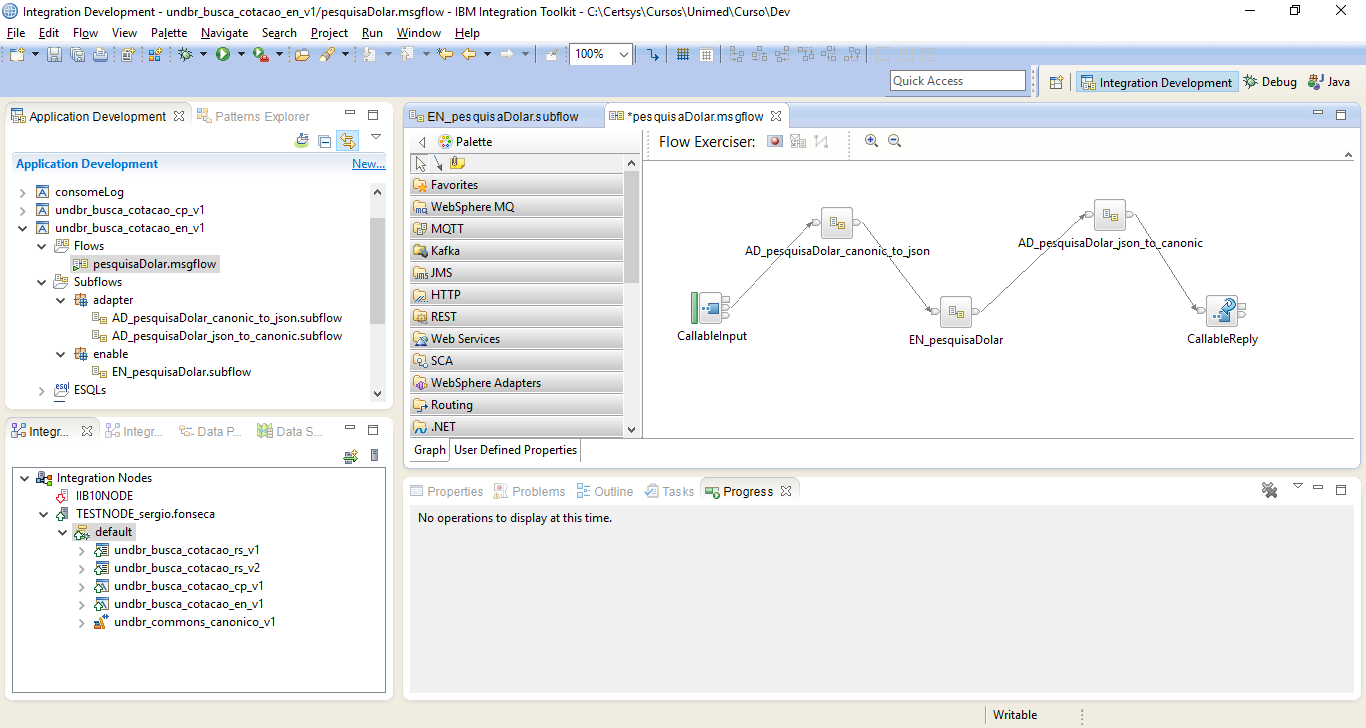
* + 1. Selecione o ***compute node***.
    2. Duplo clique no node para abrir o editor de ***ESQL***.



* + 1. Insira o ***ESQL*** abaixo, entre as tag ***begin*** e a tag ***end***.

|  |
| --- |
| DECLARE ns NAMESPACE 'http:///www.unimed.com.br/Legado/CotacaoOutput';    DECLARE I INTEGER CARDINALITY(InputRoot.JSON.Data.\*[]);  DECLARE J INTEGER 1;    WHILE J <= I DO  SET OutputRoot.XMLNSC.ns:CotacaoOutput.valores[J].data = InputRoot.JSON.Data.Item[J].data;  SET OutputRoot.XMLNSC.ns:CotacaoOutput.valores[J].valor = InputRoot.JSON.Data.Item[J].valor;  SET J = J + 1;  END WHILE;    RETURN TRUE; |

