

Proyecto Colonial Battlestar Combat Simulator

Pablo García Sánchez

pgarcia@atc.ugr.es

Departamento de Arquitectura y Tecnología de Computadores Cylon

Curso Web 2.0 Arquitectura Orientada a Servicios en Java

Escuela de Posgrado

Febrero/Marzo de 2010

La idea

- El Doctor Gaius Baltar nos ha pedido que orquestemos algunos servicios web existentes para hacer simulaciones de las estrellas de combate.



¿Cómo desarrollarlo?

- Primera parte: **BPELpreparar**
 - Paso 1: Hacer un BPEL que obtenga el número de vipers de una estrella (llamando a `getVipers(id)`)
 - Paso 2: Obtener el número de pilotos (`getPilotos(id)`) y compararlo con el número de vipers obtenido y calcular el menor (N)
 - Paso 3: Iterar por los N primeros vipers para prepararlos (`prepararViper(idViper)`)

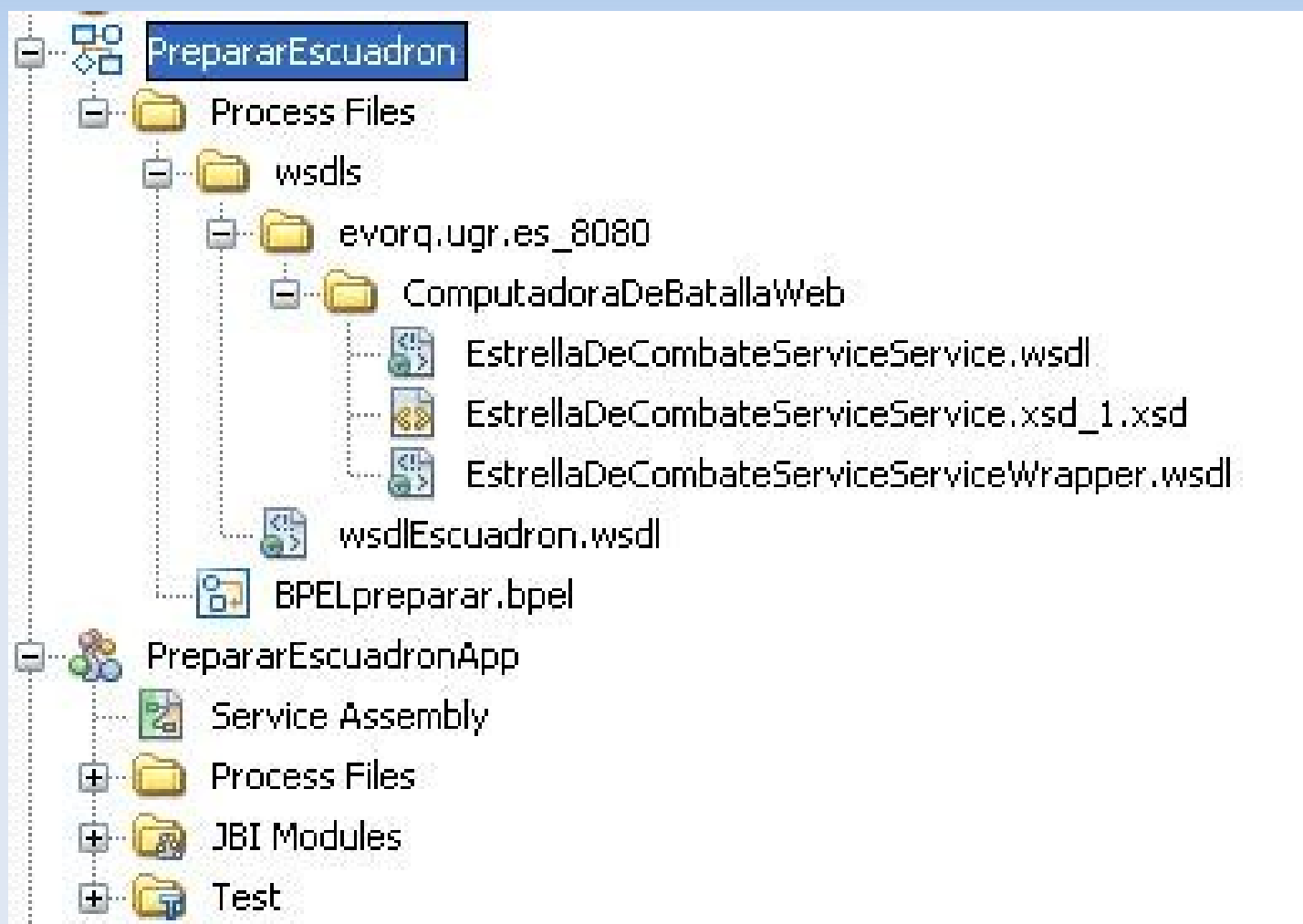
Pasos a seguir (y 2)

- Segunda parte: **BPELenfrentar**
 - Paso 1: Llamar al primer BPEL dos veces (uno por cada estrella a enfrentar) que prepara los Vipers
 - Paso 2: Llamar a `getVipersPreparados(idEstrella)` de las dos estrellas a enfrentar
 - Paso 3: Con esas dos listas llamamos a `combate(lista1, lista2)` y devolvemos el String del resultado.

