

** TallerIntegrationServerPranica\_ver1\_0**

**Abril 2016**

TABLA DE CONTENIDO

[TABLA DE CONTENIDO 2](#_Toc448347486)

[Paquete 3](#_Toc448347487)

[Servicio Flow: Primer Servicio - escribirAlLog 5](#_Toc448347488)

[Flow Services – BRANCH 11](#_Toc448347489)

[Flow Services – LOOP 20](#_Toc448347490)

[Flow Service – SEQUENCE 27](#_Toc448347491)

[Publicar Servicios - Servicio Web: PROVIDER 31](#_Toc448347492)

[Enviar mensaje error - Servicio Web 37](#_Toc448347493)

[Adaptador 40](#_Toc448347494)

[Tabla de Base de Datos 43](#_Toc448347495)

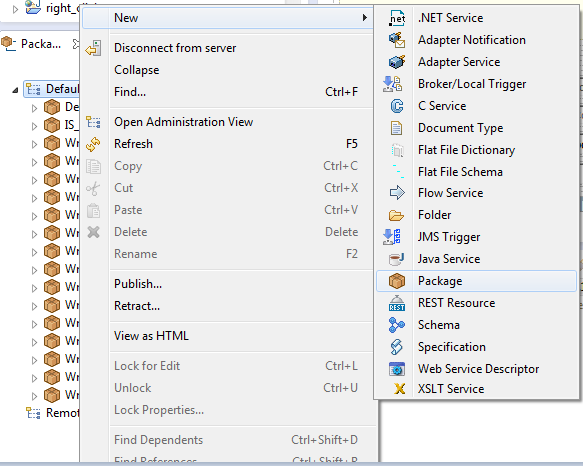
[Servicio Consultar Películas 44](#_Toc448347496)

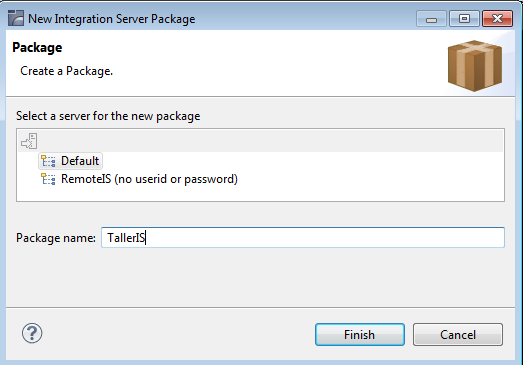
Paquete

**Crear un paquete o árbol de trabajo del proyecto.**

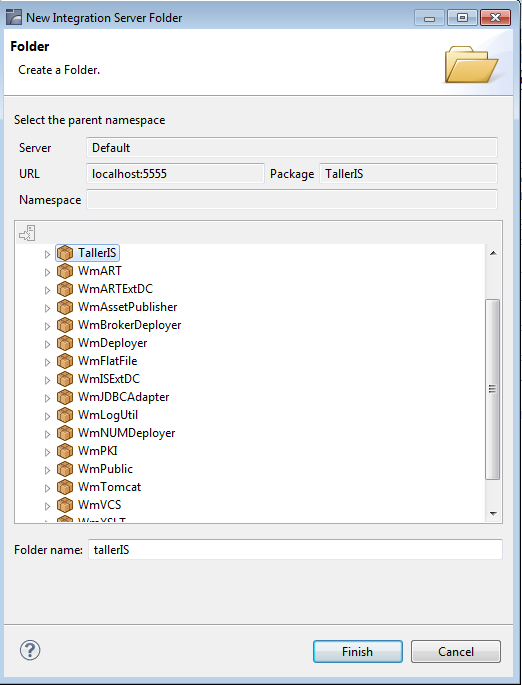
* TallerIS (Paquete)
* tallerIS (Carpeta)
  + **adapters:** Carpeta - adaptadores de conexión a BD
  + **docs**: Carpeta - tipo de documentos del proyecto
  + **services**: Carpeta – servicios del proyecto
  + **utils**: Carpeta de servicios utilitarios
  + **ws**
    - **consumer:** Carpeta – consumidor de Servicios Web
    - **provider:** Carpeta – proveedor de Servicios Web

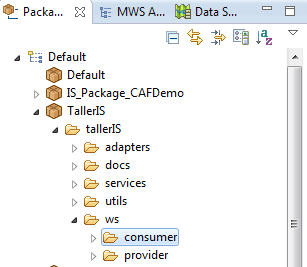
1. **Crear el paquete TallerIS**





1. **Crear carpetas del paquete**



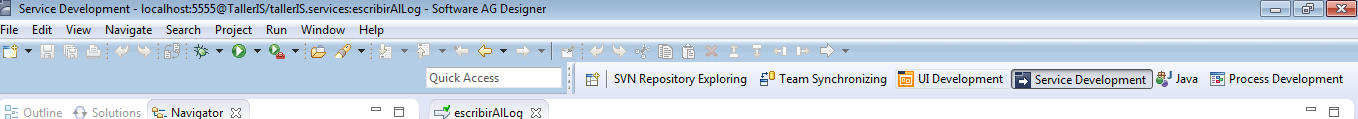
**Resultado**

Servicio Flow: Primer Servicio - escribirAlLog

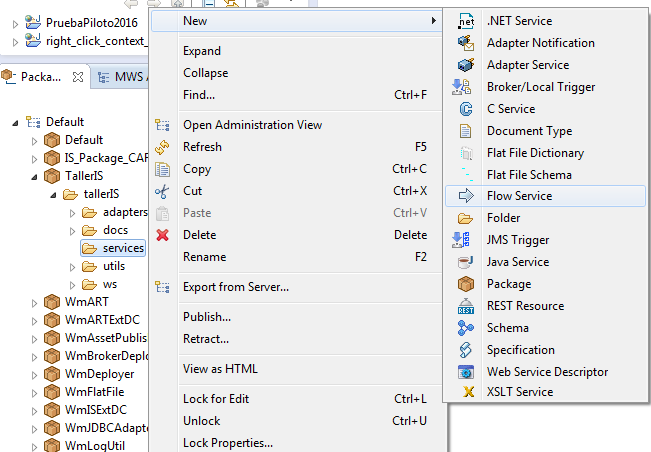
**Primer servicio: manipulación de datos de entrada y salida**

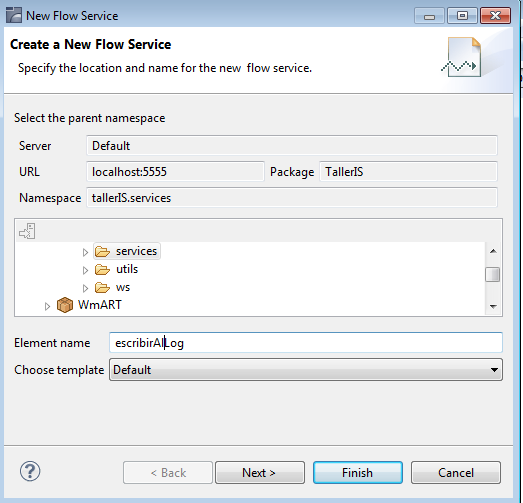
* Crear un servicio (Flow Service) de escritura a un archivo log

1. **Seleccionar la perspectiva Service Development**

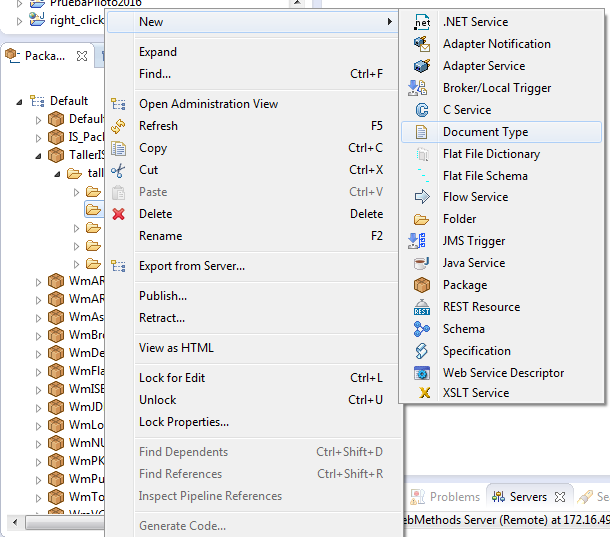


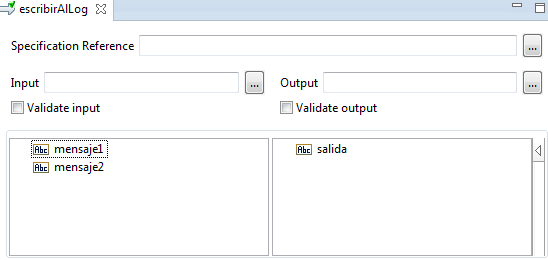
1. **Crear un Flow Service : escribirAlLog**



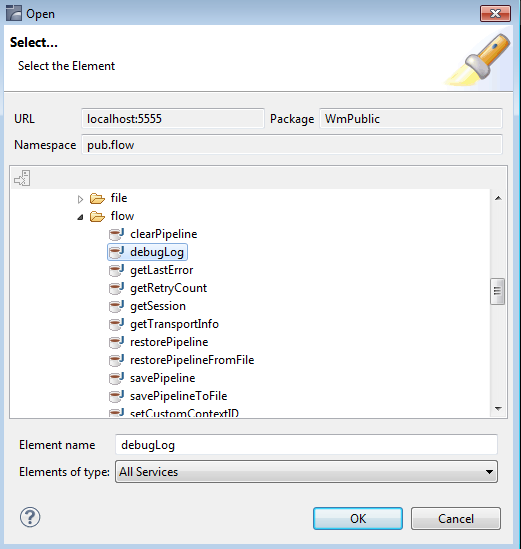


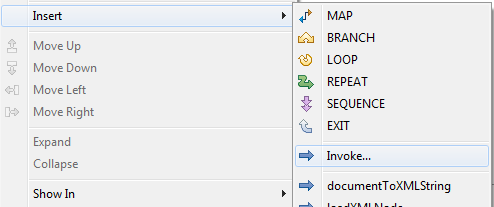
1. **Agregar parámetros de entrada y salida**



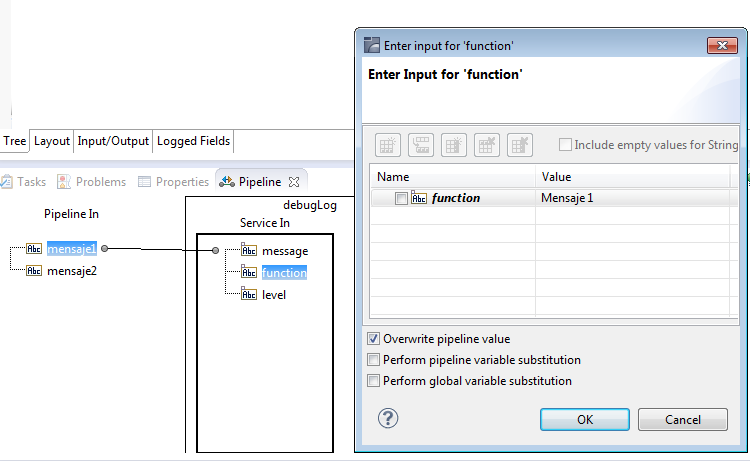


1. **Agregar el servicio WmPublic.pub.flo.debugLog:** Permite escribir a un archivo log

****

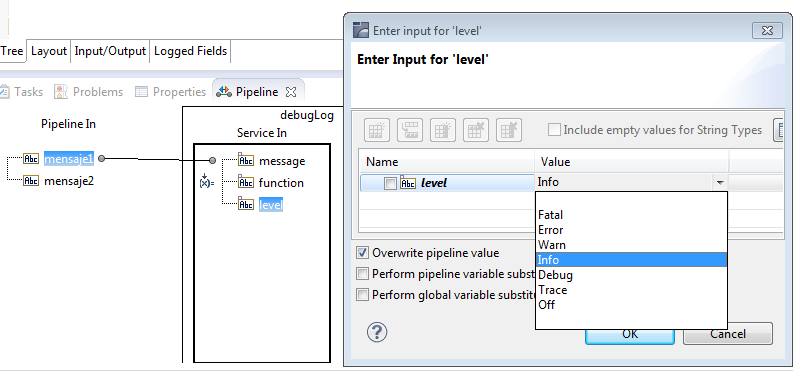
****

1. **Mapear (parsear) valores mensaje1 - al servicio debugLog**

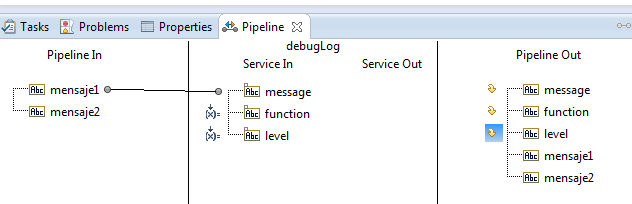


* Mapear nivel del mensaje log (info, error, debug, …)

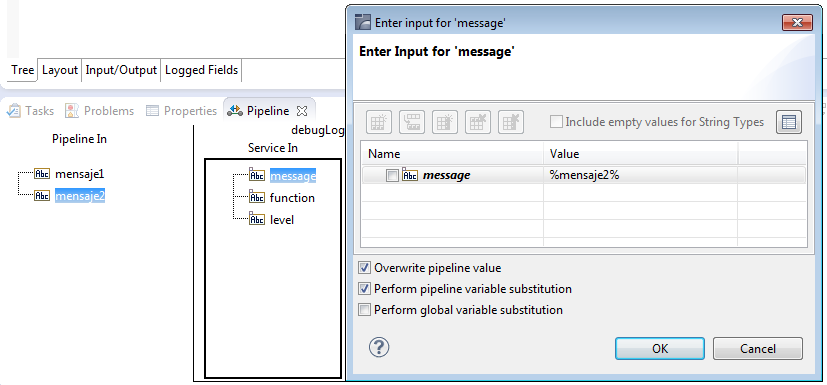
**Seleccionar el tipo de mensaje a escribir (Info, error, Fatal, Warning, etc)**



**Dropear datos ya procesados. Limpia las variables del Pipeline (contenido de información del IS)**

****

1. **Mapear valores mensaje 2 - al servicio debugLog** 
   * %mensaje2% => Imprimir el valor de la variable

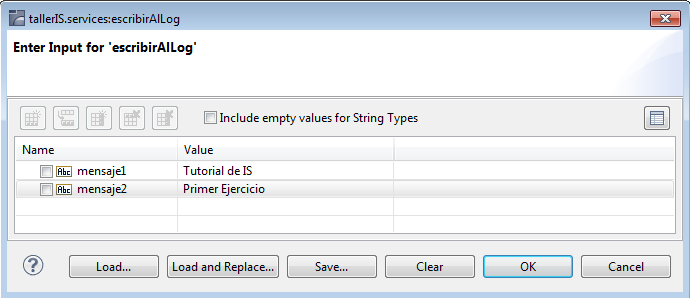


**Nota**: Seleccionar: Ovewrite pipeline value / Perform pipeline variable substitution (Al colocar variable entre % implica sustituir el valor de una variable por el dato actual en el Pipeline)