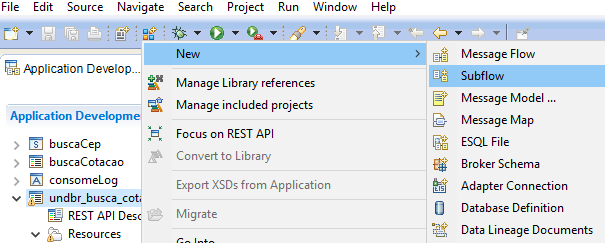
* + 1. Araste o Subflow do adapter para flow ***pesquisaDolar***.
    2. Faça as ligações entre os nodes.



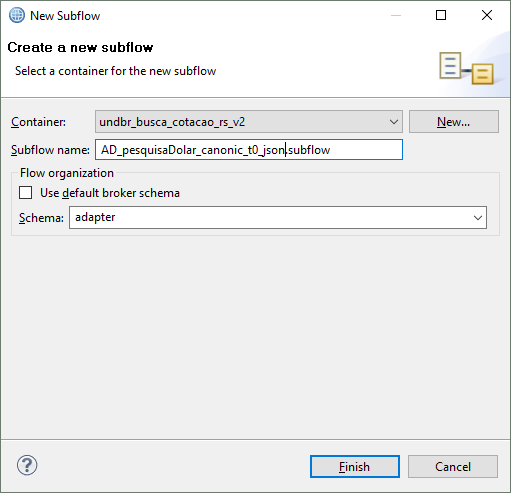
# RS - Implementando a camada adapter – Canônico para JSON.

Os dados já estão retornando, entretanto está no formato XML, precisamos agora capturar as informações em disponibilizar no formato JSON, para isso:

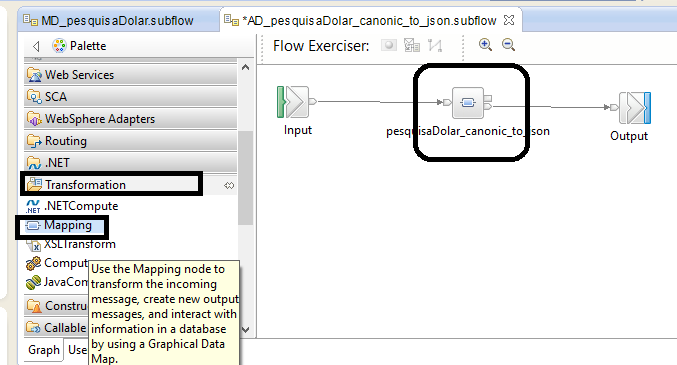
* + 1. Selecione o projeto: ***undbr\_busca\_cotacao\_en\_v1***.
    2. Clique em ***New***.
    3. Clique em ***Subflow***.

****

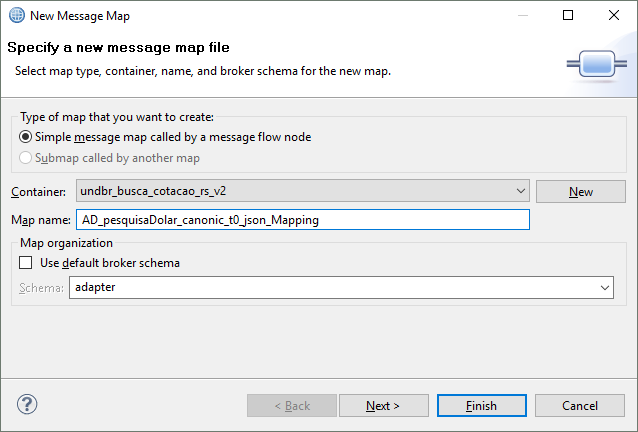
* + 1. Em Name digite: ***AD\_pesquisaDolar\_ canonic\_to\_ json.***
    2. *Certifique que o Container selecionado é o serviço em implementação.*
    3. Deselecione a opção: ***Use default broker schema***.
    4. Em schema selecione o broker schema que criamos, nesse caso: ***adapter.***



* + 1. No paleta de ***notes***, clique ***Transformation*** e selecione ***Mapping*** e arraste para a área de trabalho.