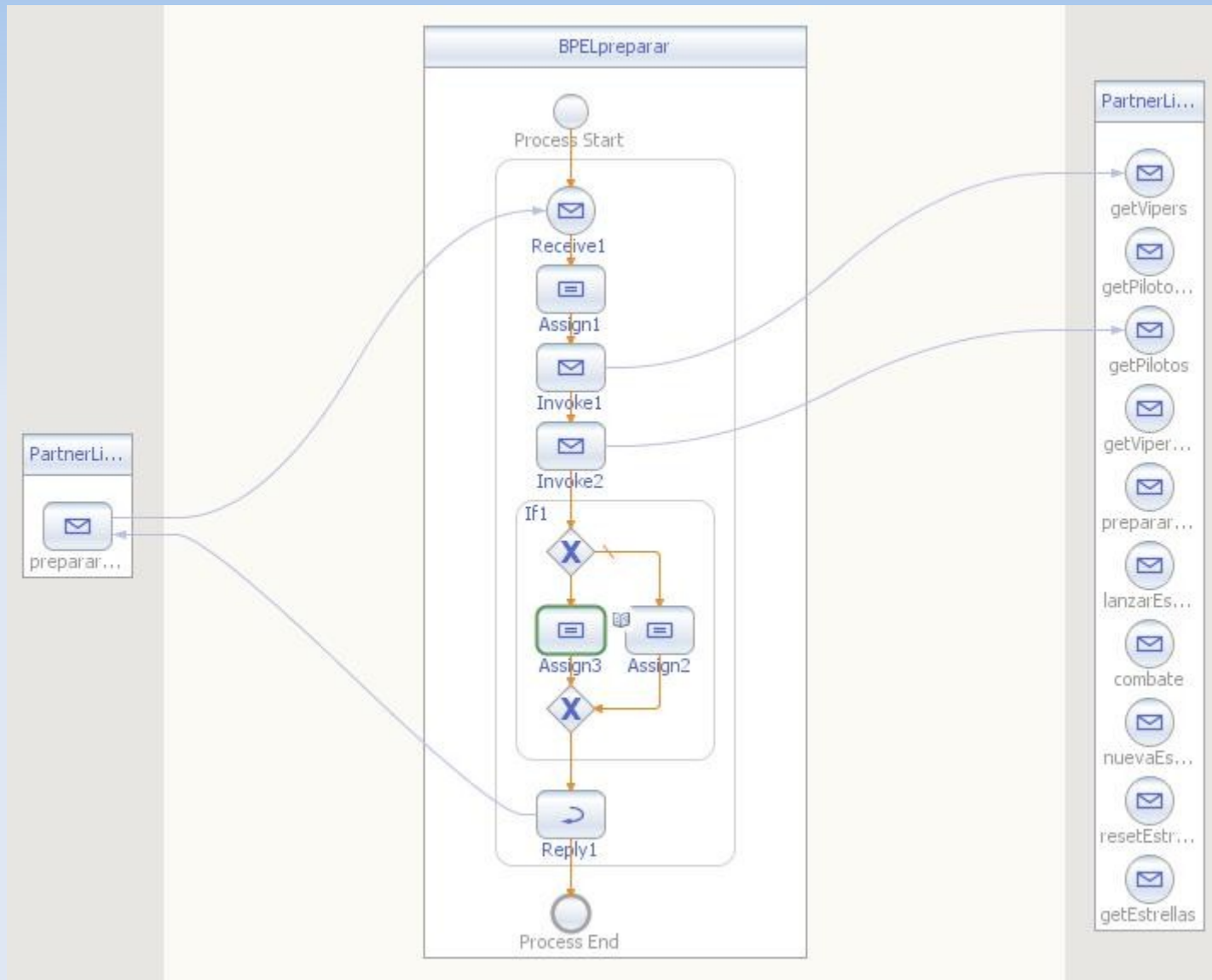
Primera Parte

- Creamos BPEL Module
 - Añadimos WSDL local: *wsdlEscuadron*
 - Operación `xsd:long = prepararEscuadron(xsd:long)`
 - Añadimos WSDL externo:
<http://evorq.ugr.es:8080/ComputadoraDeBatallaWeb/EstrellaDeCombateServiceService?wsdl>
- Creamos Composite application

