

# Desarrollo de software en sistemas distribuidos

Curso 2019



# Desarrollo de software en sistemas distribuidos

Orquestación y Coreografía  
BPEL

# [Una conclusión importante acerca de SOA y WS]



Un error común es que una SOA es una nueva versión de Servicios Web

La distinción entre servicios SOA y Servicios Web se encuentra en sus respectivos diseños.

- SOA define un modelo para la ejecución de un determinado proceso.
- Los Servicios Web facilitan la aplicación táctica del modelo SOA. Son sólo una de las muchas maneras en las que una SOA pueden aplicarse.

# [Impacto de SOA en la infraestructura de IT]

El beneficio de SOA es que las aplicaciones pueden ser construidas por la combinación de servicios interoperables y estos servicios pueden reutilizarse.

Esto conlleva los siguientes riesgos en una infraestructura de IT:

- o Tiempos de respuesta malos para los usuarios y los procesos de negocio
- o Pérdida de niveles de servicio para funciones críticas del negocio
- o El incumplimiento con la industria y las regulaciones gubernamentales
- o Infracciones de seguridad
- o Insuficiente gestión de los servicios
- o Desafíos para el testing

# Impacto de SOA en la infraestructura de IT

Mientras que históricamente la infraestructura de IT constituyó el apoyo a las aplicaciones, con SOA, su atención se enfoca hacia la gestión de los servicios que prestan apoyo a los procesos de negocio y, por tanto, los resultados del negocio

# [ Enterprise Service Bus (ESB) ]

En las aplicaciones tradicionales, todos los pasos para completar una transacción están dentro de la aplicación con una simple conexión punto a punto.

Con SOA, la aplicación de procesos de negocio inicia la transacción y el ESB provee la mensajería, transformación de datos y el enrutamiento inteligente.

El ESB cumple las siguientes funciones:

- o Ruteo de mensajes entre servicios
- o Conversión de protocolos de transporte entre solicitante y servicio
- o Transformación de formatos de mensajes entre solicitante y servicio
- o Manejo de los eventos de negocio desde fuentes diversas

# Servidor de procesos

El ESB es dirigido por el servidor de los procesos de negocio donde sus procesos de negocio están definidos y gestionados.

El servidor de procesos **gestiona y orquesta** las solicitudes a través del ESB para servicios independientes y poder así completar los procesos de negocio.

Los flujos de procesos de negocio son utilizados por el servidor de proceso para representar las interdependencias entre aplicaciones.

# [ Orquestación y coreografía de los procesos de negocio ]

Los procesos de negocio atraviesan la estructura organizativa y definen sus reglas independientemente del proceso.

Los servicios resuelven funcionalidades concretas requeridas dentro de cada unidad organizativa y se componen para realizar los procesos de negocio a través de su orquestación y coreografía.



# Comparación de conceptos de orquestación y coreografía

	<b>Orquestación</b>	<b>Coreografía</b>
<i>Objetivo</i>	Componer servicios para cumplir con un proceso de negocio dentro de una organización	Componer servicios para colaboración entre organizaciones
<i>Modelo</i>	Jerárquico. Pregunta-Respuesta	Peer - to -Peer
<i>Enfoque</i>	Componer servicios y el orden en que son ejecutados para alcanzar el objetivo de un proceso de negocio	Definir la manera en que múltiples partes colaboran para conformar una transacción de negocio
<i>Fundamento</i>	Constituye un servicio en sí mismo	Define la interacción del negocio