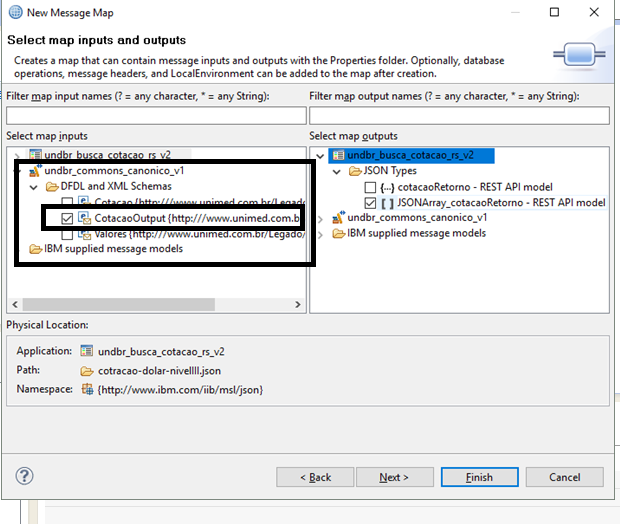
* + 1. Execute duplo clique em ***Mapping***, para abrir área de mapeamento de dados.



* + 1. Clique em ***Finish***.

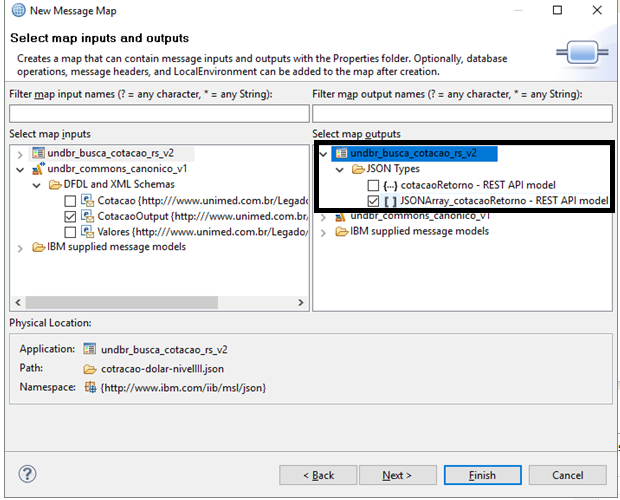
Será apresentada a tela para selecionar os artefatos para os mapeamentos de dados, ao lado esquerdo são os dados de entrada de dados para o fluxo de mapeamento, como estamos recebendo conforme o modelo canônico, devemos selecionar o xsd ***CotacaoOutput*** contido no mesmo, para isso:

* + 1. Expanda o projeto ***undbr\_commons\_canonico\_v1***.
    2. Selecione o artefato ***CotacaoOutput.***

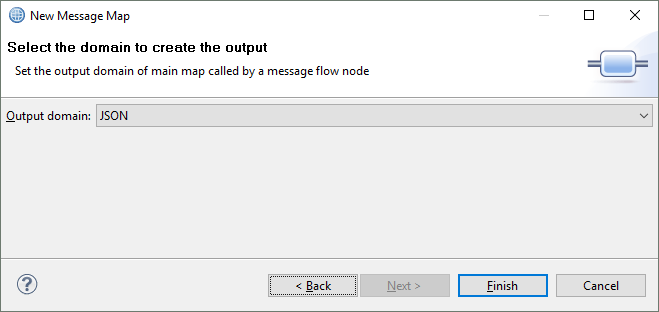


Ao lado direito é idem ao acima, porem são os artefatos para os dados de saída, devemos selecionar o artefato correspondente ao JSON que queremos devolver ao consumidor do barramento.