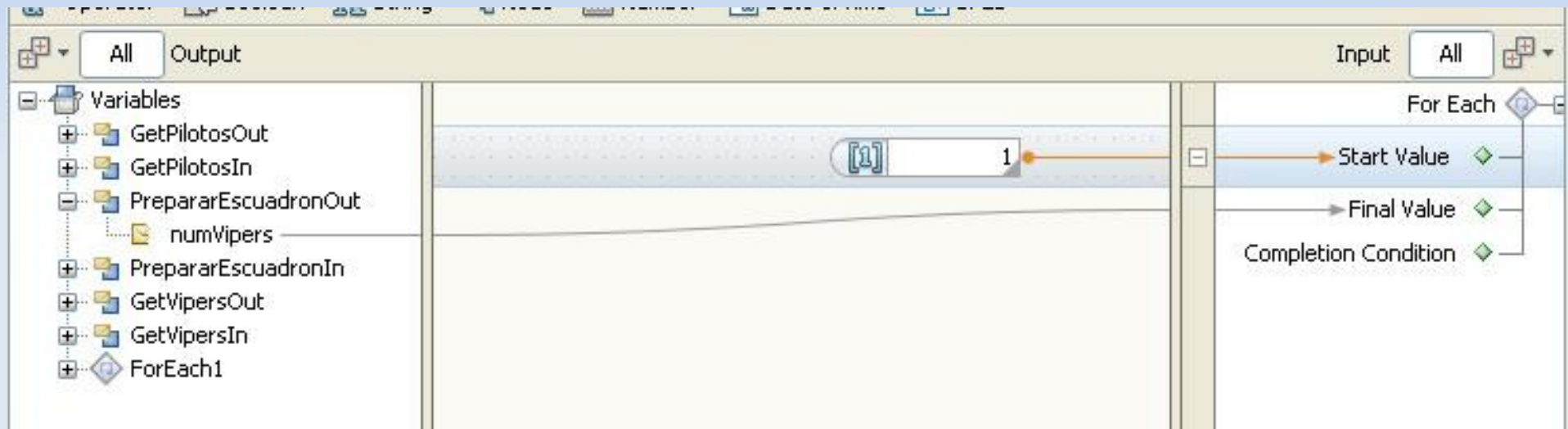
Primera Parte



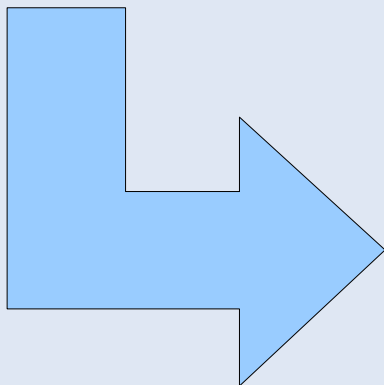
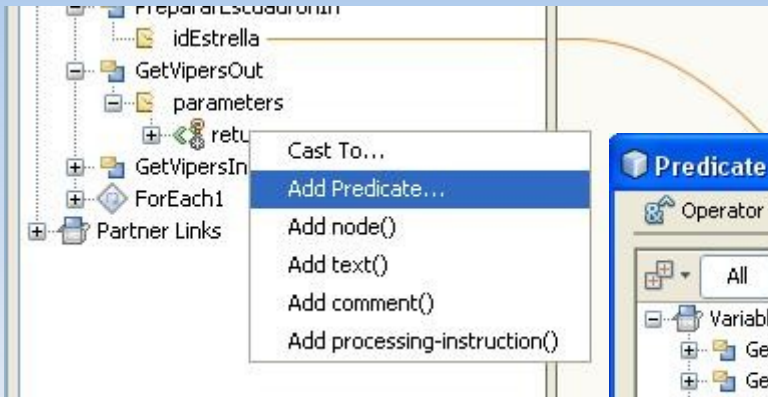
Tras el paso 1 y el paso 2...



Paso 3: Iterar (iniciar el contador del For Each)



Paso 3: Iterar (seleccionar el nodo con el contador)