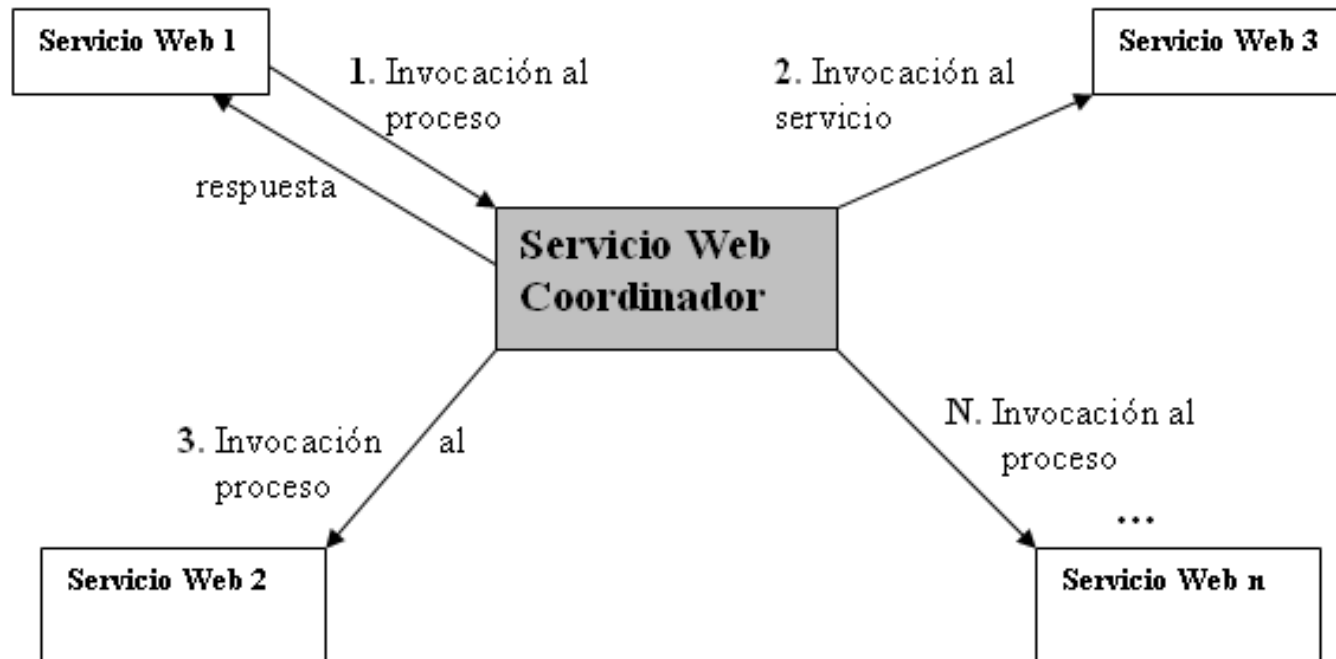
# [ BPEL (Business Process Execution Language) ]

BPEL es un lenguaje basado en XML que soporta la tecnología utilizada en los Servicios Web, incluyendo protocolos SOAP, WSDL, UDDI, entre otros.

Un proceso BPEL determinará el orden de invocación que se llevará a cabo con los Servicios Web brindando operaciones básicas de condición, iteración y asignación de variables como cualquier lenguaje de programación pero con una orientación hacia el control de procesos de negocios.

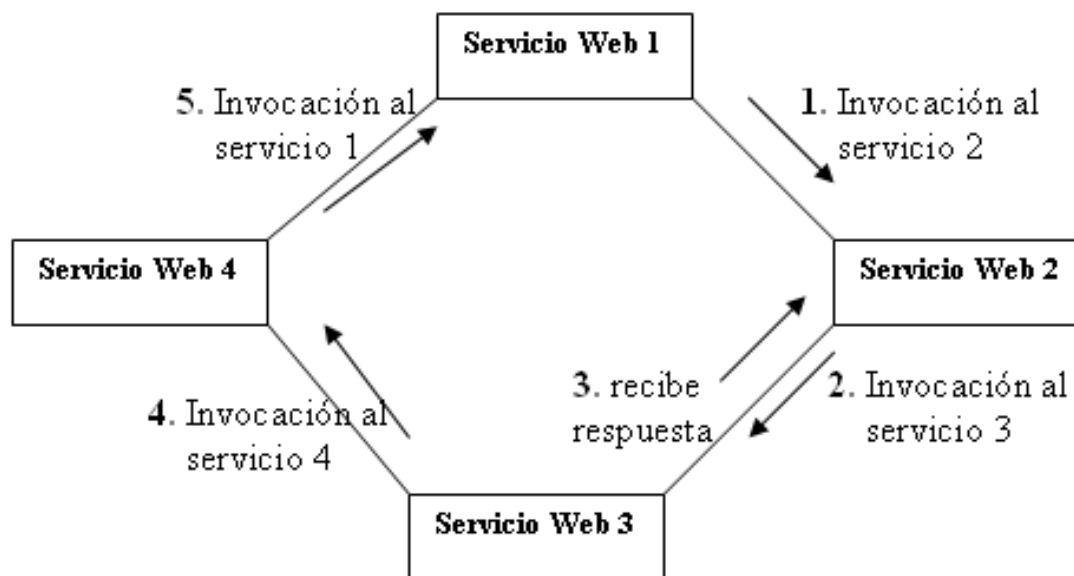
BPEL4People es una extensión que considera interacción humana.