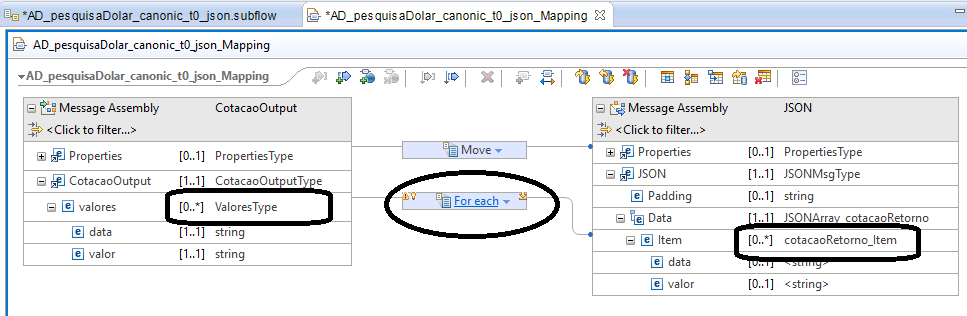
* + 1. Expanda o projeto ***undbr\_busca\_cotacao\_rs\_v2***.
    2. Selecione o artefato ***JSONArray\_cotacaoRetorno.***



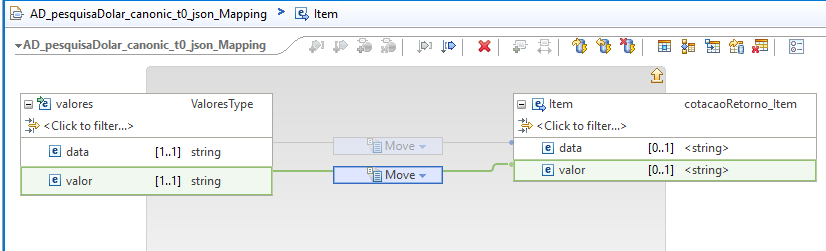
* + 1. Clique em ***Next,*** para visualizar o formato de retorno***.***
    2. Clique em ***Finish***.



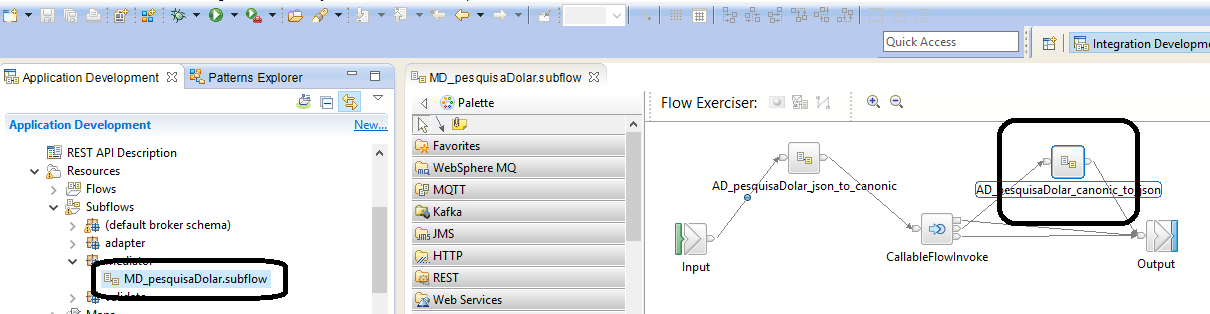
* + 1. Clique em ***Valores*** do lado esquerdo e arraste para ***Item*** do lado direito.
    2. Duplo clique em ***For each***, localizado no centro da ligação.



* + 1. Duplo clique em ***For each***, localizado no centro da ligação.
    2. Faça as ligações conforme abaixo.



* + 1. Salve o projeto e saída da tela de mapeamento.
    2. Selecione o subflow de mediator e insira o subflow de conversão do Canônico para o Json.