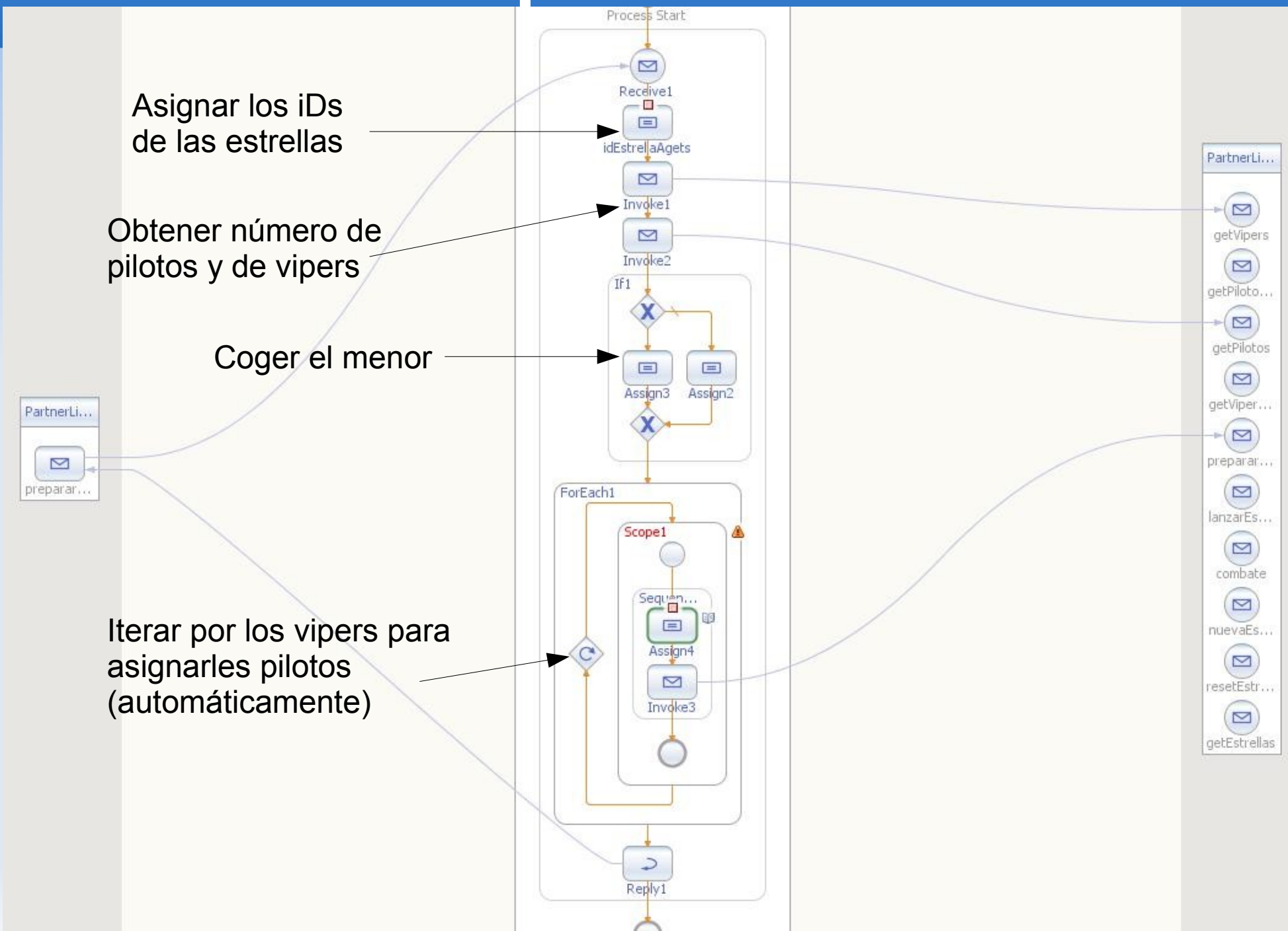


The Predicate Editor dialog box is shown. It has a title bar 'Predicate Editor' and a toolbar with icons for Operator, Boolean, String, Node, Number, Date & Time, and BPEL. The main area is divided into three panes: Variables, Output, and Input. The Variables pane shows a tree structure with the following nodes: GetPilotosOut, GetPilotosIn, PrepararEscuadronOut, PrepararEscuadronIn, GetVipersOut, parameters, return (highlighted), id, ataque, defensa, fuera, nombre, piloto, GetVipersIn, ForEach1, ForEach1Counter, and Scope1. The Output pane shows a large orange curve. The Input pane shows a 'Predicate' field. The Context field at the bottom contains '\$GetVipersOut.parameters/return'. Buttons for 'Set Focus', 'Ok', and 'Cancel' are visible.

Paso 3: Iterar (el assign del cuerpo del For Each queda así)