# Orquestación de Servicios Web con BPEL



En la orquestación un proceso central tomará el control de los Servicios Web para luego coordinar la ejecución de las operaciones expuestas por los mismos.

# Coreografía de Servicios Web con BPEL



En la coreografía no existe el rol del coordinador central, en su lugar cada Servicio Web conocerá exactamente cuando ejecutar sus operaciones y con quien interactuar. Los participantes conocen la lógica del proceso, operaciones a ejecutar, mensajes a intercambiar y en que momento ejecutarse.

# Secciones de un documento BPEL

En un documento BPEL existen cuatro secciones principales:

- <Variables> define las variables utilizadas por el proceso, en base a un esquema de tipos y mantienen el estado del proceso a lo largo del intercambio de mensajes.

- <partnerLinks> define las diferentes partes que interactúan con el proceso de negocios. Cada elemento partner link tiene asociado un partner link type y un nombre de rol.

Esta información identifica la funcionalidad que debe ser provista por el proceso de negocios e implementada por el servicio para que la interacción sea exitosa.

# [ Secciones de un documento BPEL ]

- `<partnerLinkType>` Permiten describir el rol de los Servicios Web clientes y el proceso mismo involucrados dentro del proceso BPEL. Un `partnerLink` debe tener al menos un `partnerLinkType` y como máximo dos.
- `<faultHandlers>` contiene los manejadores de excepciones que se ejecutarán en caso de que ocurra una excepción en la invocación a un servicio.

# [Tipos de procesos BPEL]

BPEL soporta dos tipos diferentes de procesos de negocios:

- **Procesos ejecutables** los cuales nos permiten especificar detalles exactos del proceso de negocios. Estos procesos pueden ser ejecutados por un motor de ejecución.
- **Protocolos de negocios abstractos**, en los cuales únicamente se define el intercambio de mensajes entre las partes. Aquí los procesos no incluyen detalles internos de procesamiento y tampoco son ejecutables.

# Composición de procesos BPEL

La estructura de un proceso de negocios en BPEL consta de pasos, donde cada uno de estos pasos es conocido como una actividad.

Las actividades de un proceso se clasifican en:

- Primitivas
- Estructuradas

Dentro del primer grupo se encuentran:

- Invocación de Servicios Web
- Espera de solicitudes desde un cliente
- Generación de una respuesta sincrónica
- Manipulación de datos
- Declaración de errores y excepciones
- Esperar por una cantidad de tiempo
- Terminar el proceso de ejecución



# Actividades estructuradas en BPEL

Las actividades estructuradas permiten la combinación de actividades primitivas.

En este grupo se encuentran:

- Ejecución secuencial (<sequence>), nos permite declarar un conjunto de actividades las cuales se ejecutaran de forma secuencial.
- Ejecución paralela (<flow>) igual al anterior solamente que las invocaciones se realizaran en paralelo.
- Selector múltiple (<switch>) implementa condicional sobre la siguiente actividad a ejecutar.
- Iterador (<while>) permite iterar sobre un conjunto de actividades.