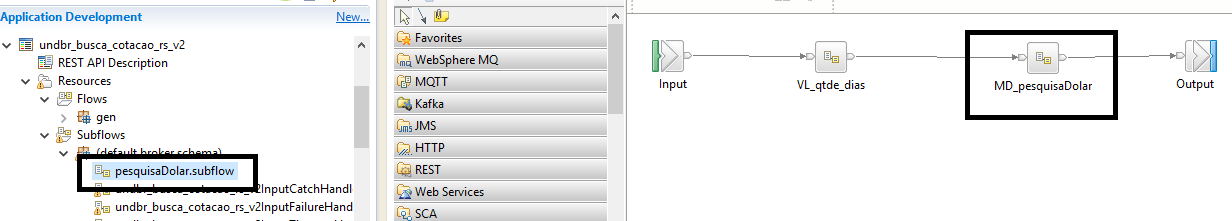


# Ligações finais.

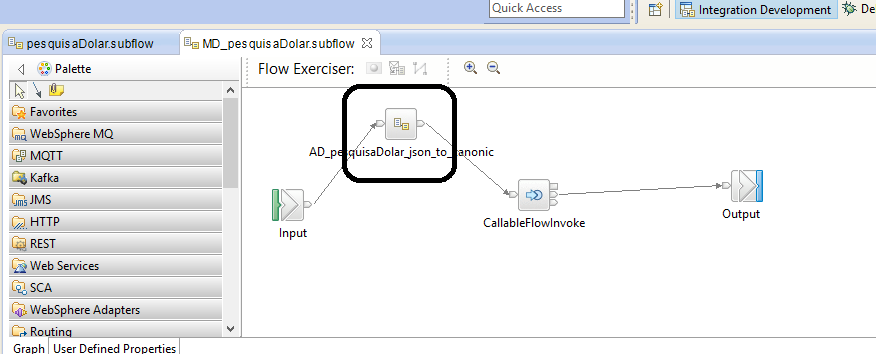
Concluímos a implementação até a chamada do serviço do legado, vamos revisar e realizar os ajustes finos, o que nos permitirá realizar um teste completo no serviço no sentido de realizar uma requisição.

Revisando:

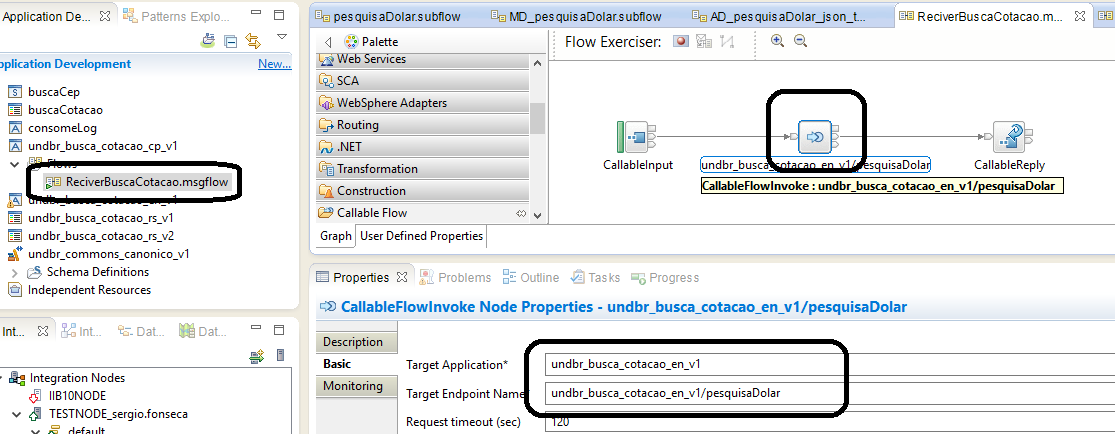
* + 1. No projeto corporativo: ***undbr\_busca\_cotacao\_rs\_v2.***
    2. Selecione o subflow: ***pesquisaDolar*** e abra a área de trabalho.
    3. Selecione o sublow: ***MD\_pesquisaDolar*** e araste para a área de trabalho.
    4. Realize as ligações entre os nodes.