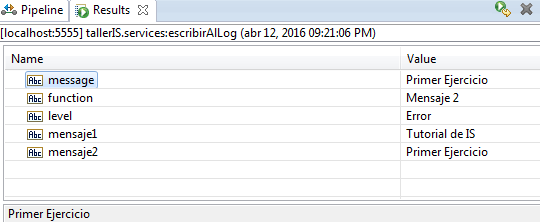
1. **Correr el servicio imprimirAlLog**

****

**Llenar variables de entrada**



**Resultado salida en el Designer**

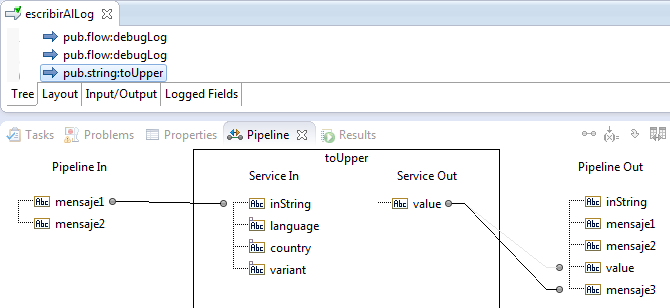


**Nota: level: Error (nivel de impresión al log se selecciono Error )**

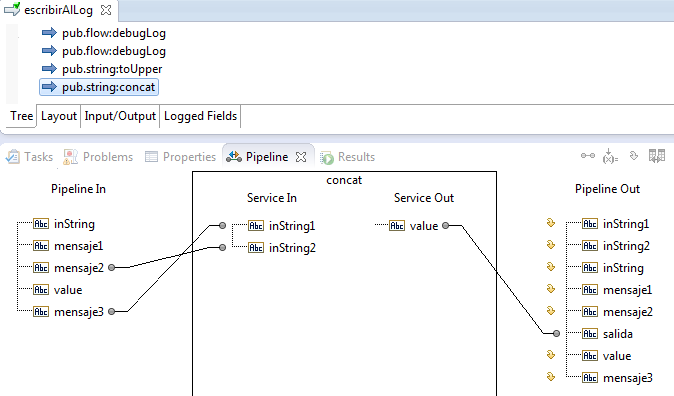
**Resultado salida en el Designer**



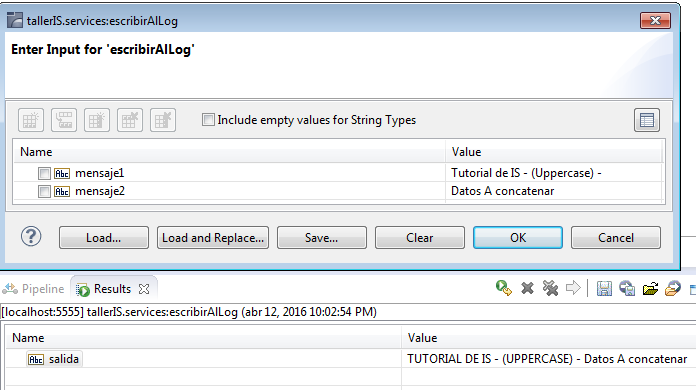
1. **Maximizar un texto y almacenarlo en variable** (WmPublic.pub.string.toUpper -> mensaje3)

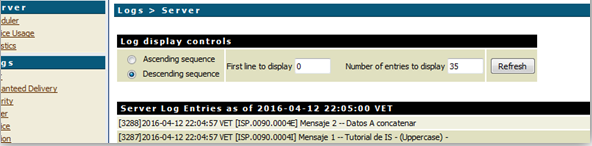


1. **Concatenar mensajes y enviarlo a la salida:** (WmPublic.pub.string.concat -> salida)



1. **Probar servicio**

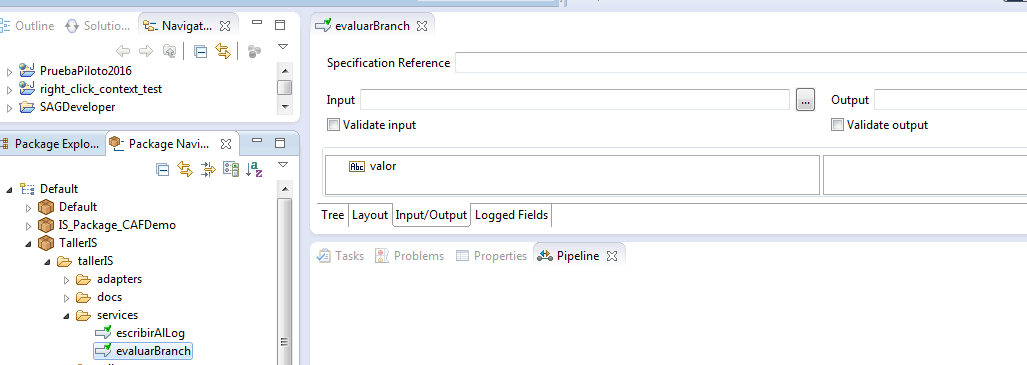




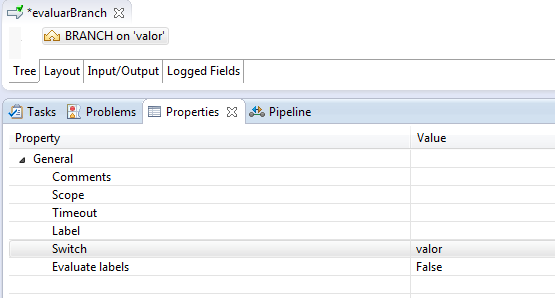
Flow Services – BRANCH

**Probar contenido de variables y niveles lógicos asociados a variables**

1. **Crear el Flow Service: evaluarBranch. Con variable de entrada valor**

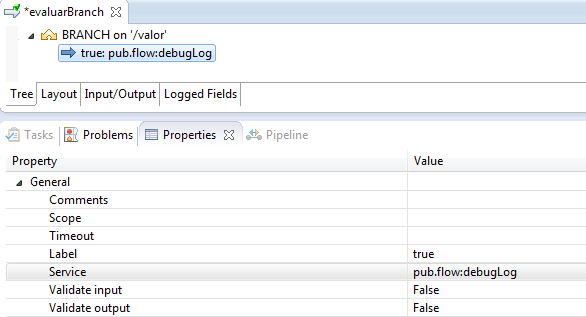


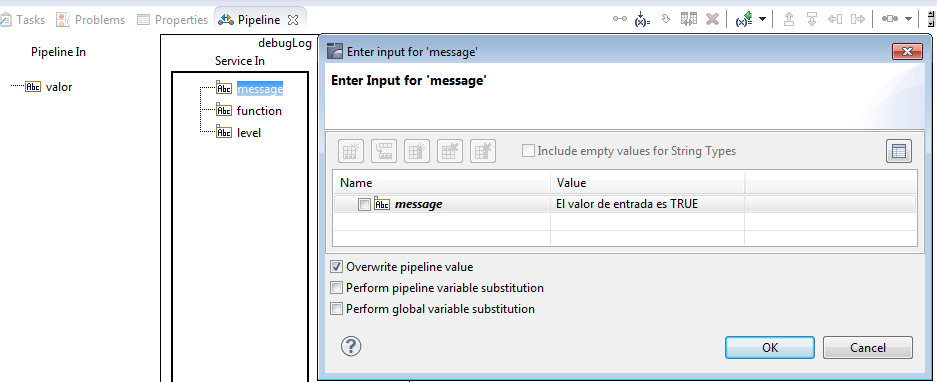
1. **Agregar un Branch** 
   * Switch: Variable a evaluar (valor)
   * Evaluate labels: False => El branch evalua la variable de entrada (valor)



1. **Agregar un servicio pub.flow.debugLog** 
   * Label: true => valor a evaluar de la variable de entrada “valor”

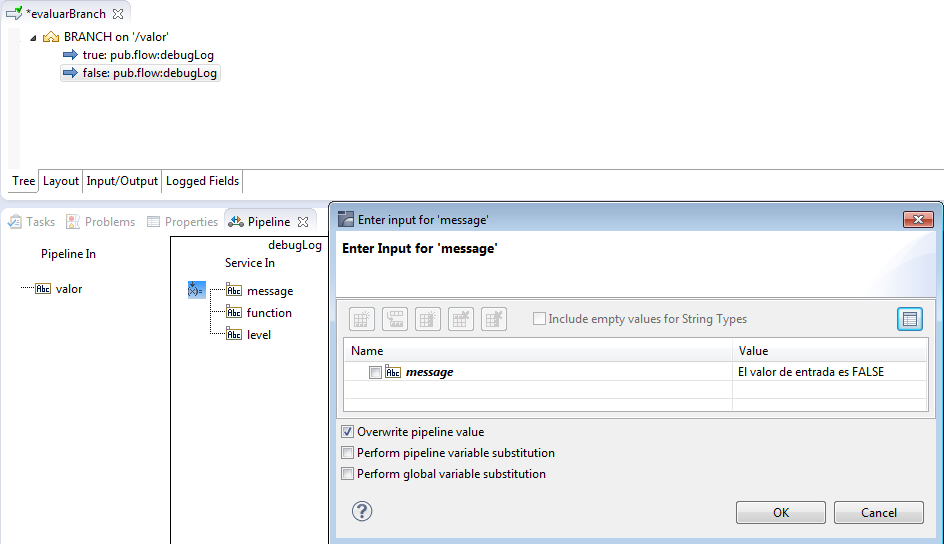
*If (valor == true) then*





1. **Agregar un servicio pub.flow.debugLog** 
   * Label: false => valor a evaluar de la variable de entrada “valor”

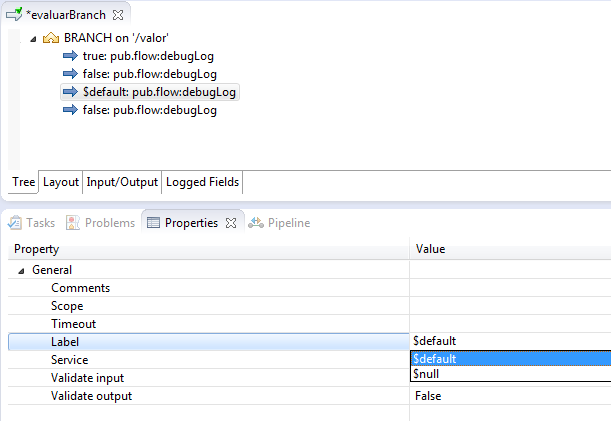
*If (valor == false) then*

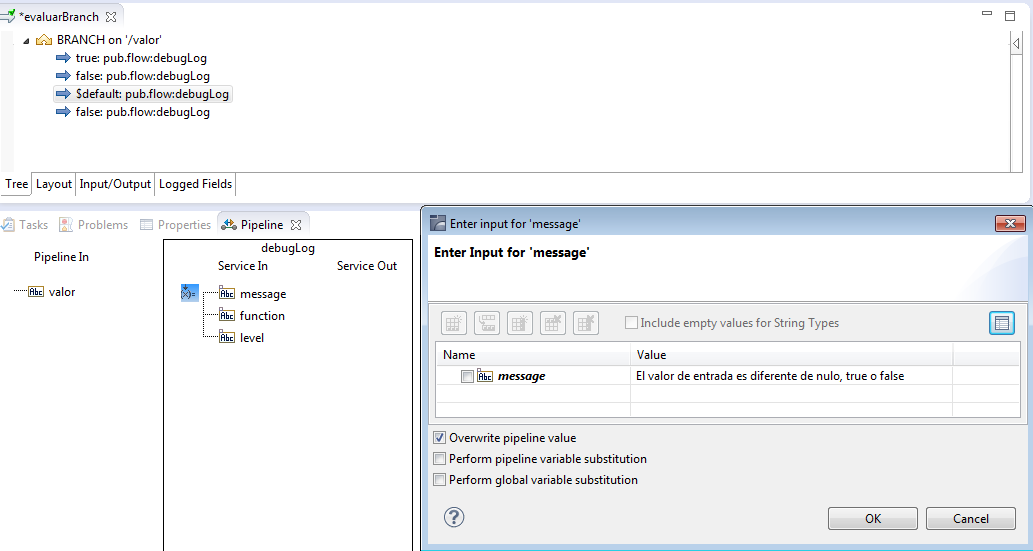


1. **Agregar un servicio pub.flow.debugLog** 
   * Label: cualquier valor diferente de nulo, true o false

=> valor a evaluar de la variable de entrada “valor”

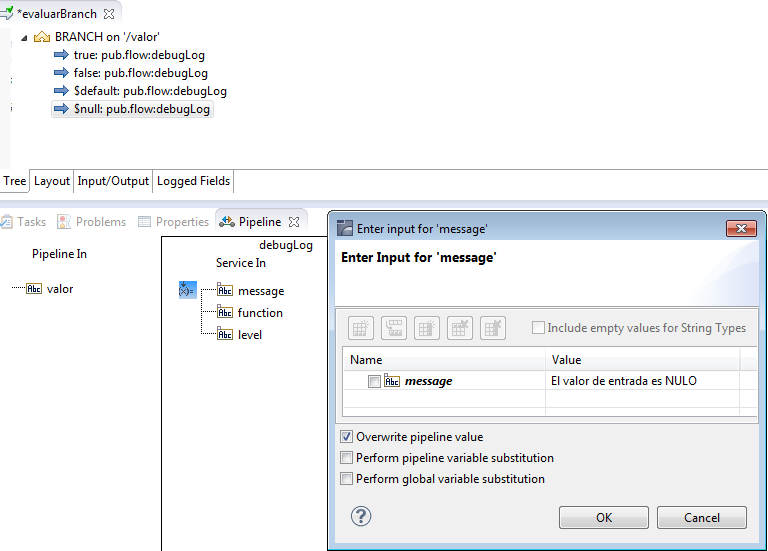
*If (valor != null or valor != true or valor != false) then*



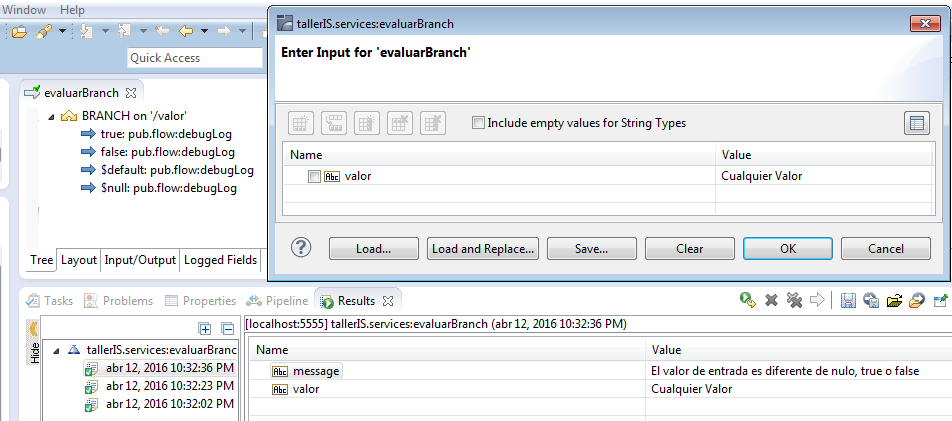


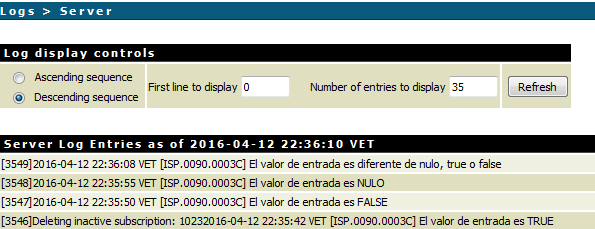
1. **Agregar un servicio pub.flow.debugLog** 
   * Label: valor null => valor a evaluar de la variable de entrada “valor” => Vacio sin valor

*If (valor == null) then*

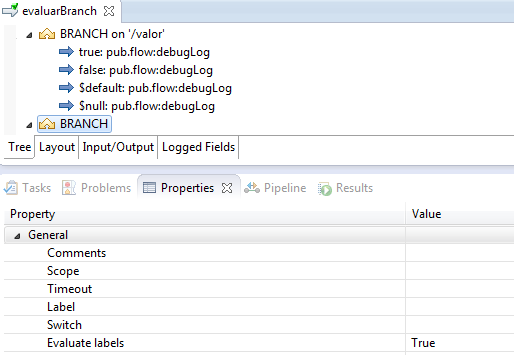


1. **Correr los cuatro casos** 
   * valor = true / false / “*vacio*” / “Cualquier Valor”