# Actividades estructuradas en BPEL

- Selección de ejecución (<pick>) permite seleccionar un camino de ejecución.
- Fallas (<throw>) permite lanzar excepciones.
- Manejador de excepciones (<faulthandler>) permite tratar las excepciones que se producen.
- Determinar alcance <scope> permite delimitar el alcance de un grupo de actividades.
- Terminar <terminate> permite finalizar la ejecución de un proceso de negocio.

# Diferencia entre BPEL y los lenguajes de programación

BPEL es portable entre plataformas.

Un proceso BPEL puede ser ejecutado sobre diferentes motores de ejecución lo cual es muy importante en el desarrollo B2B.

BPEL está especialmente diseñado para el desarrollo de procesos  
Esto implica que hará eficiente la ejecución de procesos de larga duración (ya sea minutos, horas, días),

BPEL también soporta una característica de “deshacer” actividades previas del proceso de negocio.

Es posible definir manejadores para estos casos indicando el alcance de los mismos dentro del proceso de ejecución. Estas características son conocidas como Long-Running-Transactions (LRT).

# Exposición e invocación de procesos BPEL

Los procesos BPEL se exponen a los clientes como WS, por lo tanto requieren que se defina su WSDL.

Para comenzar el proceso, el cliente deberá invocar una operación del mismo que normalmente será una sola. Cada invocación genera una nueva instancia.

# Procesos BPEL sincrónicos y asincrónicos

El proceso BPEL puede ejecutarse en forma sincrónica (el cliente espera la respuesta) o asincrónica (el cliente continua trabajando mientras llega la respuesta invocando a ese cliente).

La diferencia entre ambos:

- 1- Como se retornan los resultados. En ambos casos se esperan en la actividad *<receive>* o *<pick>*. En el síncrono se retorna con *<reply>* y en el asíncrono se usa *<invoke>* para invocar al cliente y retorna al *port type*
- 2- Como se declara la interfaz. En las operaciones síncronas se especifican por mensajes *<input>* y *<output>* mientras que las asíncronas requieren definición de dos *port types*, uno para la invocación inicial y otro para la respuesta.

# Composición de un proceso BPEL

```
<bpel:import namespace="http://service.romandk.com/" location=
"http://127.0.0.1:8080/Servicio-Web-C?wsdl" importType="http://schemas.xmlsoap.org/wsdl/"
></bpel:import>
    <bpel:import namespace="http://service.romandk.com/" location=
"http://127.0.0.1:8080/StoredProcedureService?wsdl" importType="http://schemas.xmlsoap.org/wsdl/"
></bpel:import>
    <bpel:import namespace="http://servicio.romandk.com/" location=
"http://127.0.0.1:8080/InformeTributario?wsdl" importType="http://schemas.xmlsoap.org/wsdl/"
></bpel:import>
    <bpel:import namespace="http://servicio.romandk.com/" location=
"http://127.0.0.1:8080/InformeTerritorial?wsdl" importType="http://schemas.xmlsoap.org/wsdl/"
></bpel:import>
    <bpel:import location="orquestacionArtifacts.wsdl" namespace="http://eclipse.org/bpel/sample"
importType="http://schemas.xmlsoap.org/wsdl/" />

    <bpel:partnerLinks>
        <bpel:partnerLink name="client" partnerLinkType="tns:orquestacion" myRole=
"orquestacionProvider" />
        <bpel:partnerLink name="territorial" partnerLinkType="tns:territorialPLT" myRole=
"territorialServer" partnerRole="territorialClient"></bpel:partnerLink>
        <bpel:partnerLink name="tributario" partnerLinkType="tns:tributarioPLT" myRole=
"tributarioServer" partnerRole="tributarioClient"></bpel:partnerLink>
        <bpel:partnerLink name="caratulacion" partnerLinkType="tns:caratularPLT" myRole=
"caratularServer" partnerRole="caratularClient"></bpel:partnerLink>
        <bpel:partnerLink name="categoria" partnerLinkType="tns:categoriaPLT" myRole=
"categoriaServer" partnerRole="categoriaClient"></bpel:partnerLink>
    </bpel:partnerLinks>
```

