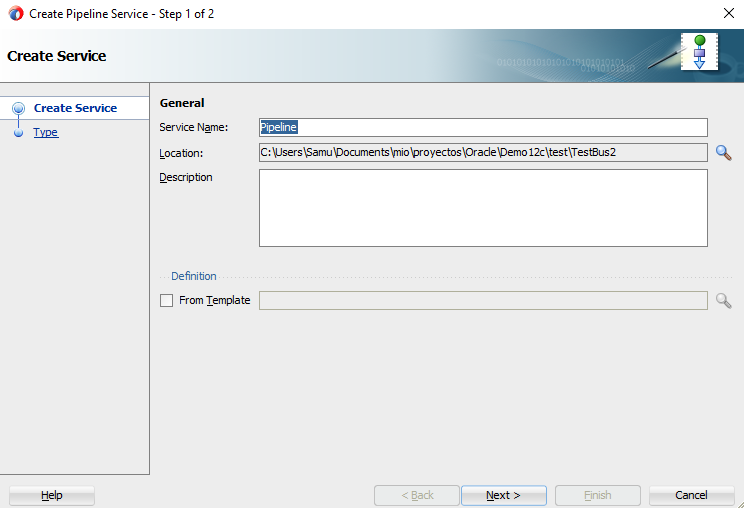
Crear pipeline en jdevelop

\* Para este tutorial, debe haber creado el servicio de base de datos

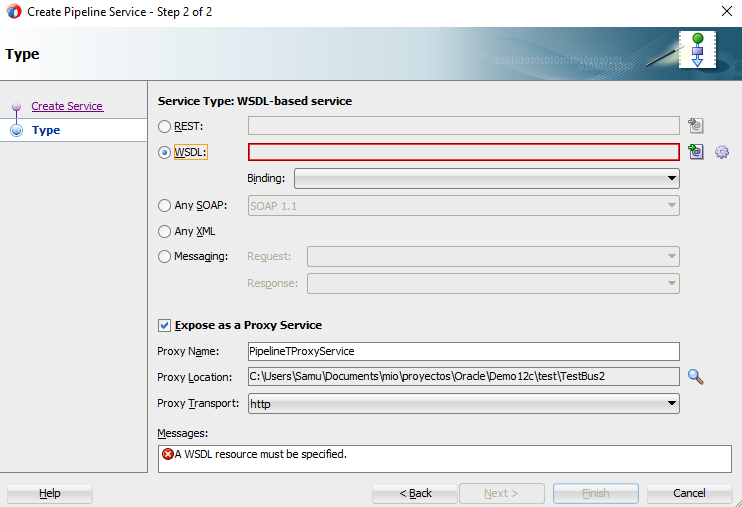
\* El pipeline nos permite el ruteo y tranformación de los datos.z

1. Partiendo desde el archivo de composición (), buscamos en el menú de la derecha, donde se encuentran los componentes, el pipeline ().
2. Presionamos la opción de pipeline y lo arrastramos a la sección del medio para poder añadirlo a nuestra composición.
3. Nos saldrá una ventana con diferentes campos a llenar:

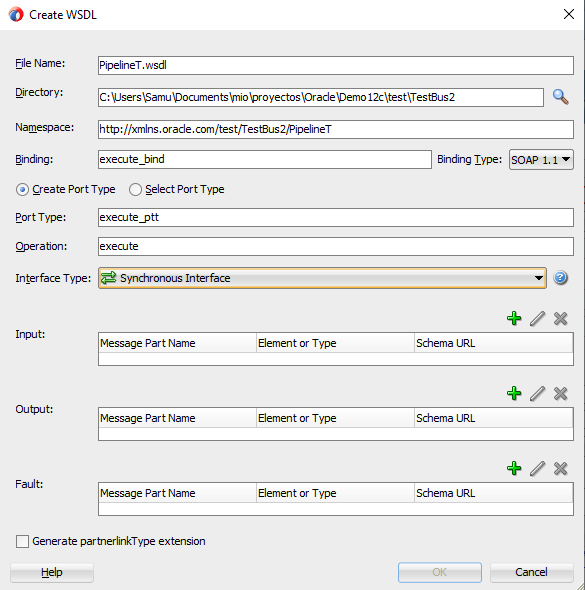


* 1. “**Service Name**”: Ponemos un nombre descriptivo para el pipline. De ser posible deja la palabra clave Pipeline para identificar el componente.
  2. “**Location**”: Si no lo modificamos, entonces se guardará en la raíz de nuestro proyecto. Para hacer el cambio de directorio:
     1. Damos en la lupa de la derecha (al lado del campo).
     2. Seleccionamos la carpeta con el nombre Pipeline en nuestro caso.
     3. Damos “ok”.
  3. “**From Template**”: Puede que el arquitecto del sistema haya creado una plantilla para los “**pipeline**” y tengas que usarla. En ese caso, marchas la casilla de la izquierda y en la lupa buscas la plantilla a utilizar.