"Preparar" listo

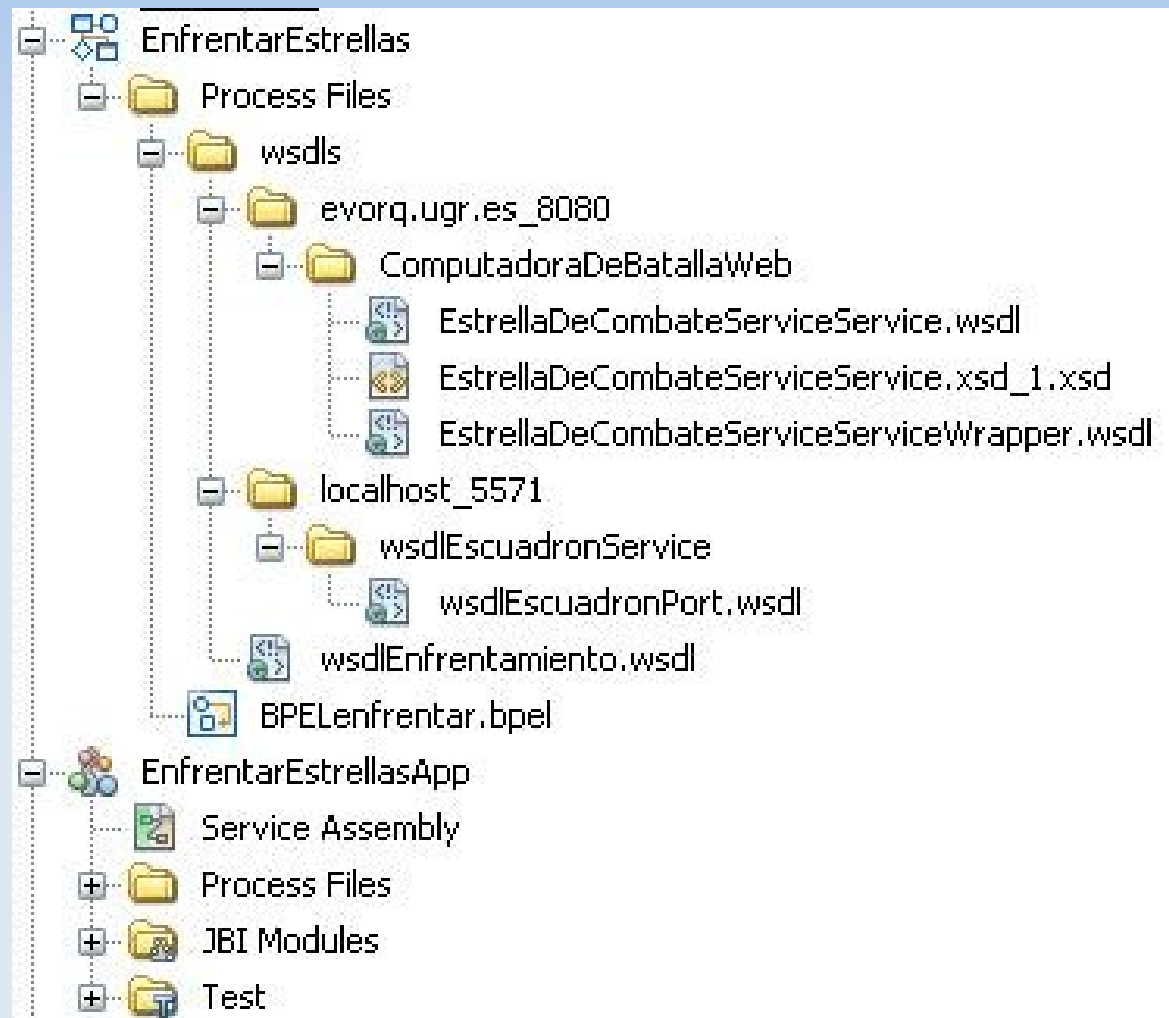


Segunda Parte

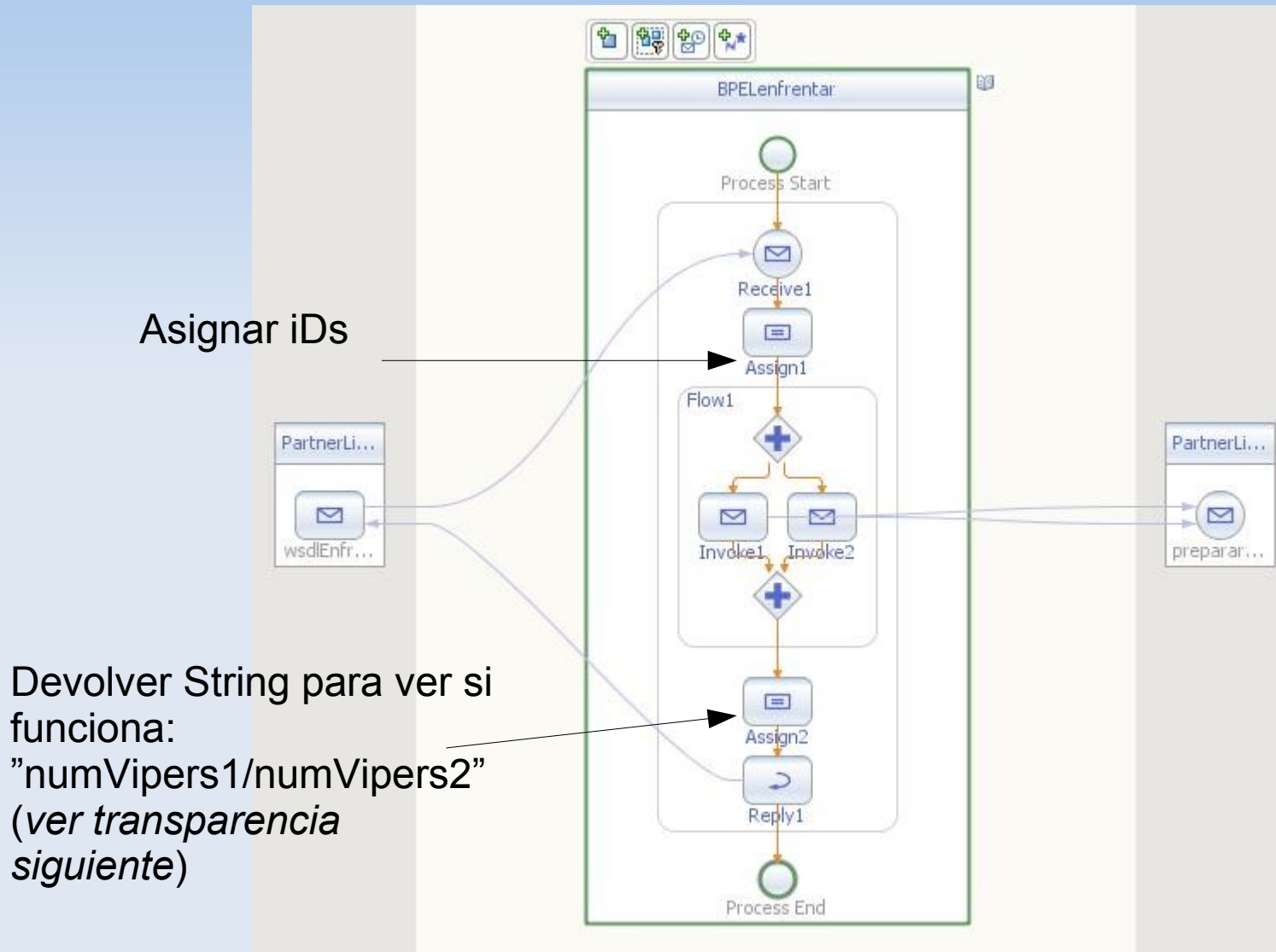
- Creamos un nuevo proyecto BPEL Module y Composite Application y añadimos:
 - El WSDL a ComputadoraDeBatallaWeb
 - El WSDL del anterior proyecto
 - El WSDL local
 - Con la operación

```
xsd:string = enfrentar(xsd:long id1, xsd:long id2)
```