## *import*

Lista todos los documentos *wsdl* de los servicios web utilizados en el proceso BPEL

## *PartnerLink*

Lista los servicios participantes en el flujo del proceso de negocio, los roles que cumplen y *partnerLinkType* asociados

# Composición de un proceso BPEL

```
<bpel:variables>
  <bpel:variable name="input" messageType="tns:orquestacionRequestMessage"/>
  <bpel:variable name="output" messageType="tns:orquestacionResponseMessage"/>
  <bpel:variable name="territorialIn" messageType=
"ns1:ObtencionInformeTerritorial_getInformeTerritorial"/>
  <bpel:variable name="territorialOut" messageType=
"ns1:ObtencionInformeTerritorial_getInformeTerritorialResponse"/>
  <bpel:variable name="tributarioIn" messageType=
"ns1:informeTributarioService_obtenerInformeTributario"/>
  <bpel:variable name="tributarioOut" messageType=
"ns1:informeTributarioService_obtenerInformeTributarioResponse"/>
  <bpel:variable name="deudasize" type="ns2:int"/>
  <bpel:variable name="indexdeuda" type="ns2:int"/>
  <bpel:variable name="caratulaIn" messageType=
"ns3:StoredProcedureCaratular_caratularTramite"/>
  <bpel:variable name="caratulaOut" messageType=
"ns3:StoredProcedureCaratular_caratularTramiteResponse"/>
  <bpel:variable name="categoriaIn" messageType="ns3:InvocaCC_invocaservicioCC"/>
  <bpel:variable name="categoriaOut" messageType="ns3:InvocaCC_invocaservicioCCResponse"/>
</bpel:variables>
```

## Variables

Sirven para mantener el estado de los procesos de negocio. permiten almacenar mensajes que mantendrán los valores recibidos y enviados por los servicios. Los tipos de cada variable pueden ser mensajes WSDL, un tipo de dato simple definido en XML o un elemento XML.