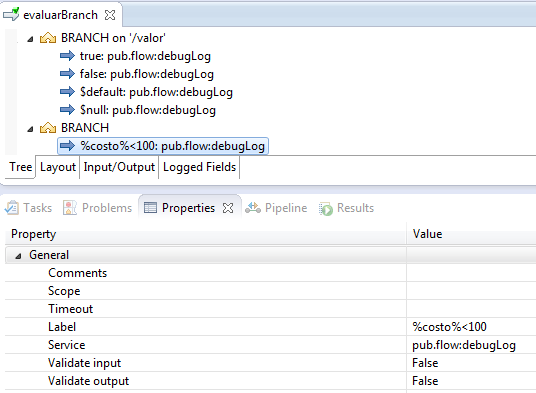


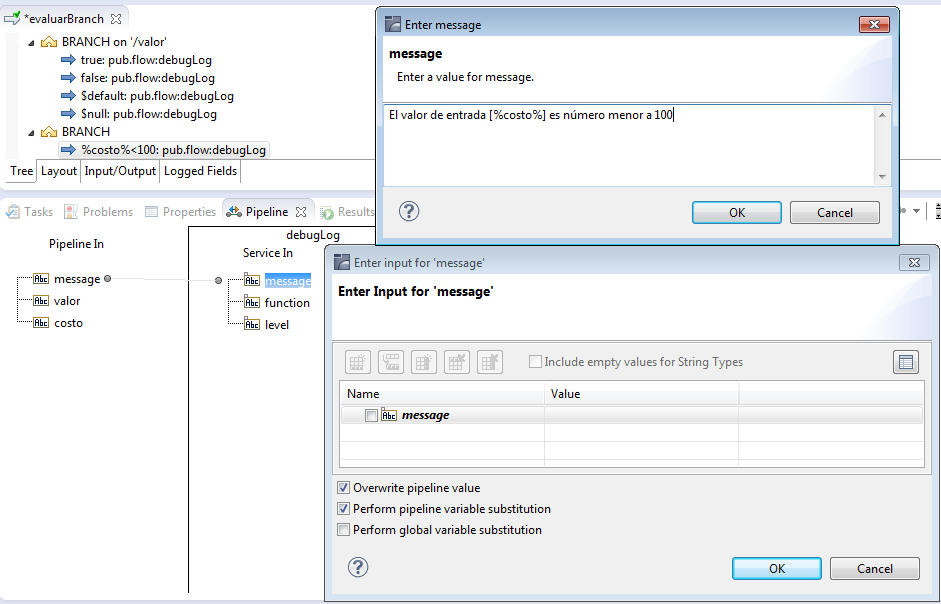
1. **Agregar un segundo Branch** 
   * Switch: vacio
   * Evaluate labels: True => El branch no evalúa la variable de entrada sino los labels dentro del branch (costo)



1. **Agregar un servicio pub.flow.debugLog** 
   * costo => valor a evaluar de la variable de entrada “costo”

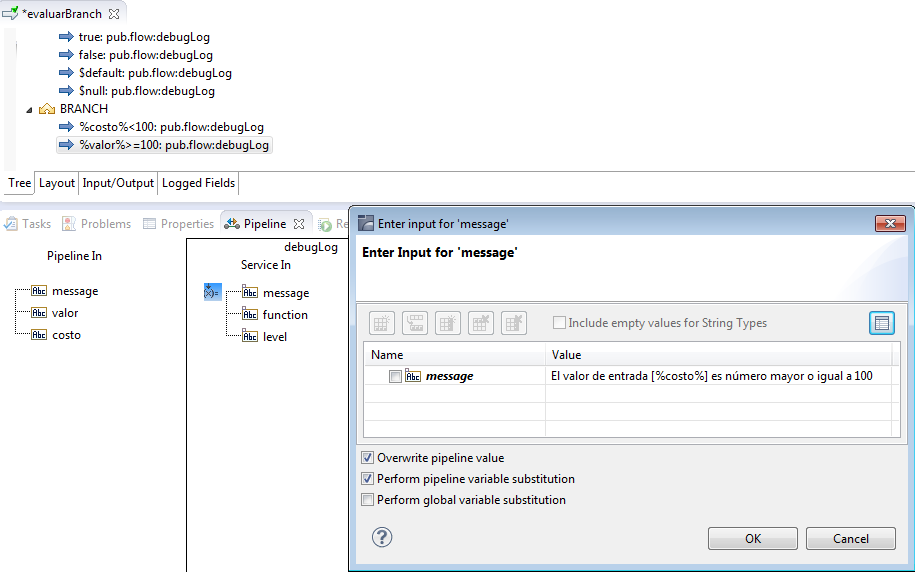
*If (costo < 100) then*





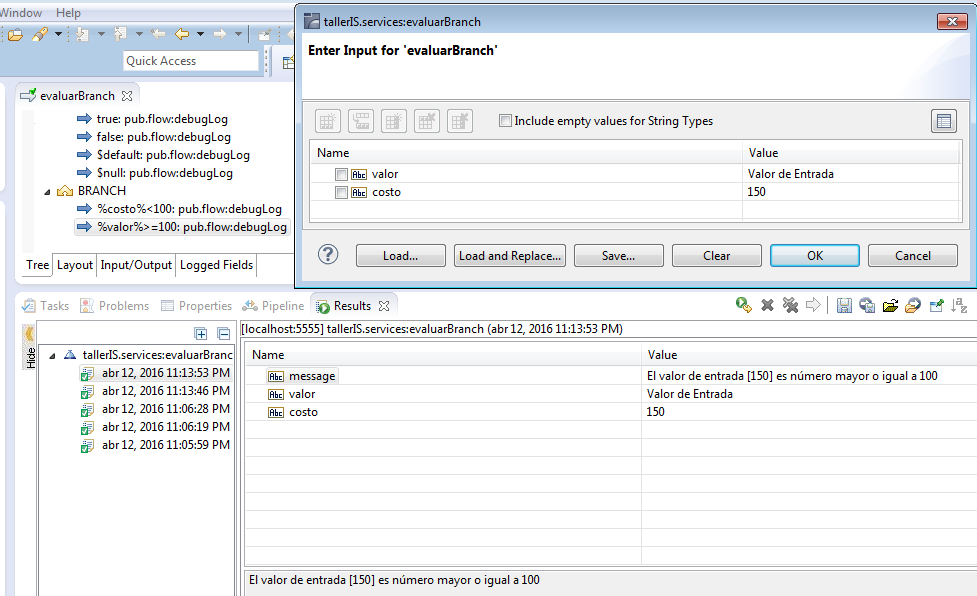
1. **Agregar un servicio pub.flow.debugLog** 
   * costo => valor a evaluar de la variable de entrada “costo”

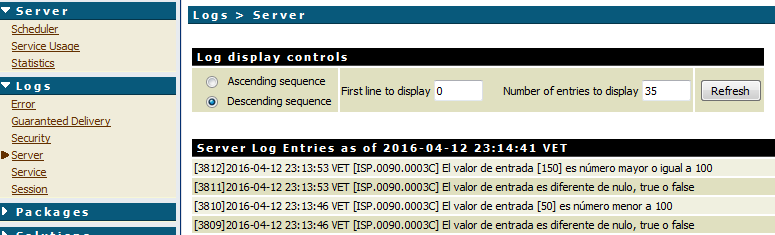
*If (costo >= 100) then*



**Nota**: Seleccionar: Ovewrite pipeline value / Perform pipeline variable substitution

1. **Probar los dos casos para el nuevo branch**
   * costo = 50
   * costo = 150

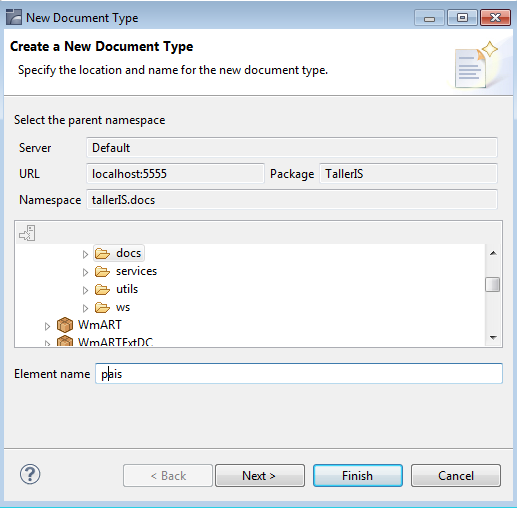


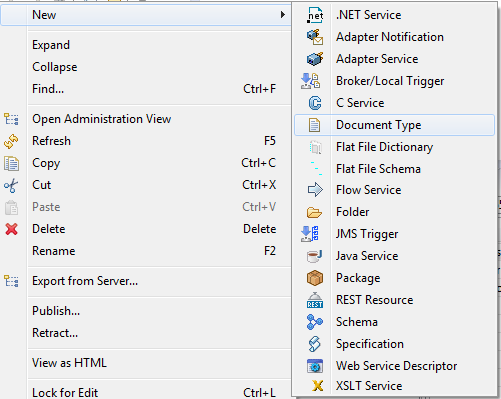


Flow Services – LOOP

**Probar recorrido de listas**

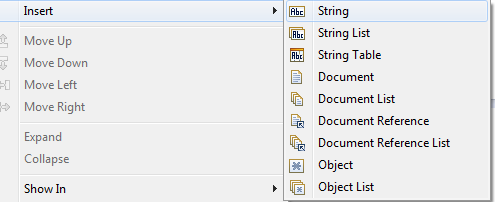
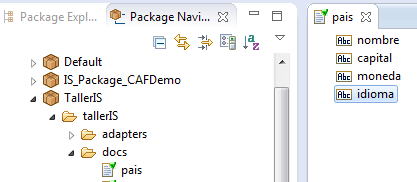
1. **Crear documento pais (Document Type)**



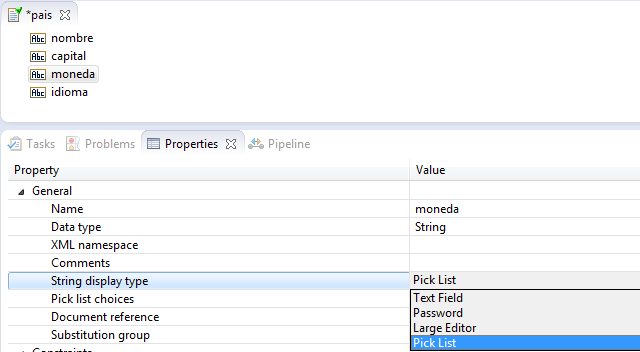


**Agregar variables tipo string al documento país creado**

* + nombre
  + capital
  + moneda
  + idioma

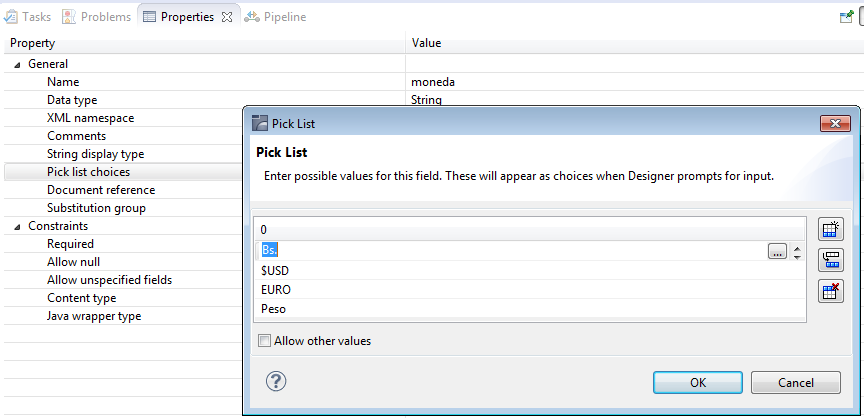


**Definir valores de la variable moneda (lista de valores – Pick List)**



**Definir valores (Pick List choices)**

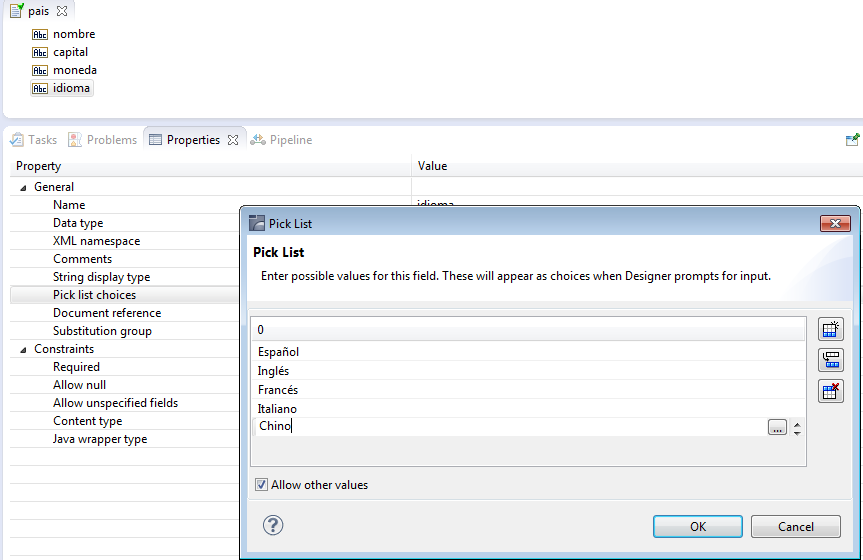
* + Bs.
  + $USD
  + EURO
  + PESO



Nota: No seleccionar Allow other values - Permitir solo esos valores

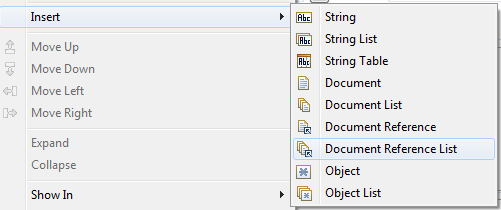
**Definir valores de variable idioma (lista de valores – Pick List)**

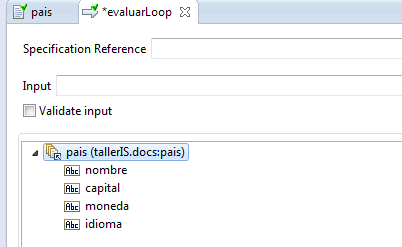
* + Español
  + Inglés
  + Francés
  + Italiano
  + Chino



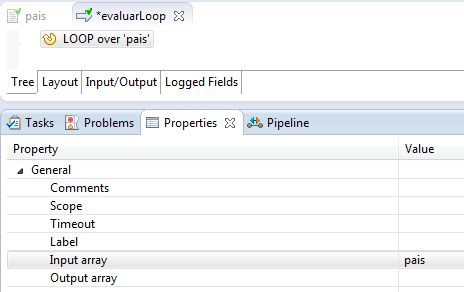
Nota: Seleccionar Allow other values - Permitir otos valores

1. **Crear Flow Service: evaluarLoop**
2. **Agregar una lista de documentos “país” – (Document Reference List) de nombre “país”**

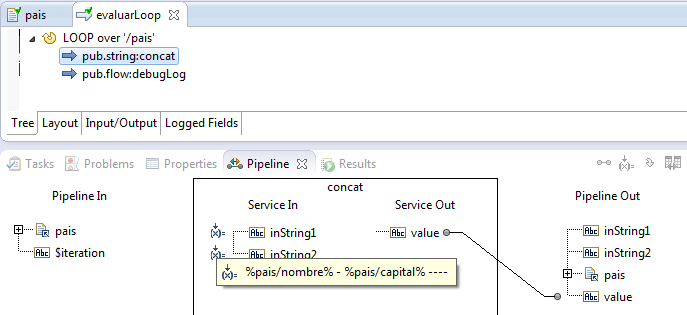




1. **Agregar un Loop - pasar lista o arreglo de valores (Input Array – país)**

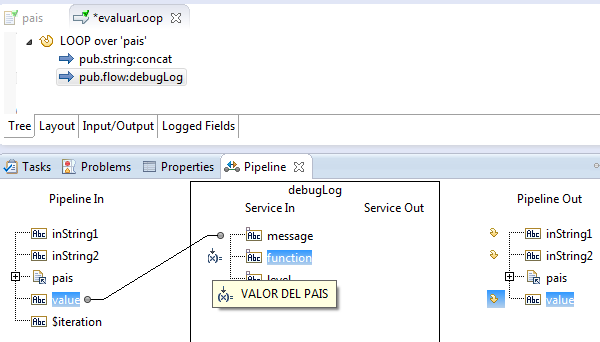


1. **Concatenar los datos dentro del LOOP**
   * inString1: %pais/nombre% - % pais/capital% ----
   * inString2: % pais/moneda% - % pais/idioma%



Nota: Copiar la variable de entrada (ej: nombre) y pegarla en intString1 (país/nombre) entre %% para tomar el valor de la variable

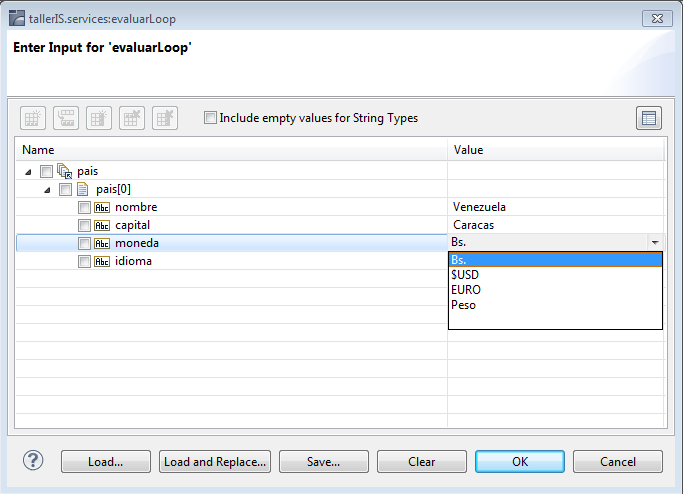
1. **Imprimir datos concatenados –** Eliminar del pipeline (drop) variables utilizadas