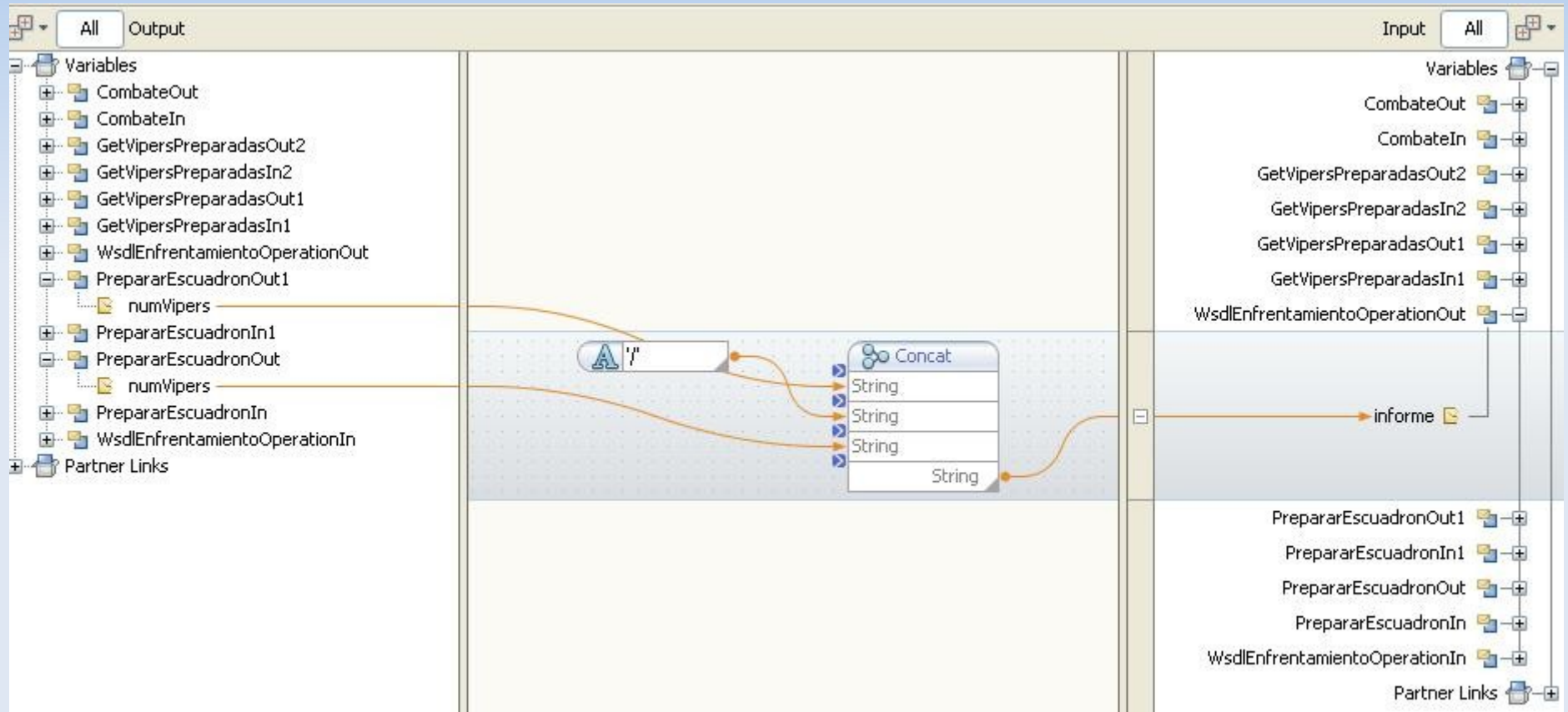
Segunda Parte



Segunda parte: Paso 1

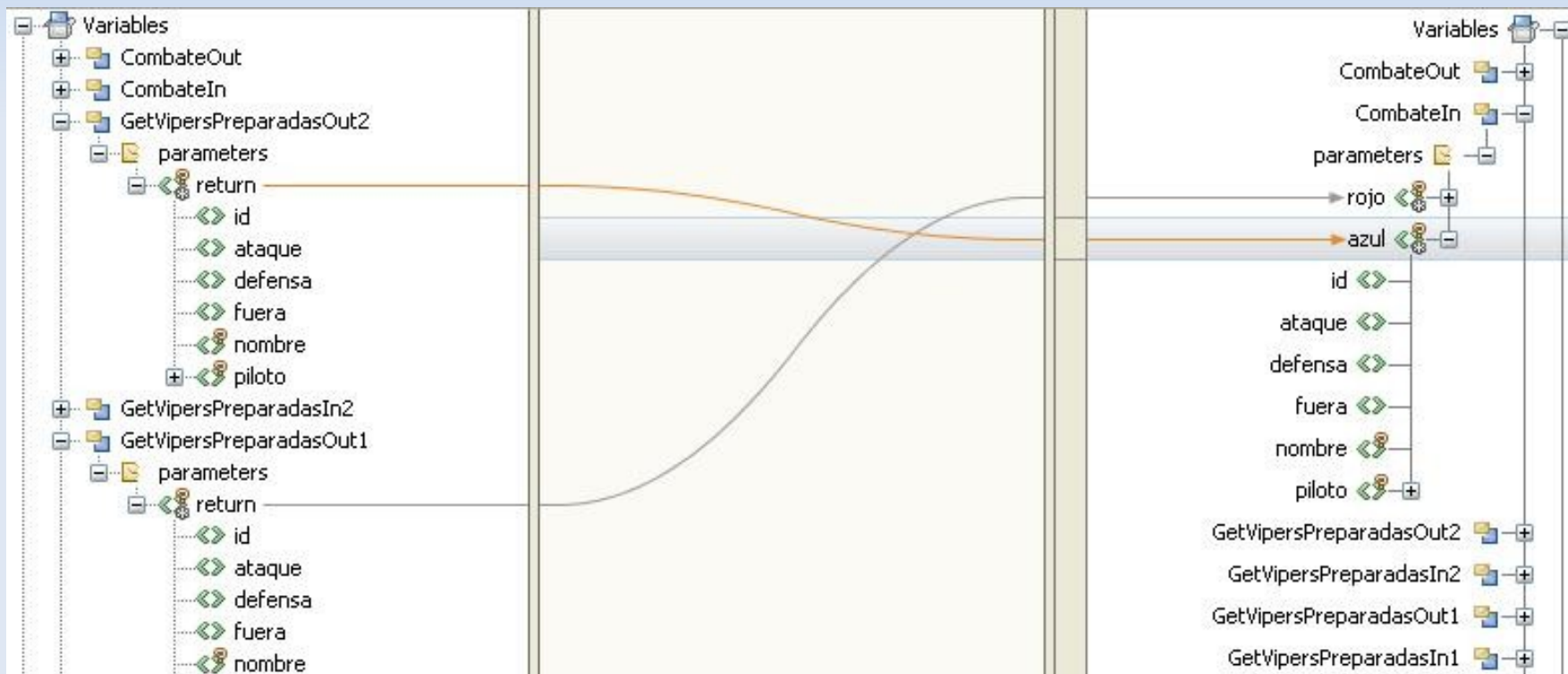


Devolver String

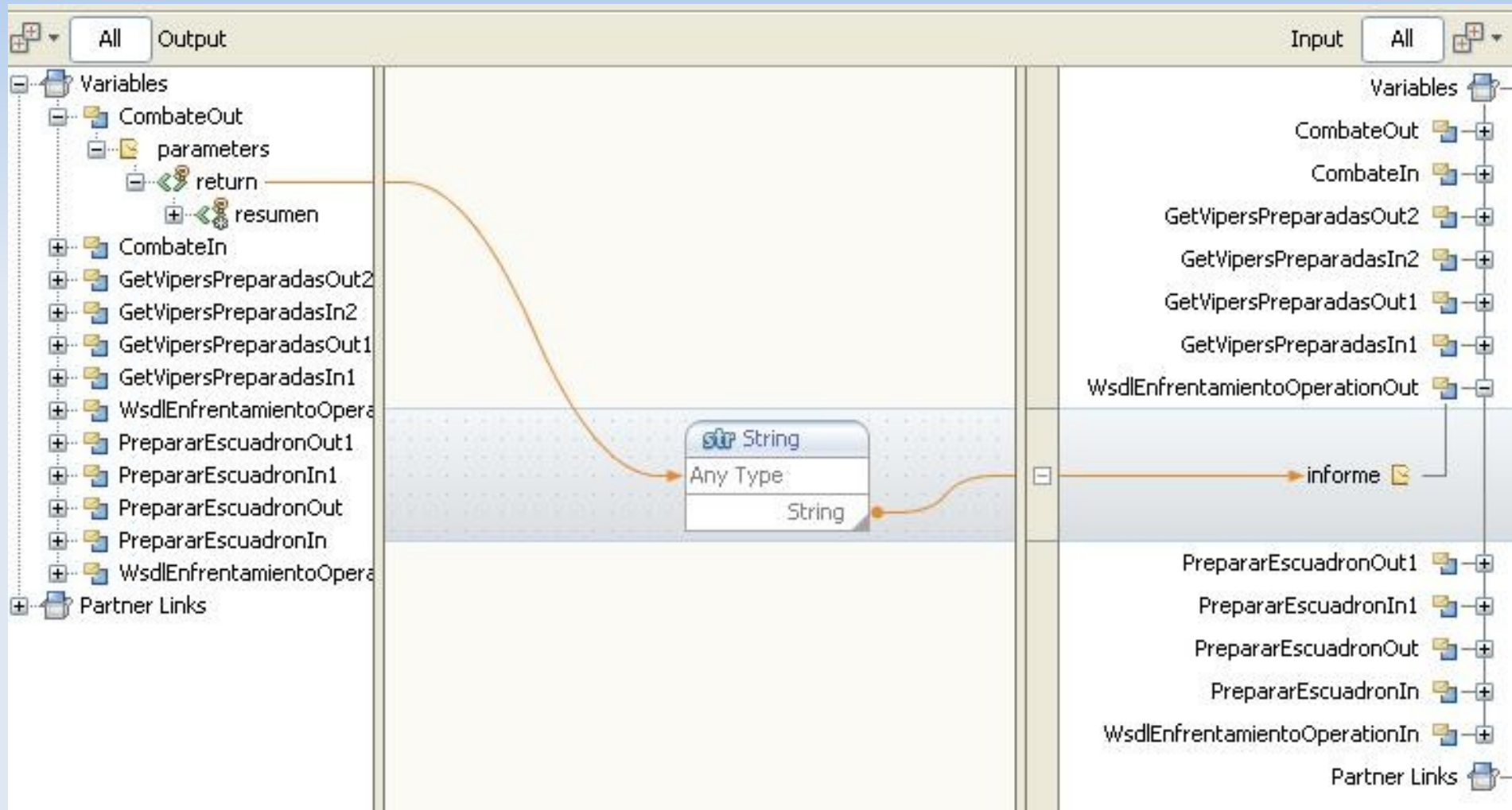


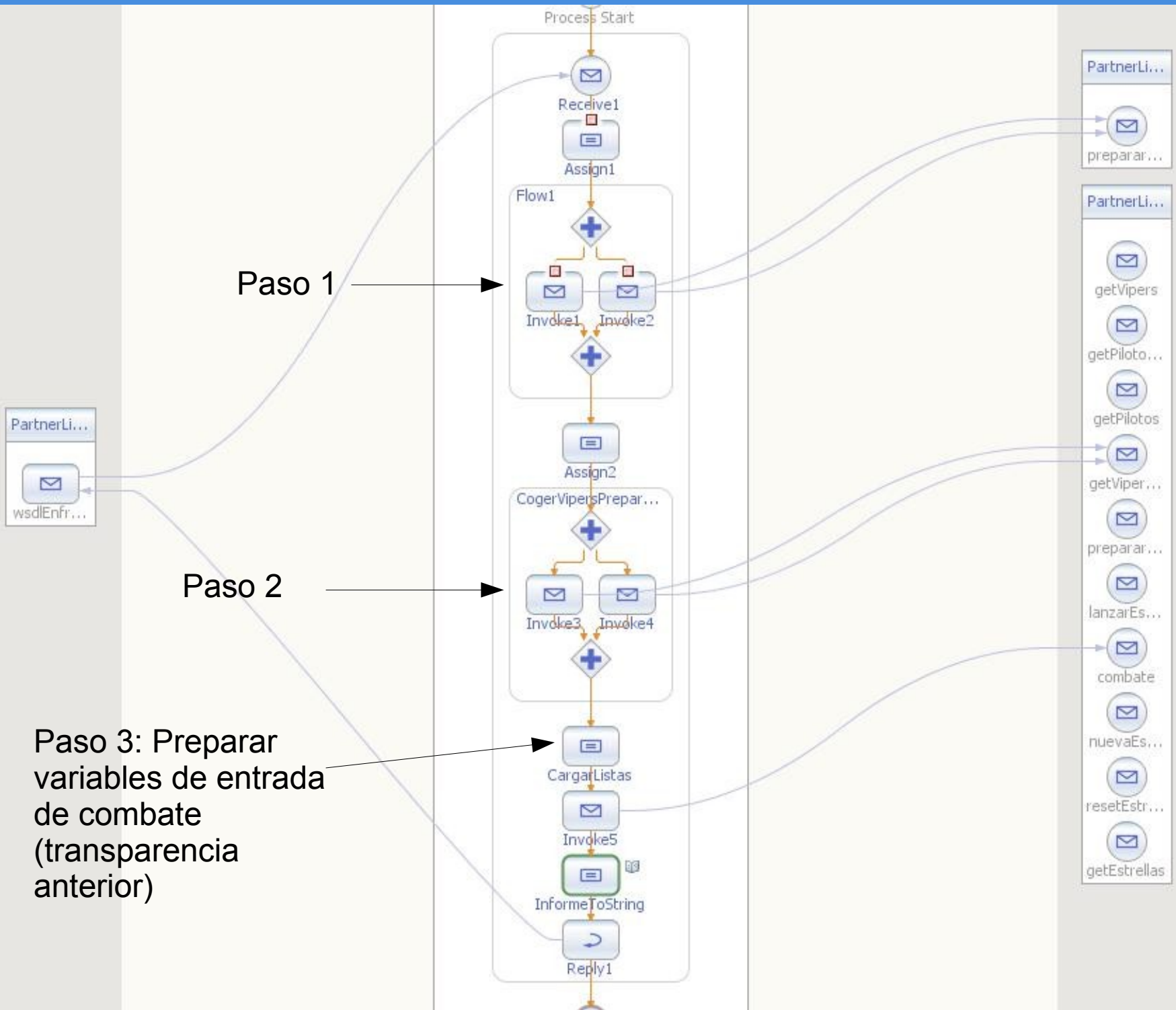
Paso 2 y 3

Llamamos a "getVipersPreparadas" y luego con estas vipers llamamos a "combate":



Paso 2 y 3: Convertir informe a String





Y debería funcionar...

- ```
<?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body><m:wsdIenfrentamientoOperationResponse xmlns:m="http://j2ee.netbeans.org/wsdl/wsdIenfrentamiento">