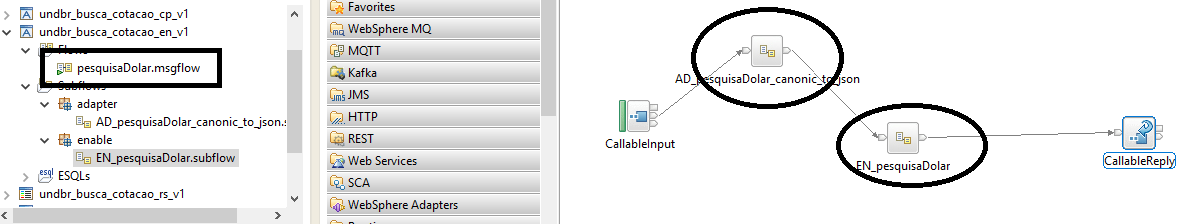
* + 1. No projeto corporativo: ***undbr\_busca\_cotacao\_rs\_v2***.
    2. Selecione o subflow: ***MD\_pesquisaDolar*** e abra a área de trabalho.
    3. Arraste e solte o Subflow ***AD\_pesquisaDolar\_json\_to\_canonic***.
    4. Realize as ligações entre os nodes.



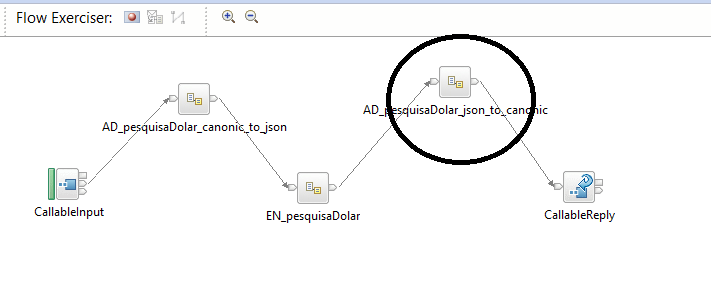
* + 1. No projeto corporativo: ***undbr\_busca\_cotacao\_cp\_v1***.
    2. Selecione o flow: ***reciverBuscaCotacao*** e abra a área de trabalho.
    3. Na paleta de nodes, clique em ***Callable Flow***.
    4. Arraste e solte na área de trabalho os node ***CallableFlowInvoke*** epreencha os seguintes dados:
       1. Target Application: ***undbr\_busca\_cotacao\_en\_v1***.
       2. Target Endpoint Name: ***undbr\_busca\_cotacao\_en\_v1/pesquisaDolar***.



* + 1. No projeto corporativo: ***undbr\_busca\_cotacao\_en\_v1***.
    2. Selecione o flow: ***pesquisaDolar*** e abra a área de trabalho.
    3. Arraste e solte o Subflow ***AD\_pesquisaDolar\_canonic\_to\_json***.
    4. Arraste e solte o Subflow ***EN\_pesquisaDolar.***
    5. Realize as ligações entre os nodes.



* + 1. No projeto corporativo: ***undbr\_busca\_cotacao\_en\_v1***.
    2. Selecione o flow: ***pesquisaDolar*** e abra a área de trabalho.
    3. Arraste e solte o Subflow ***AD\_pesquisaDolar\_json\_to\_canonic***.
    4. Arraste e solte o Subflow ***EN\_pesquisaDolar.***



* + 1. Realize as ligações entre os nodes.

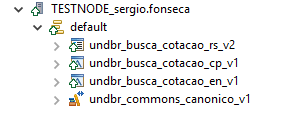
# Realizando deploy.

Para realizarmos o deploy precisamos entender as dependências dos projetos, no nosso projeto, o fluxo de integração inicia-se no projeto ***RS***, que invoca o projeto ***CP*** que por sua vez invoca o projeto ***EN***.

Portanto o deploy do projeto será em ordem inversa a mencionada acima, porque para que ocorra o deploy precisa que as dependências já estejam disponíveis no servidor.

O raciocínio é o seguinte: primeira deploy dos projetos que não tem dependências normalmente os projetos comuns e depois os demais projetos, na seguinte ordem:

1. undbr\_commons\_canonico\_v1
2. undbr\_busca\_cotacao\_en\_v1
3. undbr\_busca\_cotacao\_cp\_v1
4. undbr\_busca\_cotacao\_rs\_v2



|  |
| --- |
| ***Observação***:  Faça um teste para validar o fluxo, via ***soapui***, mas antes habilite o modo de debug e veja a entrada dos dados e a passagem dos mesmo entre os fluxos. |