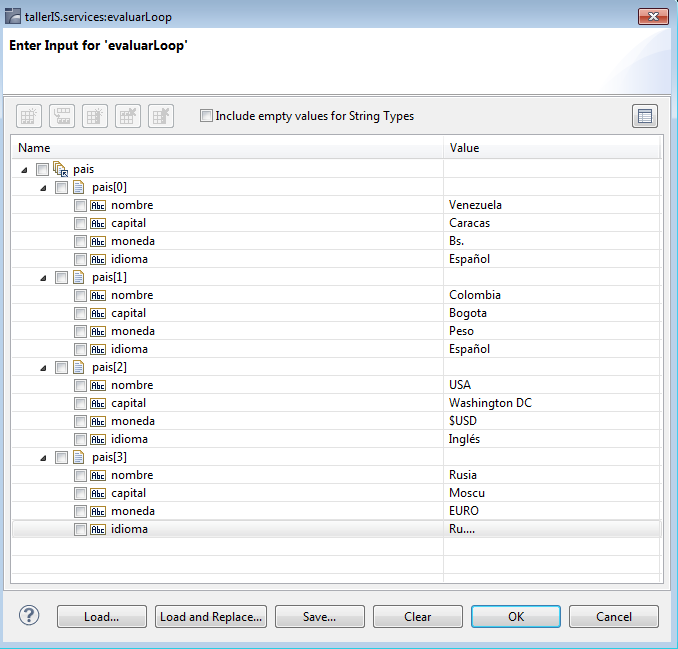
* + Correr y probar servicio: Llenar lista de valores

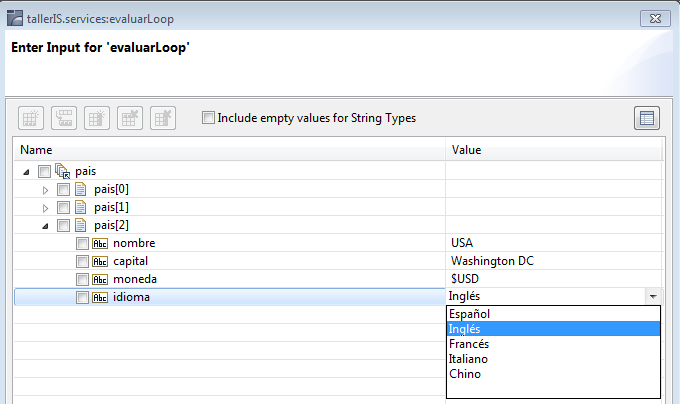
1. **Probar el servicio**

**Agregar un registro del documento y llenar los valores de entrada**

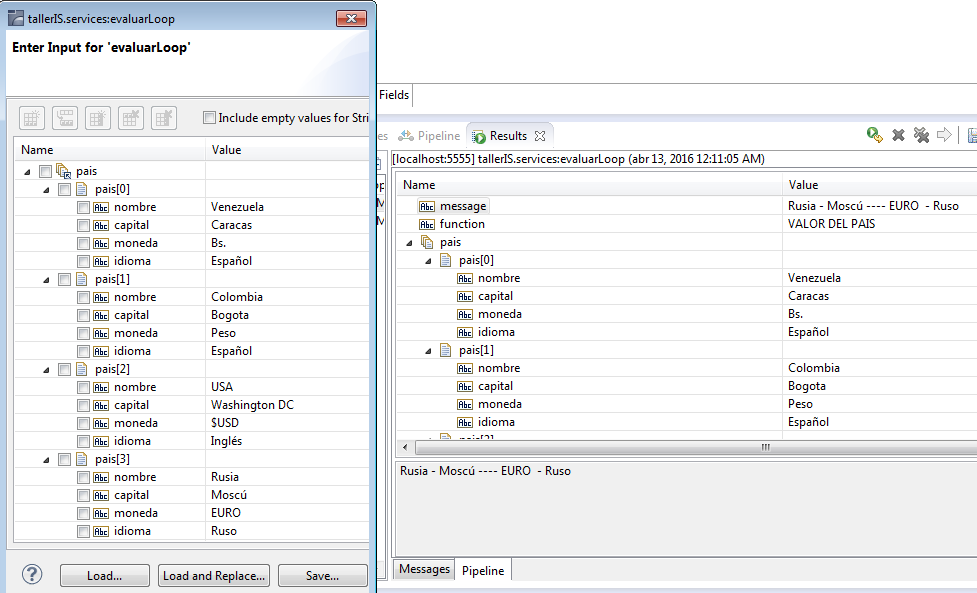


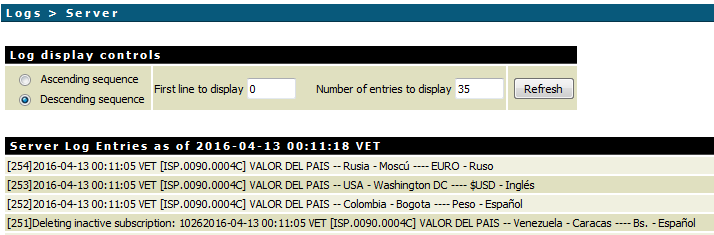
**Agregar cuatro registros del documento y llenar los valores de entrada**





**Resultado Salida:**

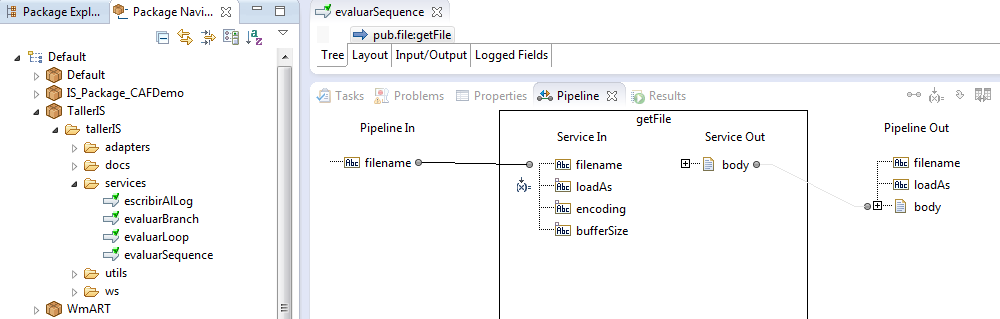
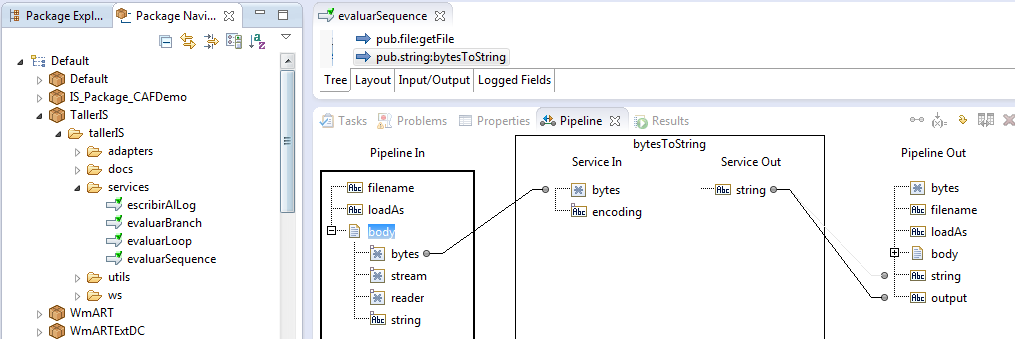


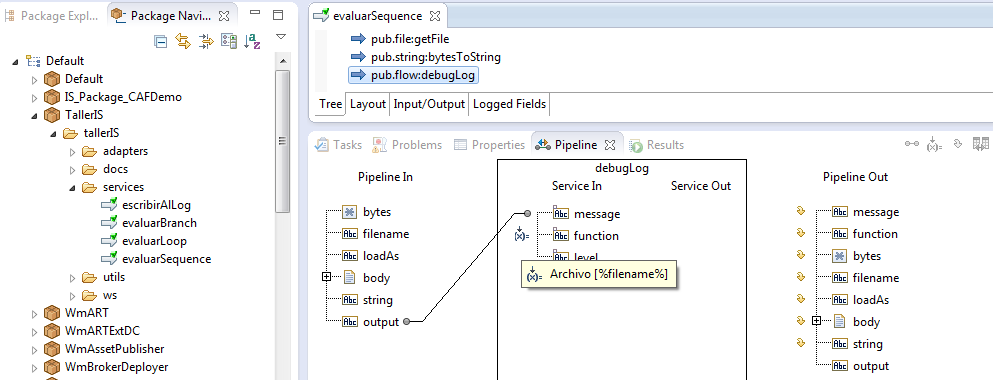


Nota: El campo idioma: Ruso no estaba en la lista original de valores (se agregó el check que permite incluir otros valores de la lista para este campo o variable)

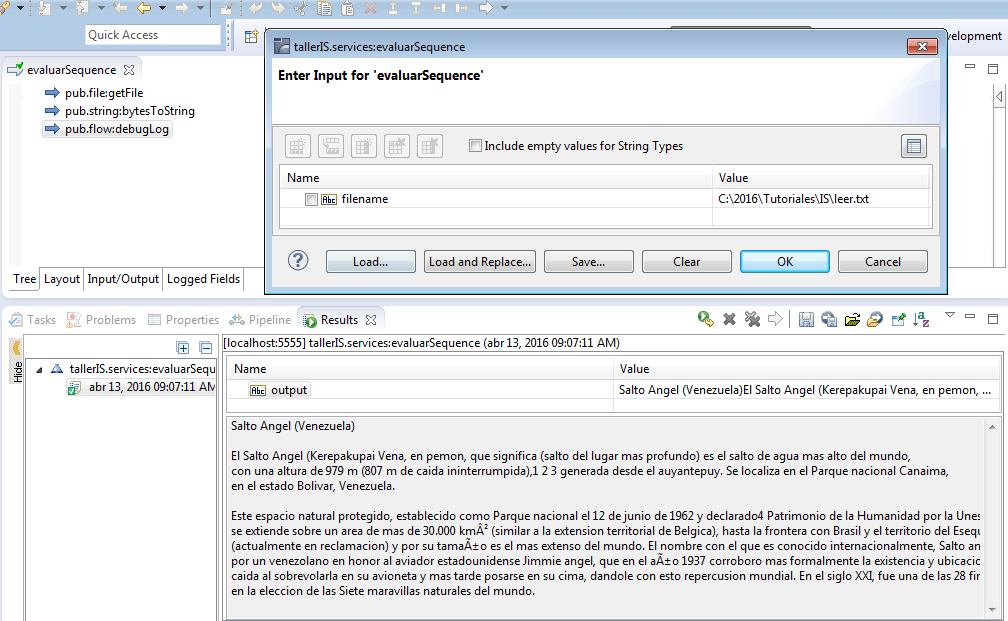
Flow Service – SEQUENCE

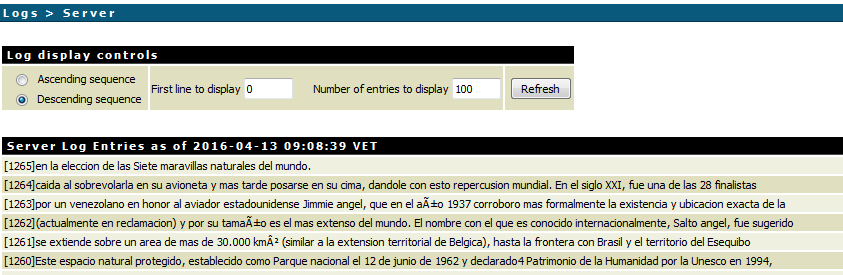
**Lógica try/catch para servicios Flow. Permite enviar exception.**

1. **Crear un Flow Service: evaluarSequence**
   * Entrada de datos:
   * Input: filename (String)
   * Output: output (String)
2. **Agregar el servicio pub.file:getFile (Leer un archivo) –** Mapear datos
3. **Agregar el servicio pub.string:bytesToString (Convertir datos bytes a String)**
4. **Agregar servicio pub.flow:debugLog (Imprimir resultado en archivo log)**

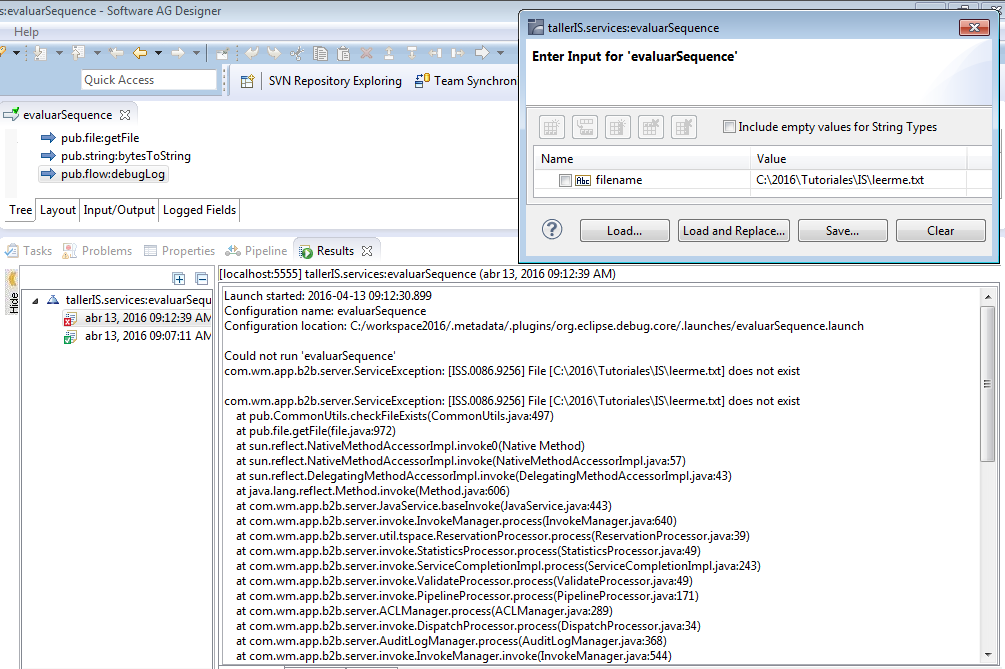


1. **Probar el servicio con un archivo valido (que existe)**

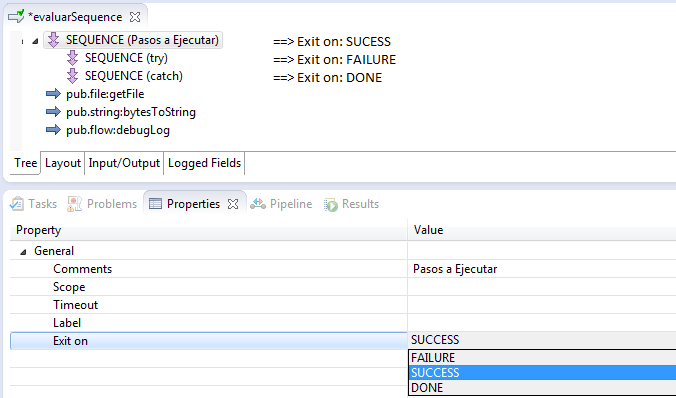




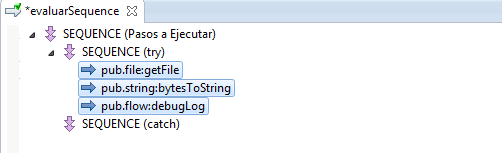
1. **Probar con un archivo que no existe:** Se envía una exception del servicio pub.file:getFile. La cual debe ser captura y procesada



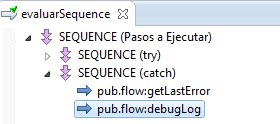
1. **Uso de Sequences (try/catch para caturar la exception)** 
   * Sequence 1: Exit on: SUCCESS (Termina si toda la secuencia 1 es exitoso)
   * Sequence 2: Exit on: FAILURE (Termina si cualquier paso de la secuencia 2 falla)
   * Sequence 3: Exit on: DONE (Termina cuando la secuencia 3 es completado)
   * Agregar comentarios a todas las secuencias