<informe xmlns:msgns="http://j2ee.netbeans.org/wsdl/wsdIenfrentamiento" xmlns="">true1987Vipper id(19)pilotada por Gaius Baltar dispara a Vipper id(87)pilotada por Comandante Adama destruyendolofalse1988Vipper id(19)pilotada por Gaius Baltar dispara a Vipper id(88)pilotada por Laura Roslin pero falla!
true8819Vipper id(88)pilotada por Laura Roslin dispara a Vipper id(19)pilotada por Gaius Baltar destruyendolo BLA...BLA... EL bando azul (segundo) gana el comabte.</informe>

</m:wsdIenfrentamientoOperationResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

# Errores a tener en cuenta

- Si hay acceso denegado al hacer *Clean and Build* ir a la gestión de *Composite Applications* (página 12 del tutorial) y darle a *undeploy* de la aplicación que se queja. Si se sigue quejando cerrar el servidor.
- Si al testear el segundo BPEL en el SOAP sale algo como `<detail xmlns=""><detailText>BPCOR-6135:A fault was not handled in the process scope;...` hay que volver a desplegar el primer BPEL (testearlo antes para comprobar que funciona)

# Mejoras (si hay tiempo)

Añadir Fault Handlers

Modificarlo para que en vez de devolver un *xsd:string* devuelva directamente un *InformeDeCombate*

# Listo!

- Ahora Cáprica 6 (la de la izquierda) puede inyectar el virus al sistema informático de Baltar para contagiar toda la Flota Colonial.