# Composición de un proceso BPEL

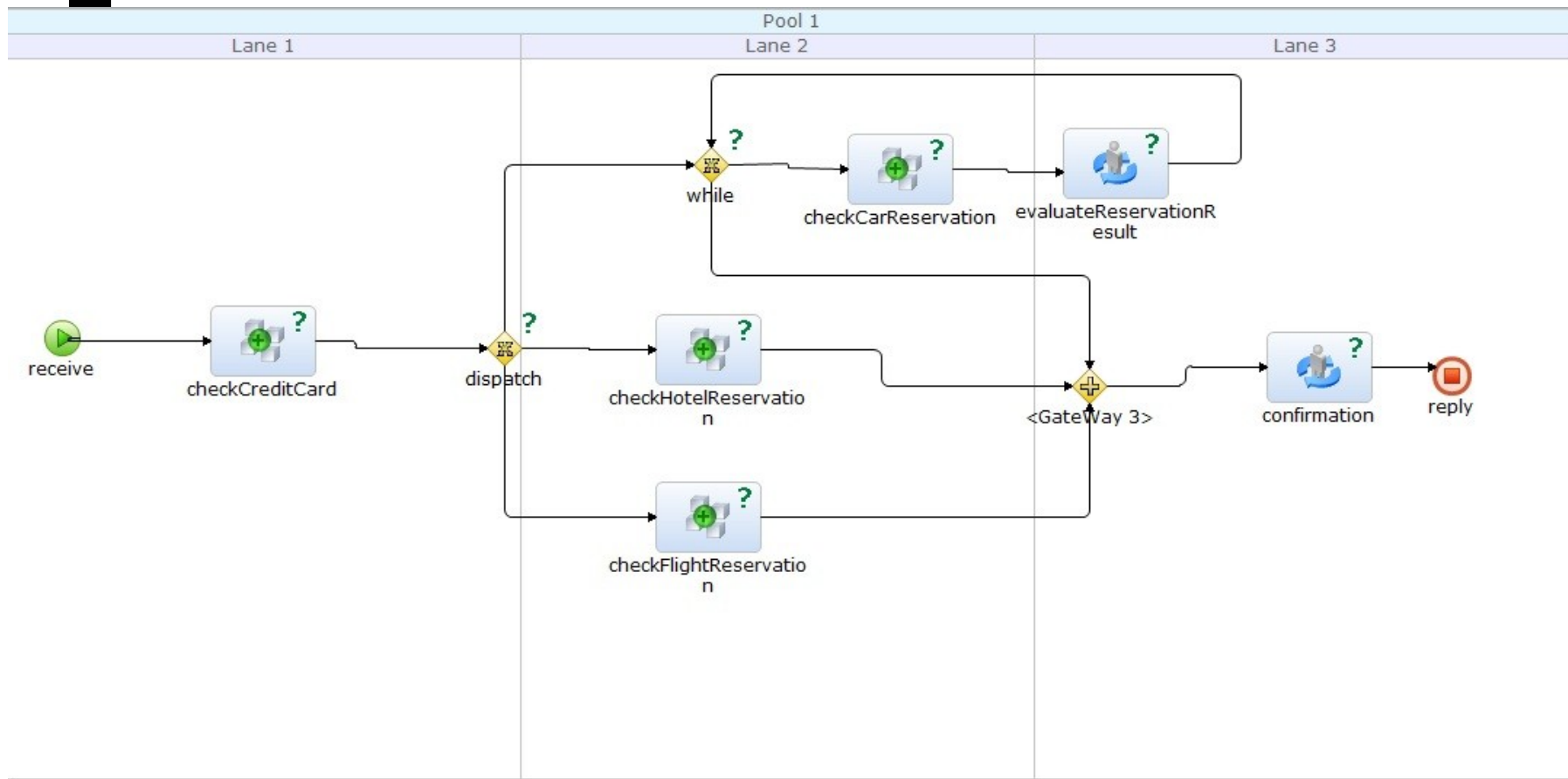
```
<bpel:sequence name="main">
  <bpel:receive name="receiveInput" partnerLink="client" portType="tns:orquestacion"
    operation="process" variable="input" createInstance="yes" />
  <bpel:assign validate="no" name="requerimiento">
    <bpel:copy>
      <bpel:from>
        <bpel:literal xml:space="preserve"><tns:PruebaTramiteResponse xmlns:tns="http://eclipse.org/bpel/tramite"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tns:result></tns:result>
</tns:PruebaTramiteResponse>
</bpel:literal>
      </bpel:from>
      <bpel:to variable="output" part="payload"></bpel:to>
    </bpel:copy>
    <bpel:copy>
      <bpel:from part="payload" variable="input">
        <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
          <![CDATA[tns:input/tns:arg0]]>
        </bpel:query>
      </bpel:from>
      <bpel:to part="payload" variable="output">
        <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
          <![CDATA[tns:result/tns:arg0]]>
        </bpel:query>
      </bpel:to>
    </bpel:copy>
  </bpel:assign>
  <bpel:if name="Estado viabilidad ?">
    <bpel:condition><![CDATA[${output.payload/tns:result/tns:arg0/tns:tramite/tns:estado = 2}]]></bpel:condition>
    <bpel:sequence name="viabilidad legal"><bpel:assign validate="no" name="Assign">
      <bpel:else>
    </bpel:if>
    <bpel:reply name="replyOutput" partnerLink="client" portType="tns:orquestacion"
      operation="process" variable="output" />
  </bpel:sequence>
```