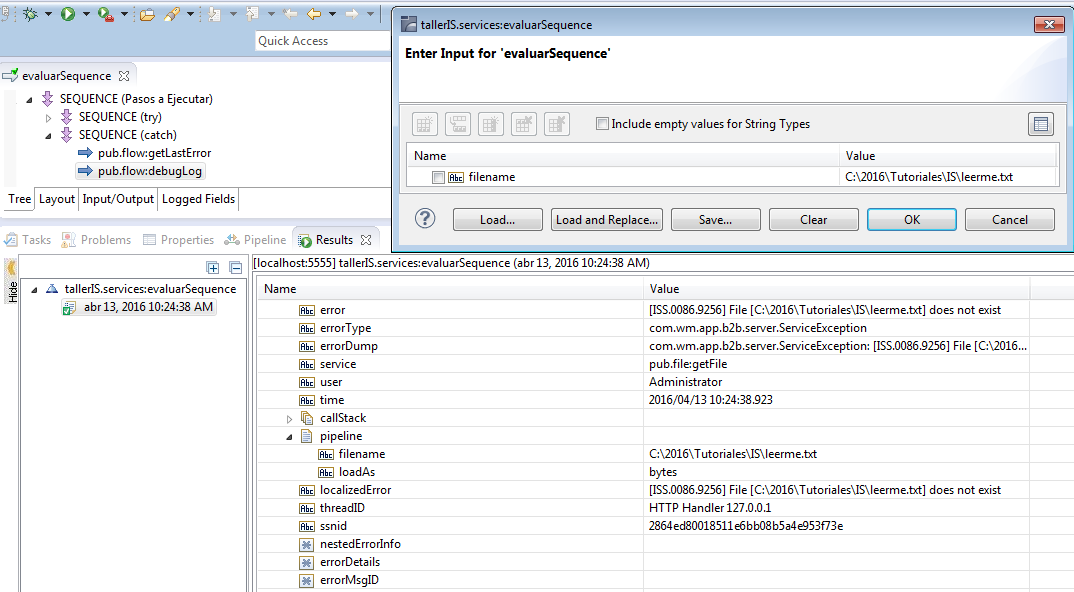
1. **Agregar los servicios dentro de Sequence – try**

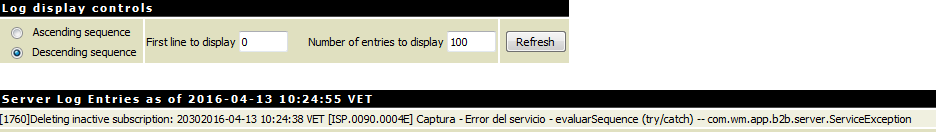


1. **Agregar el servicio pub.flow.getLastError (Captura la exception arrojada en el bloque try)**
2. **Imprimir en el log el error capturado (pub.flow.debugLog)**



1. **Probar el servicio con archivo no existente para captura la exception arrojada por el servicio**

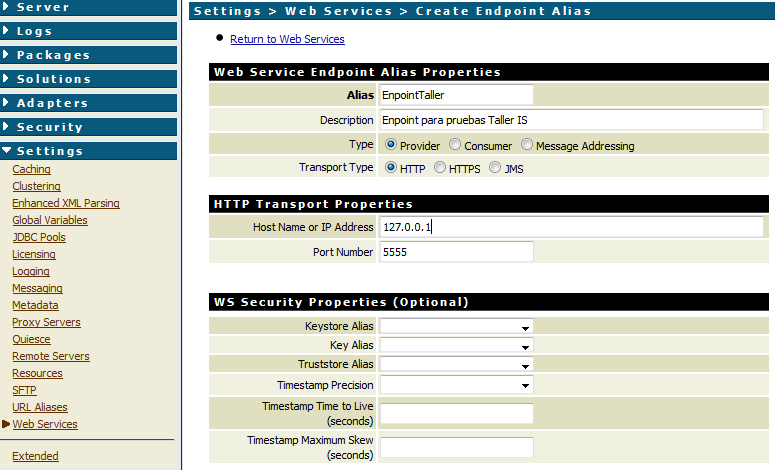


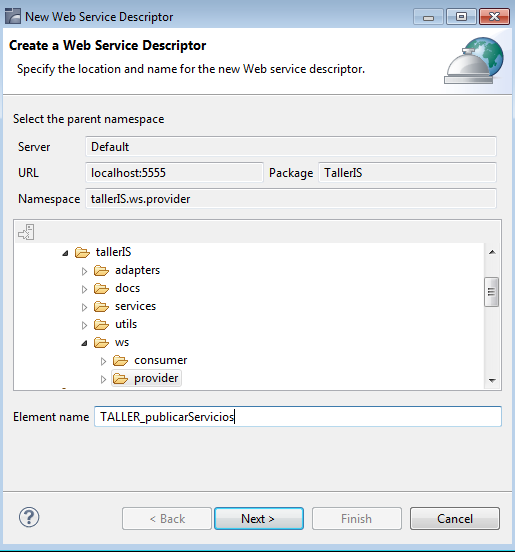


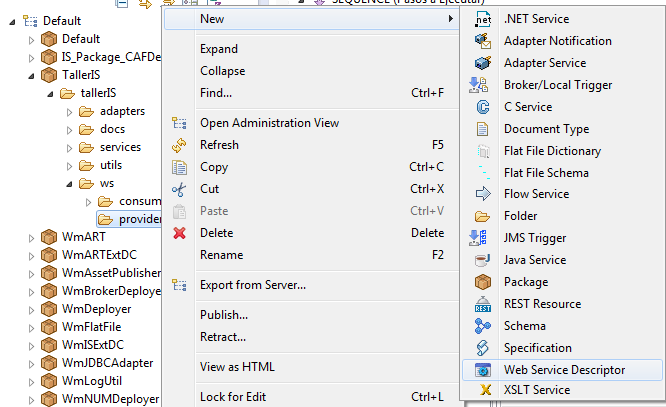
Publicar Servicios - Servicio Web: PROVIDER

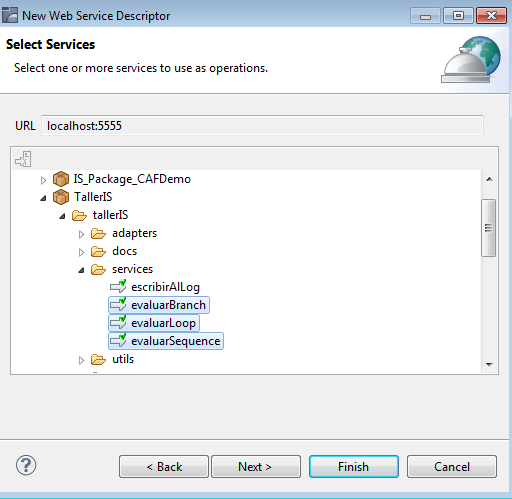
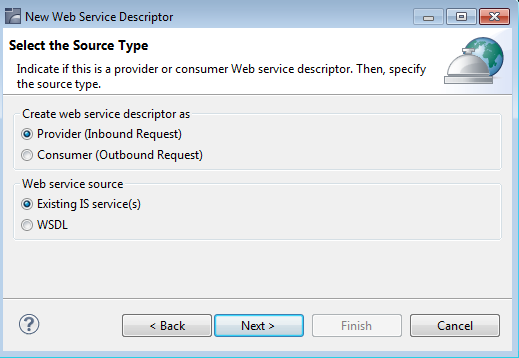
**Generar un Servicio Web con las operaciones previamente generadas**

1. **Crear un Endpoint Alias en el Integration Server**

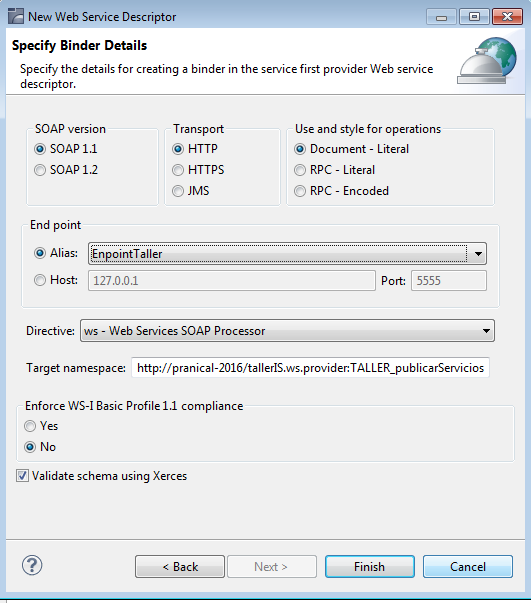


1. **Generar un provider: TALLER\_publicarServicios - como Servicio Web (Web service Descriptor)**
   * servicio-operación: evaluarBranch
   * Servicio-operación: evaluarLoop
   * Servicio-operacion: evaluarSequence

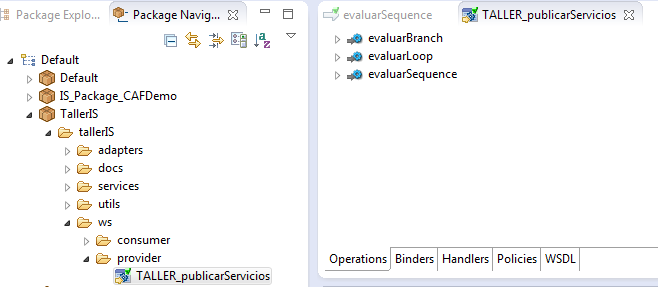




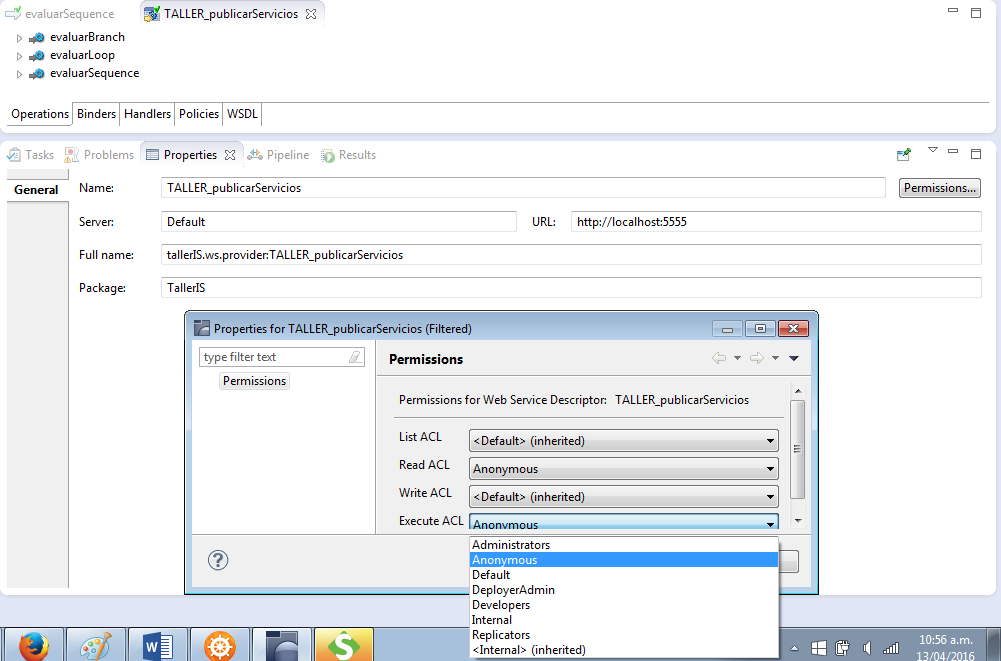
**Seleccionar el Endpoint Alias creado previamente**



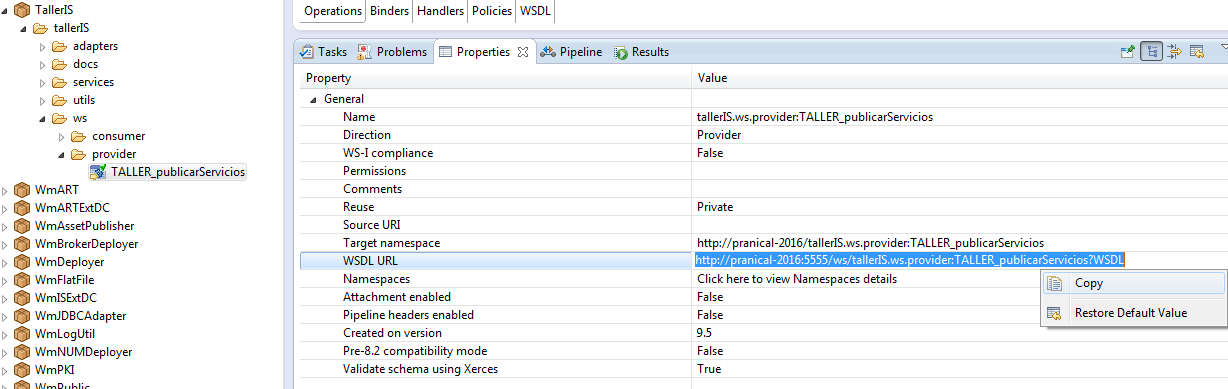
**Resultado**

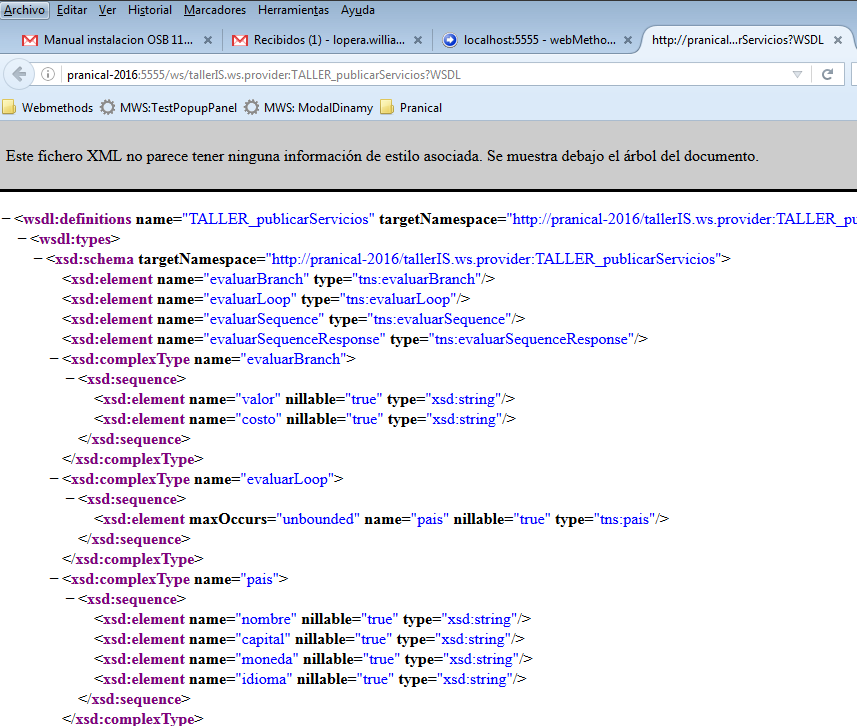


1. **Seleccionar la permisología Anonymous. Permite realizar pruebas de las operaciones (SOAPUI)**

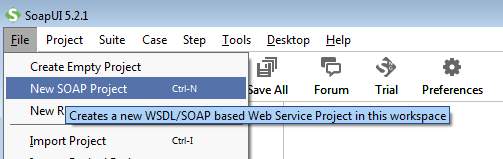


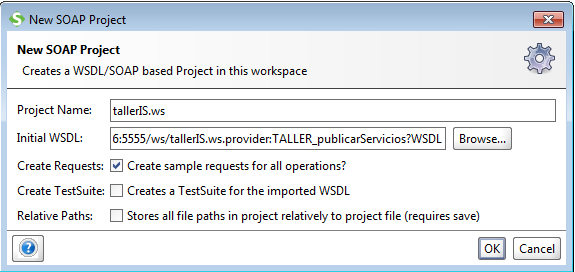
1. **Probar las operaciones Servicio Web - (SOAPUI)**
   * Copiar El WSDL y Probar en navegador.

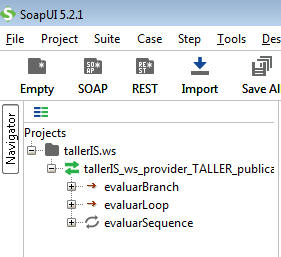




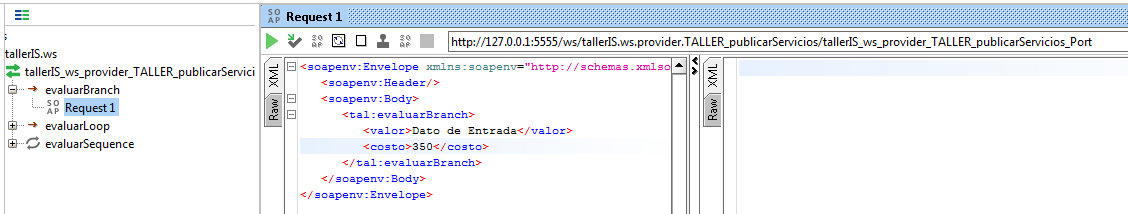
**Probar Servicio WEB en SOAPUI**

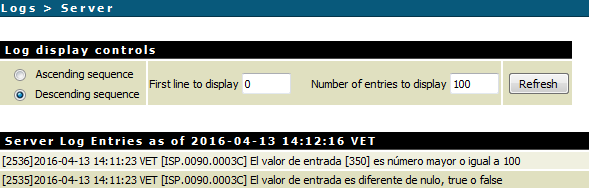


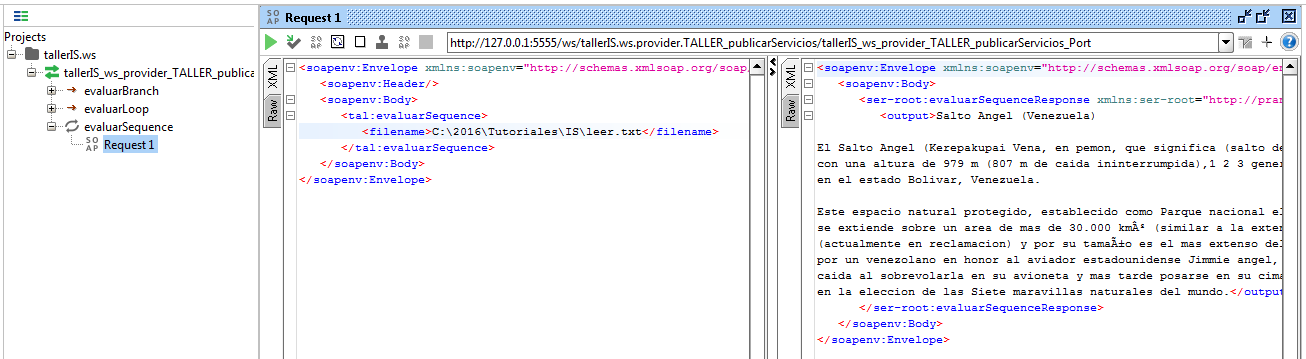




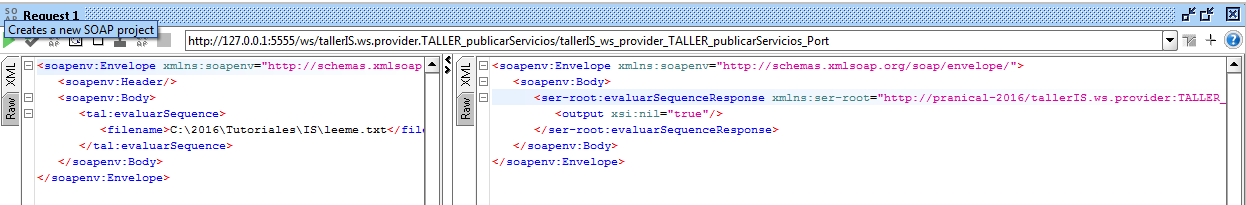
**Probar Operación evaluarBranch**





**Probar Operación evaluarSequence – Archivo existente**

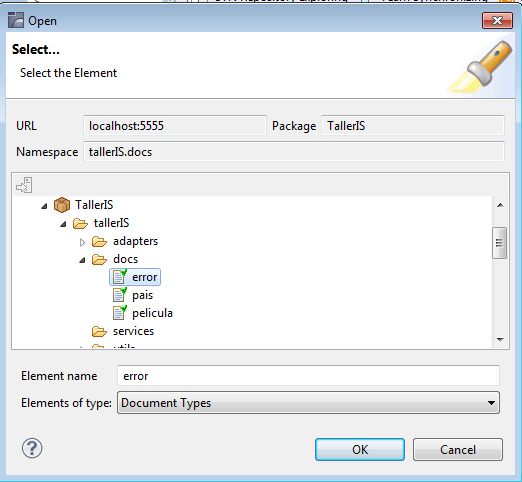
**Probar Operación evaluarSequence – Archivo no existente**

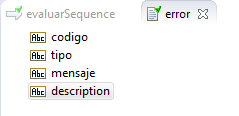
****



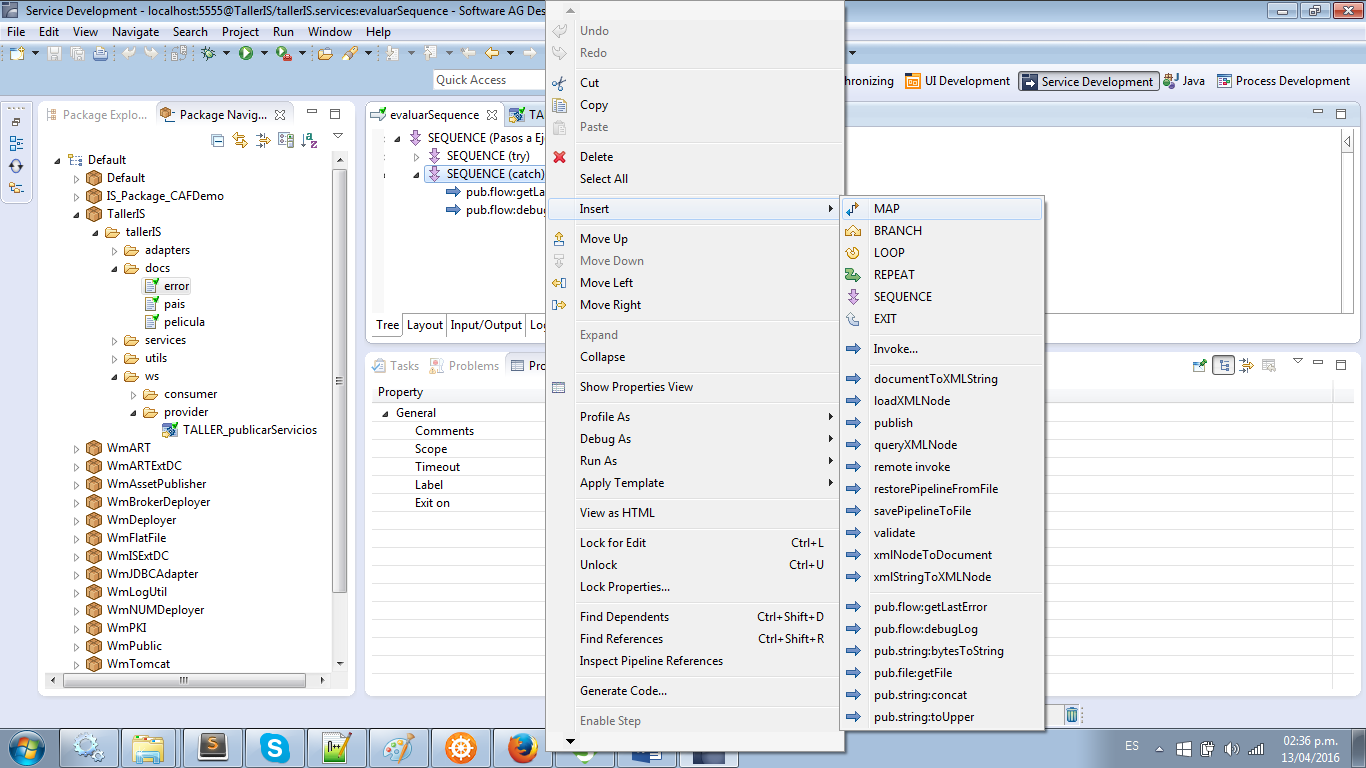
Enviar mensaje error - Servicio Web

**Generar un Custom Fault del Servicio Web para enviar mensajes de error (SOAP-ERROR) - faultcode**

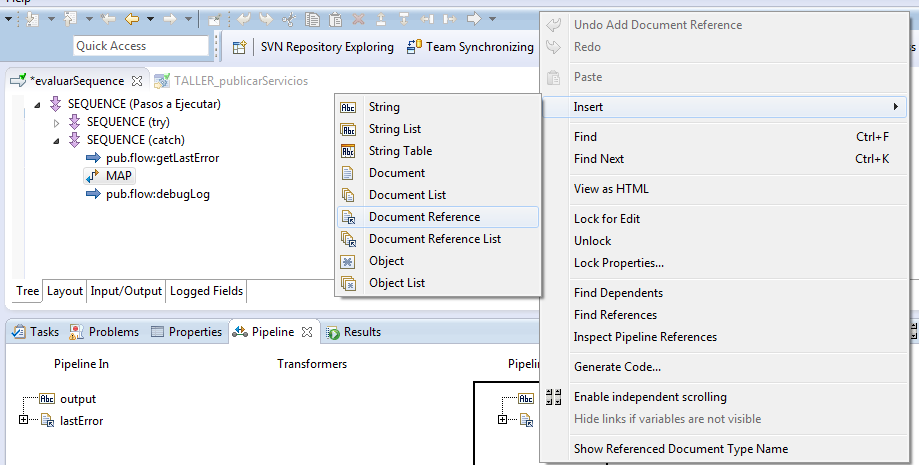
1. **Generar documento (Document Type) de nombre error**
   * codigo: Código que identifica al error
   * tipo: Tipo de error (INFO, WARNING, ERROR, FATAL, VALIDATION)
   * mensaje: Mensaje a enviar al cliente consumidor del Servicio Web (puede ser enviado a una vista del cliente – mensaje de información)
   * descripcion: Envio del mensaje Exception generado por la operación (puede ser
   * enviado a un archivo log)



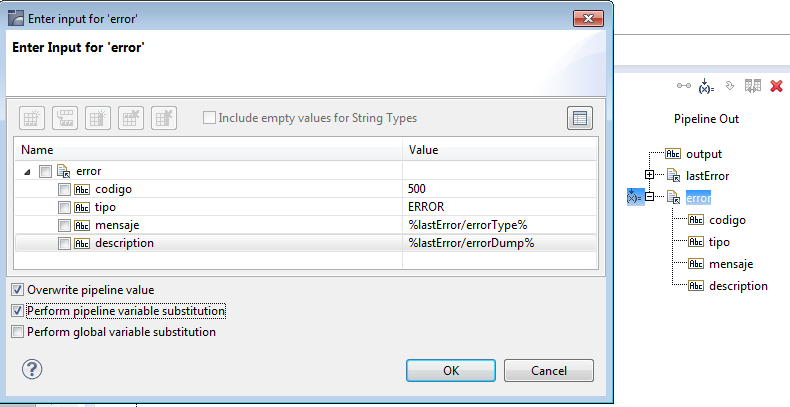
1. **Agregar Un MAP para incluir el documento error en el try/catch del Flow Service evaluarSequence**

****

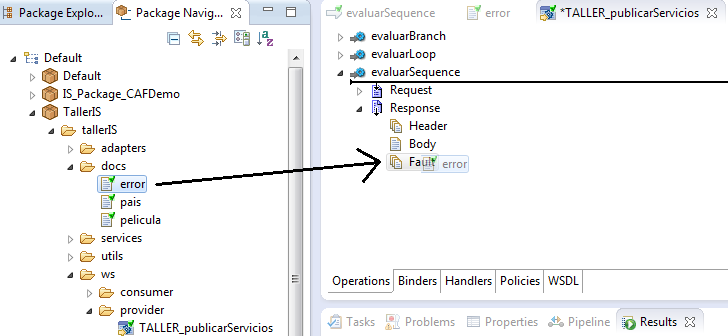
1. **Agregar un documento de referencia de nombre error dentro del MAP**



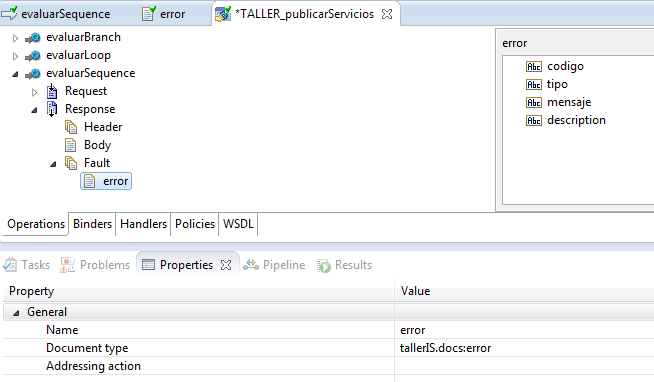
1. **Mapear o parsear los valores del documeno error**



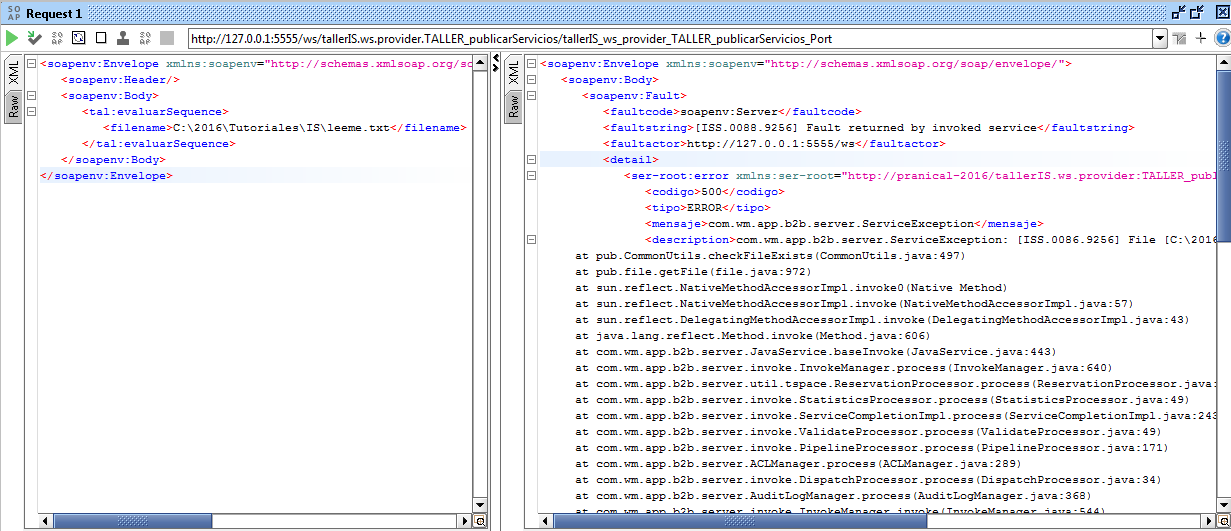
1. **Agregar el documento error a la opción Fault de la operación Provider TALLER\_publicarServicios (Arrastrar documento error al tag Fault del Provider)**



**Resultado**



1. **Probar la operación evaluarSequence del Servicio Web con SOAPUI**

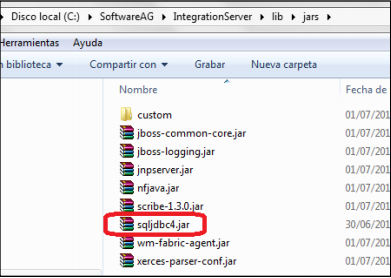


Adaptador

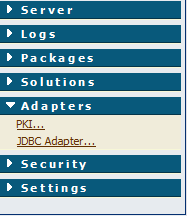
**Generar un adaptador para conexión a Base de datos**

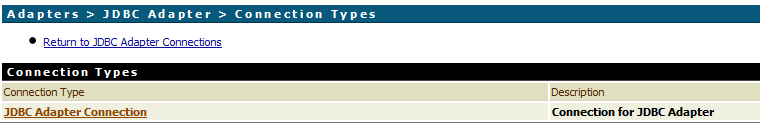
1. **Copiar el archivo [..\curso\sqljdbc4.jar] de acceso a SqlServer 2012 en**