## Lógica del proceso

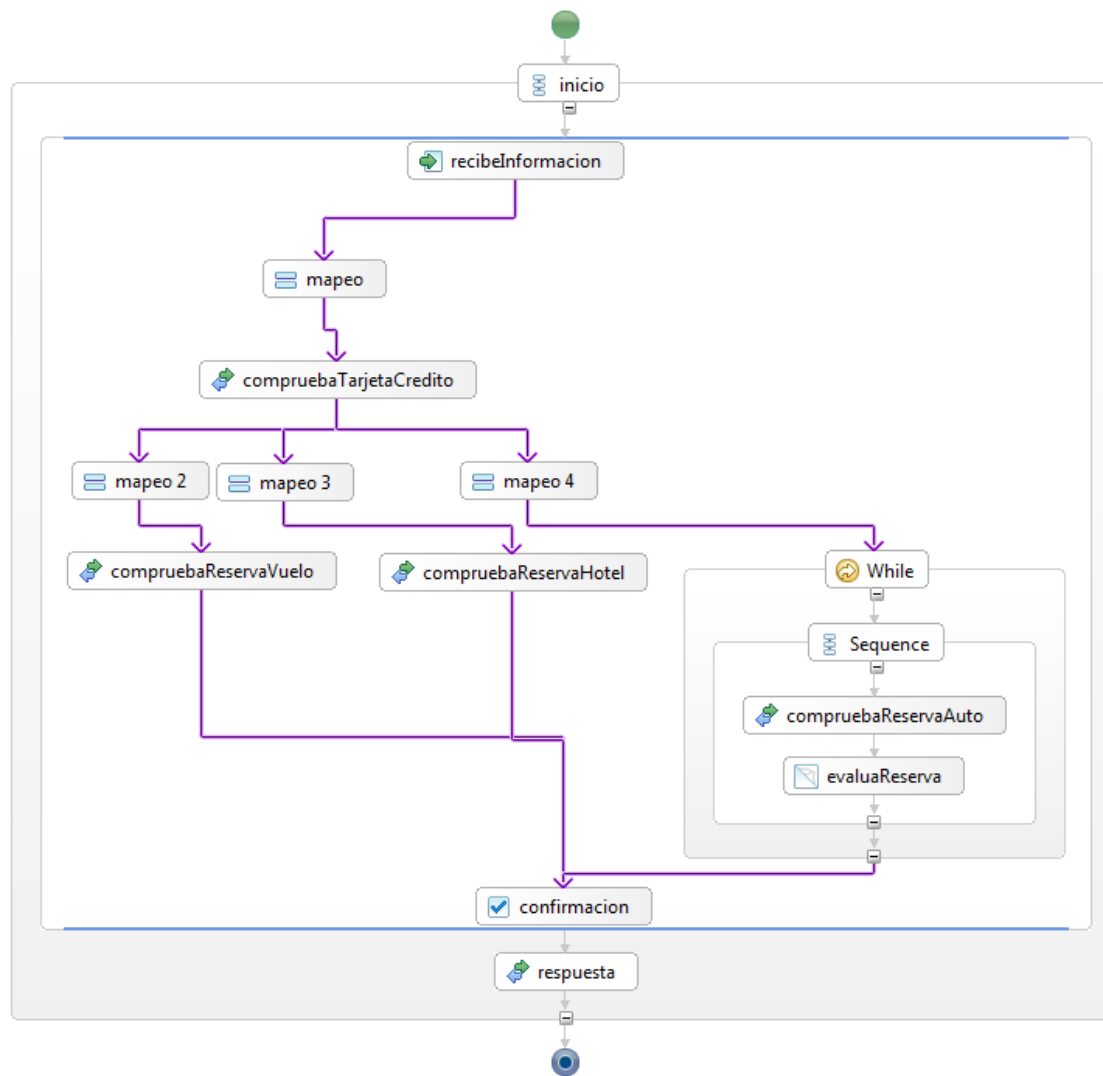
Dentro del tag **<bpel:sequence>** se define la lógica de proceso de negocios, y la declaración de los receive, invoke, assign, ... de los procesos de negocios



# Proceso de Reserva de Viaje en BPMN



# Proceso de Reserva de Viaje en BPEL



# Proceso de Reserva de Viaje en BPEL