**[C:\SoftwareAG\IntegrationServer\lib\jars]**



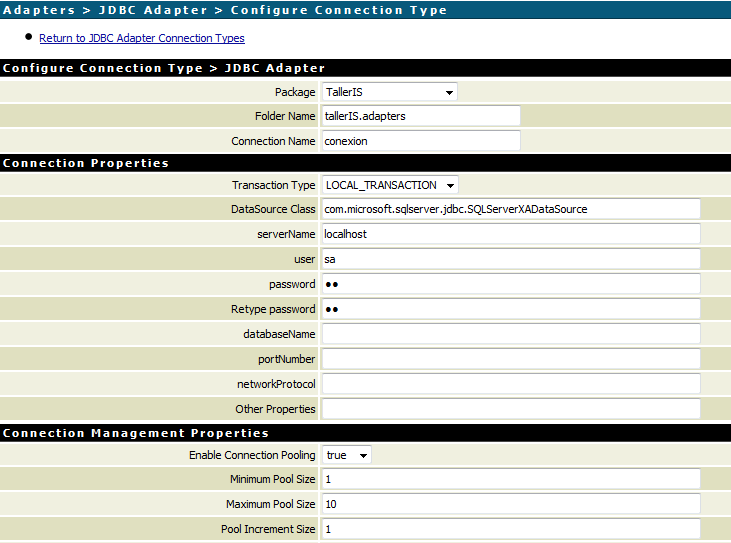
1. **Conectarse al Integration Server [Navegador]**
   * [http://localhost:5555]
   * [user/password: Administrator/manage]
2. **Crear un JDBC Adapter**



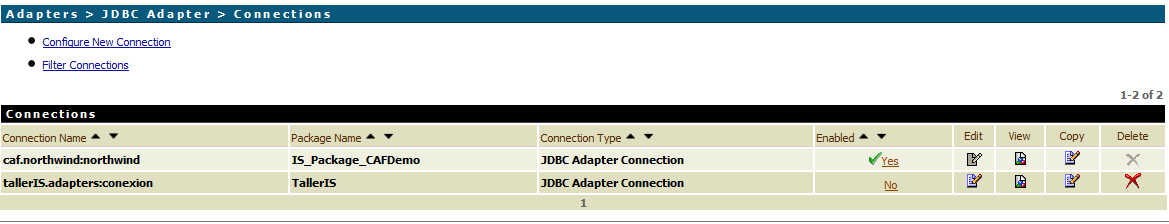


**Incluir los datos del JDBC Adapter:**

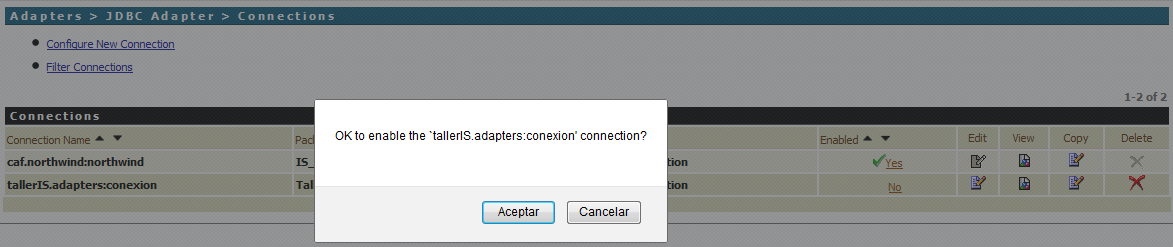
* **Package Name:** TallerIS (Paquete creado previamente en el Designer)
* **Folder Name:** tallerIs.adapters (Carpetas previamente creadas en el Designer)
* **Transaction Type:** LOCAL\_TRANSACTION (Define tipo de conexión)
* **DataSouce class:** com.microsoft.sqlserver.jdbc.SQLServerXADataSource
* **serverName :** Nombre del servidor de Base de Datos



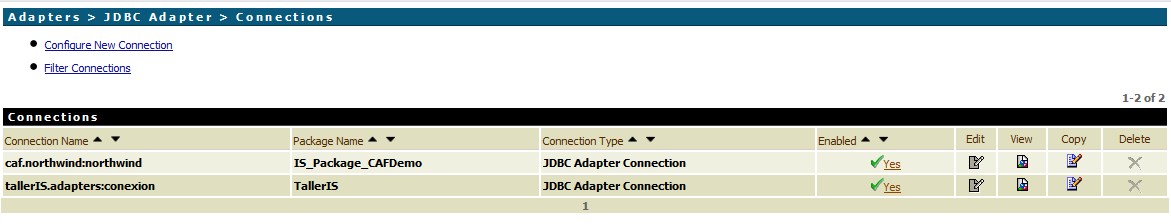
**Se genera la conexión:**



**Conectar a la Base de Datos**



**Conexión Exitosa:**



**En el Designer se genera el adaptador de conexión a Base de Datos creado**

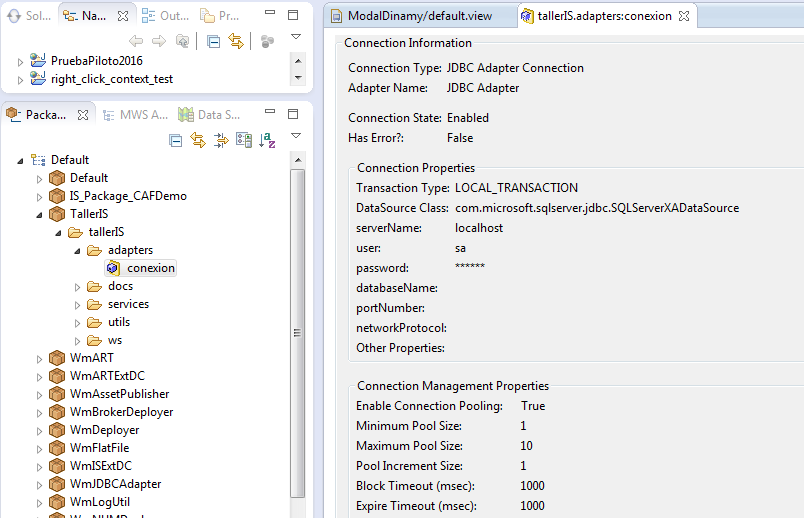
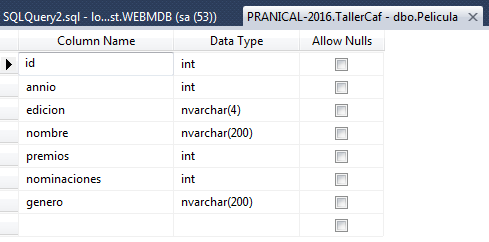
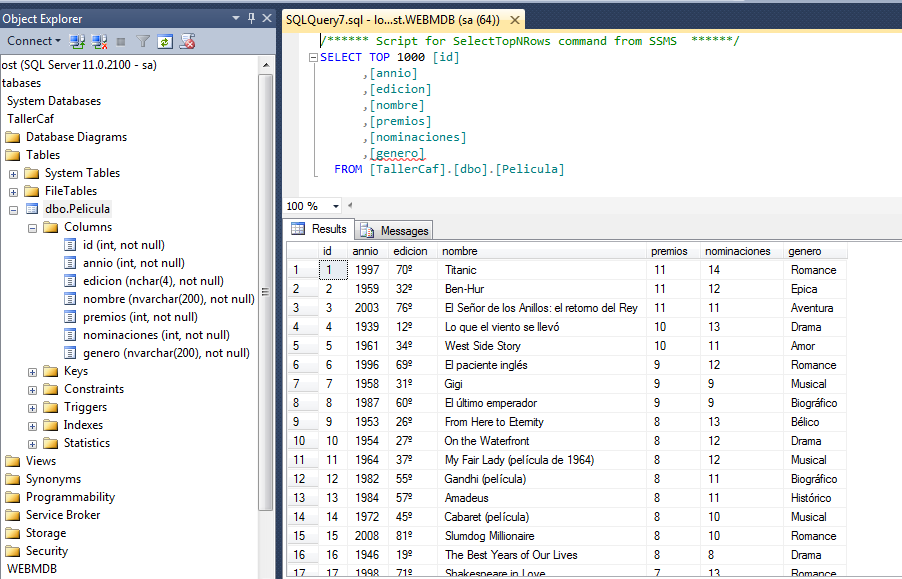
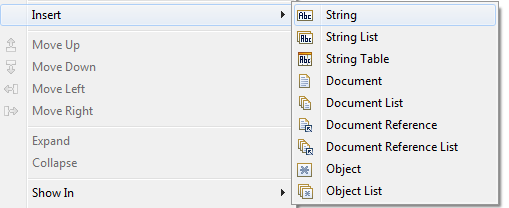


Tabla de Base de Datos

**Correr el archivo [../script.sql] para generar la tabla película y los datos de la tabla Requeridos por el taller.**





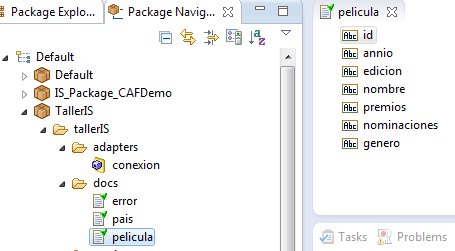


Servicio Consultar Películas

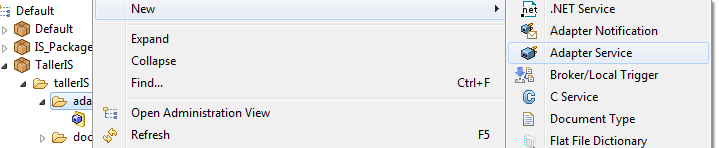
**Crear un servicio que permita consultar una lista de películas de una base de datos**

1. **Crear el documento película (Document Type)**

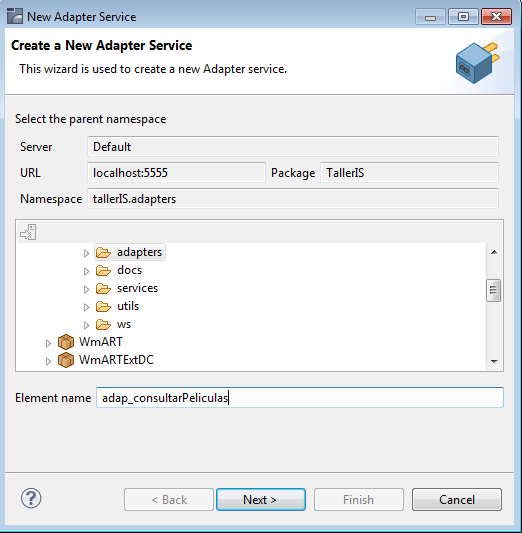
* id
* annio
* edicion
* nombre
* premios
* nominaciones
* genero

****

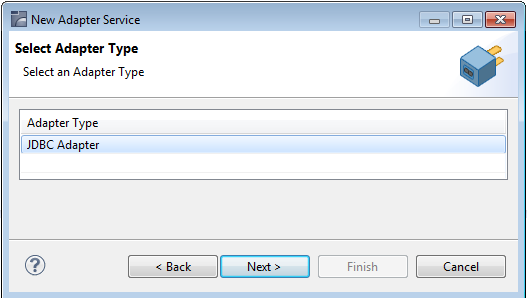
1. **Crear un adaptador para consultar los valores de películas ganadoras del oscar de una base de datos de SQLSERVER 2012**

****

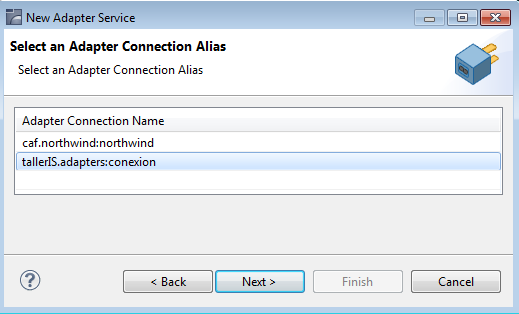
**Crear el adaptador adap\_consultarPelicula**

****

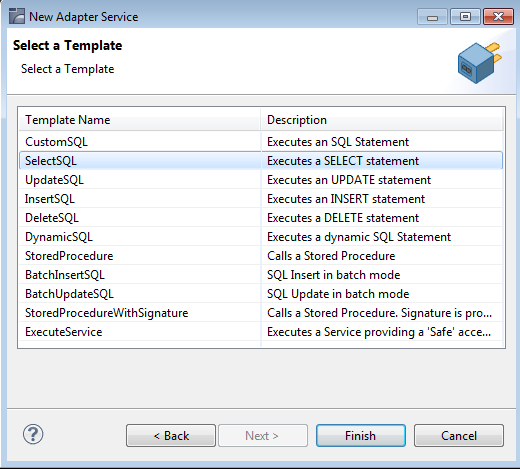
**Seleccionar el tipo de adaptador (driver de conexión)**

****

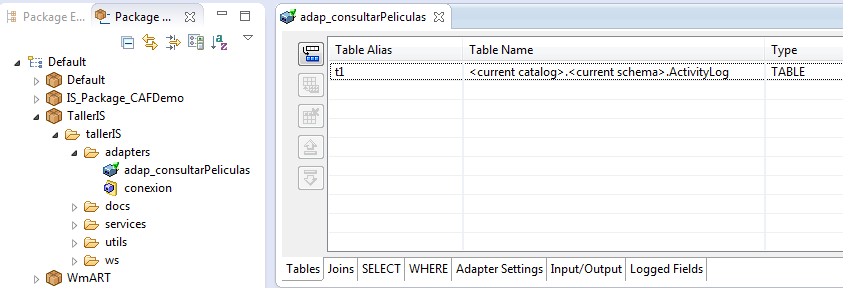
**Seleccionar el Adaptador de conexión (Adaptador creado previamente)**

****

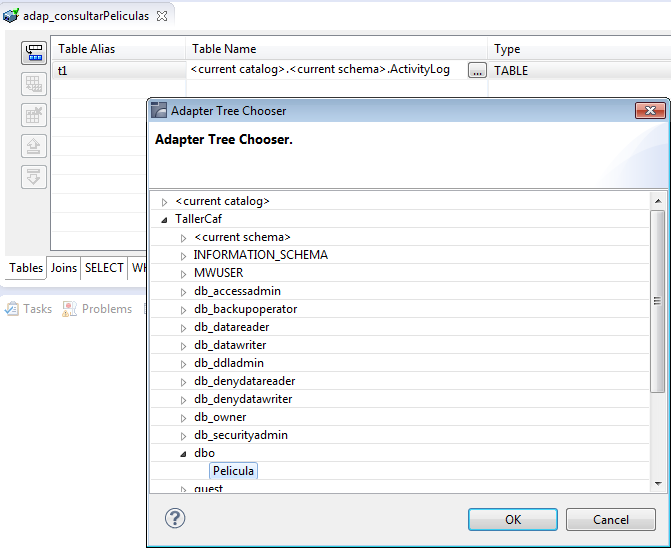
**Seleccionar la plantilla de operación a realizar en la base de datos. En este caso seleccionamos SelectSQL (Query de consulta)**

****

**Se genera la plantilla del adaptador seleccionado**

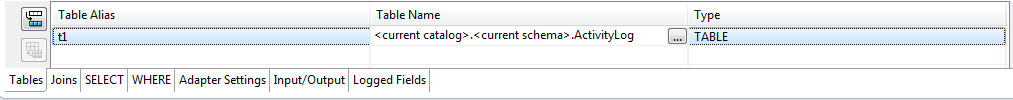
****

1. **Parametrizar los parámetros de la consulta a la base de datos. Tabla Película**

* **Seleccionar la tabla a consultar (Pelicula)**

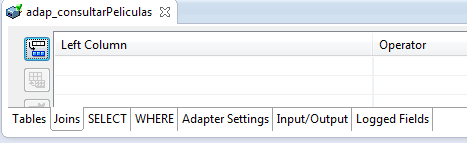
**Resultado**

* **Seleccionar Tables**

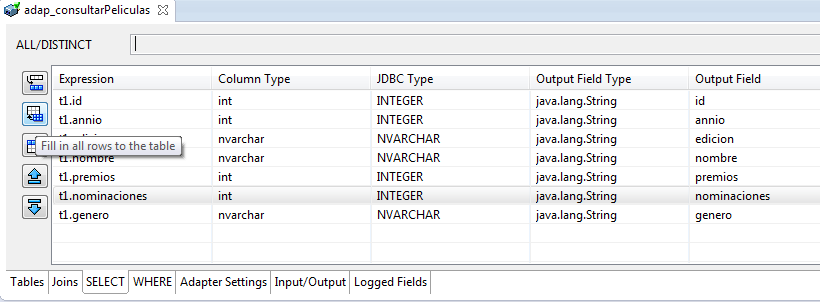
****

**Nota: La opción Tables: Permite seleccionar las tablas a procesar**

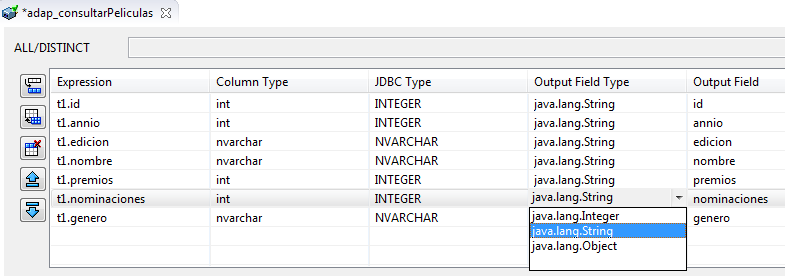
* **Realizar Joins**

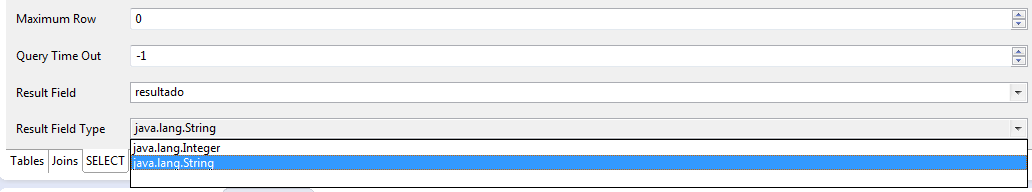
****

**Nota: La opción Joins: Permite Realizar proceso de relaciónes (Joins) entre tablas**

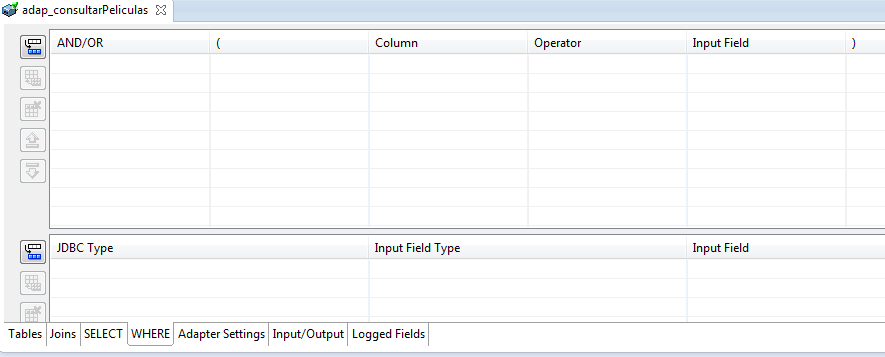
* **Realizar el Select de los campos a consultar de la tabla Pelicula** 
  + **Presionar opción Fill in all rows to the table para poblar la tabla con los campos contenidos en la tabla Pelicula**

**Nota: La opción Select: Permite seleccionar los campos de consulta de las tablas a procesar**

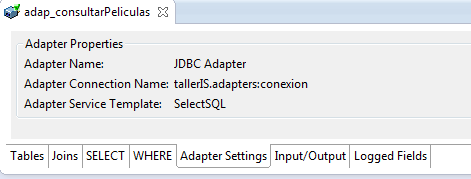
* **Poblar los datos requeridos por la tabla para Parsear los tipos de valores dados y esperados por los campo de la tabla (Seleccionar todos los campos como String -–> Servicio Web [uso de Provider])**
* **Agregar una variable de control [resultado (Integer o String)]. Esta variable retorna 0 si no hay registros o la cantidad de registros retornados por la consulta**

****

* **Realizar un WHERE de la consulta o query a procesar (en este caso se desea consultar todos los datos)**

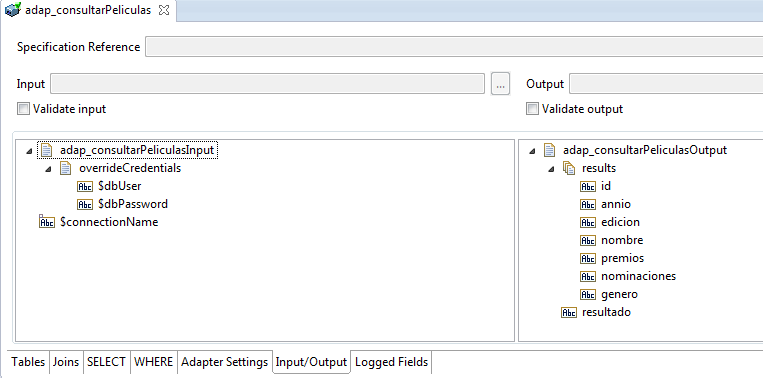
****

**Nota: La opción Where: Permite realizar consultas por parametros**

* **Consultar tipo de conexión de Base de Datos**

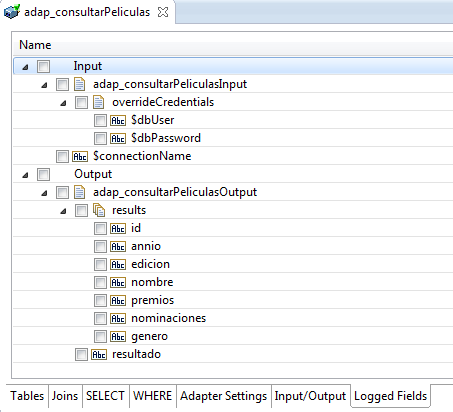
**Nota: La opción Adapter Settings: Permite consultar el tipo de conexión y el tipo de plantilla seleccionada**

* **Consultar los valores parámetros de entrada y salida del adaptador (contrato)**



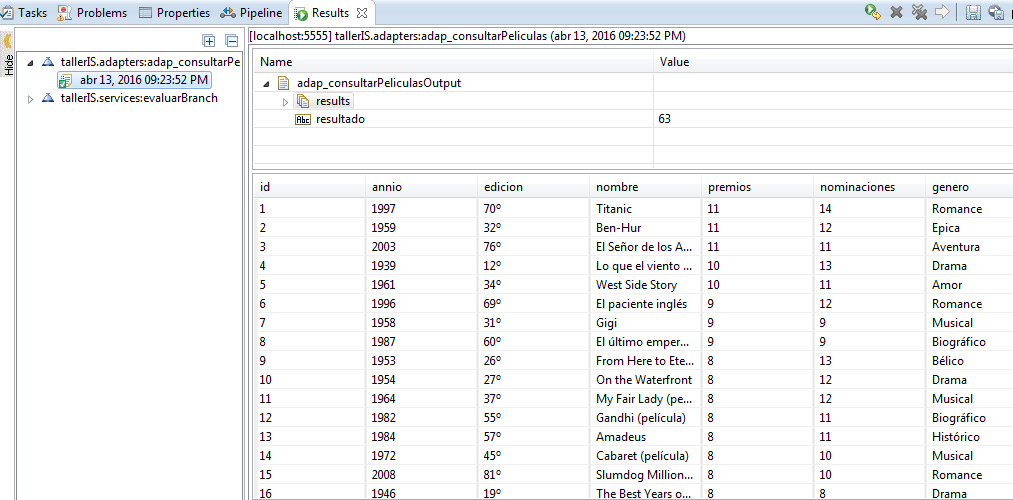
**Nota: La opción Input Output: Permite consultar parámetros de entrada/salida**

* **Seleccionar parámetros de entrada y salida para impresión en archivos log’s**

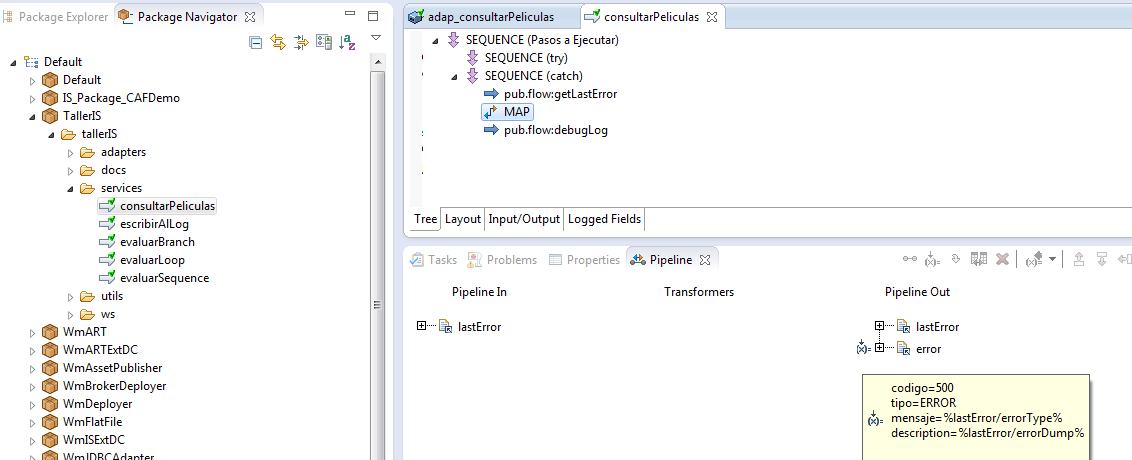


**Nota: La opción Logged Fields: Permite seleccionar los parámetros de entrada/salida a imprimir en archivos log’s**

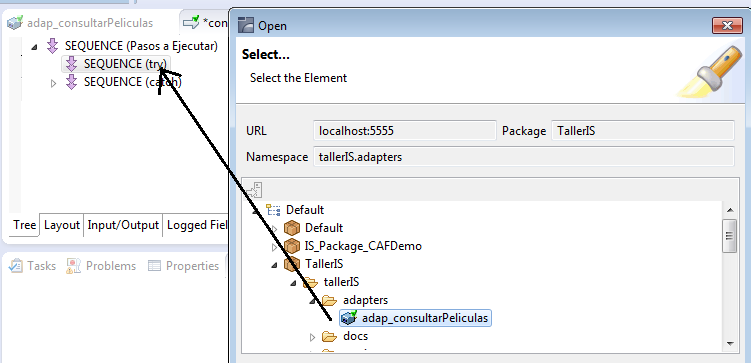
1. **Probar el adaptador: adap\_consultarPeliculas**



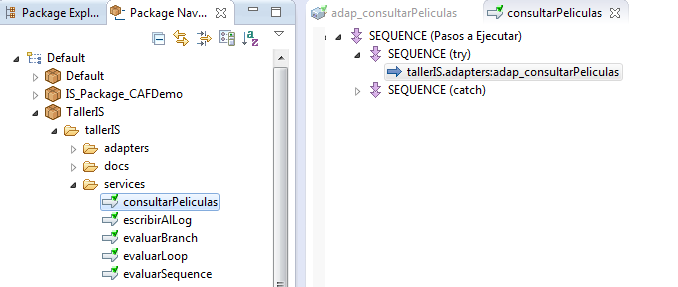
1. **Crear el Flow Service: consultarPeliculas**

****

1. **Agregar el adaptador al Servicio**

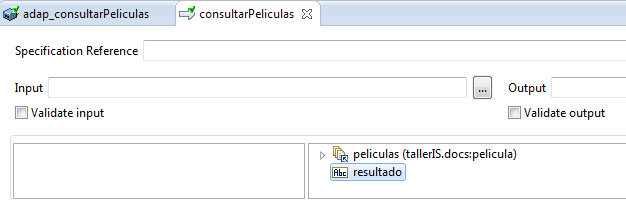
****

**Resultado:**

****

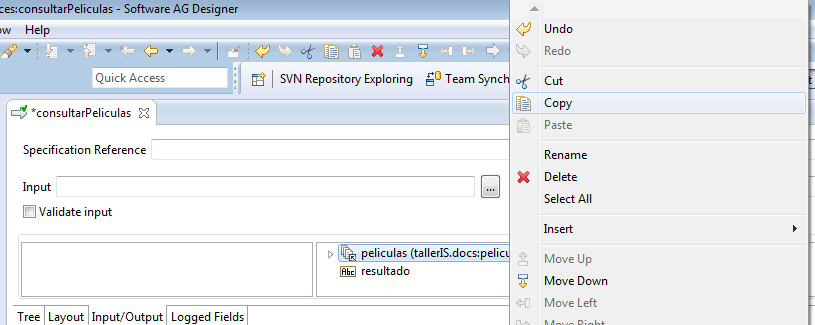
1. **ffsdfds**
2. **Agregar documentos de salida del servicio**

* **películas:** lista de Document de Referencia del documento película
* **resultado:** String con la cantidad de datos devueltos por la consulta

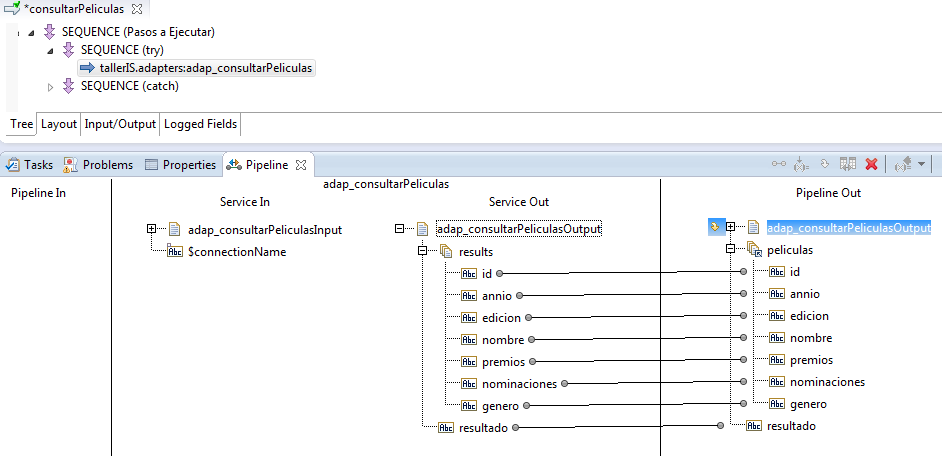


1. **Mapear los datos retornados por el adaptador al servicio**

**Copiar el documento películas de salida en la opción Input/Output**

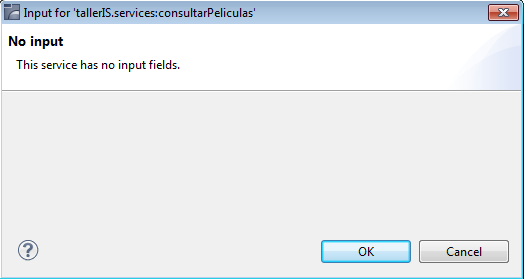
****

**Pegar el documento películas copiado en la opción Pipeline y mapear los valores de respuesta del adaptador al documento películas copiado de salida**

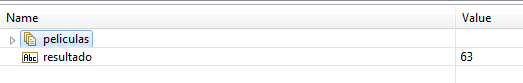
****

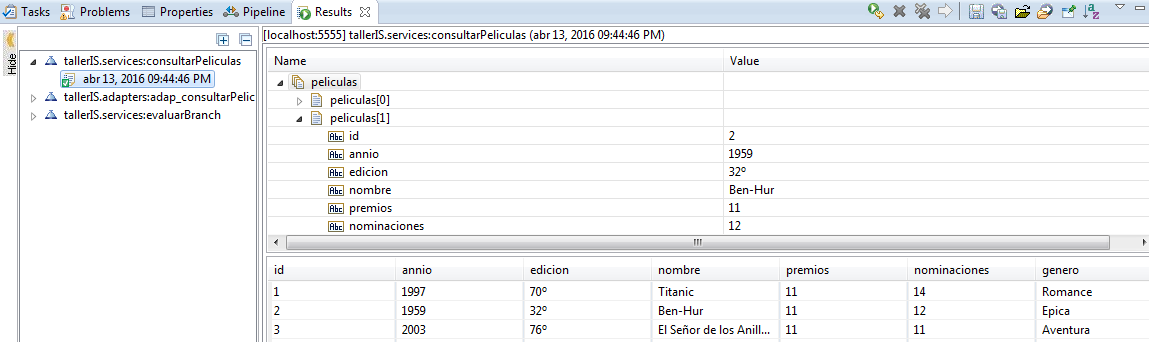
1. **Probar el servicio consultarPeliculas**

**Se abre una ventana informando que no existe parámetros de entrada al servicio**



**Salida de datos del servicio: Retorna una lista de películas (documento) y el campo resultados con el total de datos obtenidos de la consulta**

****

****