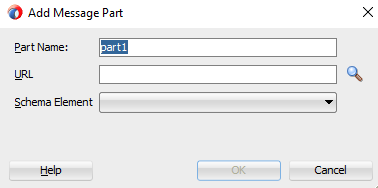
1. Presionamos “**Next**” y vamos a la ventana de “**Type**”. Aquí configuraremos los mensajes de entrada y salida de nuestro servicio. Además de si queremos crear una “capa de presentación” para el pipeline.



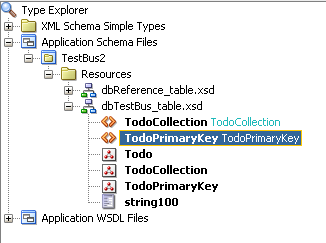
1. Ahora tendríamos que realizar la configuración de los mensajes de entrada y salida. Para ello tenemos varias opciones:
   1. “**REST**”: Configuración de las variables de entrada y salida mediante json.
   2. “WSDL”: Configuración de las variables de entrada y salida mediante xml.
   3. “**Any SOAP**”: Cualquier petición SOAP que reciba.
   4. “**Any XML**”: Cualquier petición XML que reciba.
   5. “**Messaging**”: Configuración manual del mensaje de entrada y salida.
2. En nuestro caso vamos a configurar el “**WSDL**” con los datos creados mediante el servicio de base de datos.
3. Presionamos el icono de tuerca () al lado de “**WSDL**” para realizar crear el archivo que nos permitirá definir los datos para la petición del “**Pipeline**”.



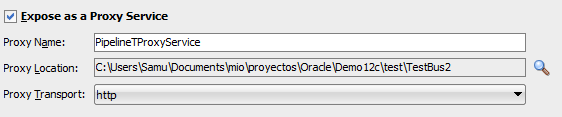
1. Aquí tenemos varias opciones.
   1. “**File Name**”: Nombre que le pondremos al archivo. Intenta ser descriptivo en el nombre.
   2. “**Directory**”: Por defecto, lo guardará en la misma carpeta donde se encuentre el Pipeline. Si queremos trabajar más organizado, creamos una carpeta que se llame “**WSDL**” y lo guardamos ahí.
   3. “**Namespace**”: Nombre de dominio a utilizar para permitir imports desde otros archivos.
   4. “**Binding**”: Tipo de comportamiento que vamos a realizar. Ser descriptivo con esta parte.
   5. “**Binding Type**”: Que vamos a utilizar para la petición SOAP o XML.
   6. “**Port**”: Tipo de puerto a utilizar. Es un nombre descriptivo para el Pipeline.
   7. “**Operation**”: Aquí definimos el nombre de la operación a realizar. Puede ser “**selectByPrimaryKey**” o “**ObtenerID**” como desee.
      1. Pero tenga en cuenta que si pone el mismo nombre del método que va a utilizar, puede no configurar en el pipeline posteriormente el método del servicio DB a utilizar.
   8. “**Interface Type**”: Esto es importante para definir el sentido de trabajo del pipline. En este caso vamos a utilizar el “**Synchronous Interface**” para decir que recibiremos y devolveremos un mensaje.
      1. Al seleccionar esta opción, se nos habilitaran tres cuadros de opciones.
   9. “**Input**”: Aquí declaramos el mensaje en entrada esperamos. Para crear uno nuevo, presionamos en el 



* + 1. “**Part Name**”: El nombre que identifica al mensaje. Trata, de ser posible, incluir la palabra “**input**” para identificar que es un mensaje de entrada.
    2. “**URL**”: Buscaremos el esquema creado para definir la estructura de los datos. Cuando creamos el servicio de BD, este nos creó una carpeta “**resource**” con el esquema básico.



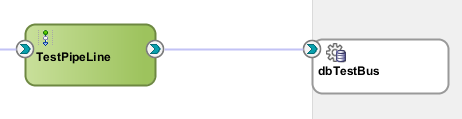
* + 1. Damos en la lupa y este nos despliega la estructura del proyecto para buscar el esquema.
    2. Dentro de este, buscamos el “**element**” que va a ser nuestro mensaje de entrada, en este caso decidimos que fuera el “**primary key**”; por lo que debemos seleccionar “**TodoPrimaryKey**”. Presionamos “**OK**” para hacer válida la selección.
    3. Presionamos “**OK**” para salir de la ventana de selección de mensaje de entrada.
  1. Ahora nos toca crear el mensaje de salida. Seguimos los mismos pasos anteriormente descritos, pero:
     1. Intenta que el “**Part**” tenga la palabra “**output**” para identificarlo como mensaje de salida.
     2. En la “**URL**” seleccionaremos el “**element**” a utilizar para el mensaje de salida; en este caso se decide utilizar el “**TodoCollection**”.
  2. El tercer campo es por si tenemos un esquema para mostrar los errores de manera personalizada; en este caso no lo necesitamos por lo que damos “**OK**” para terminar con la creación de los mensajes.
  3. En el mismo servicio de creación del Pipeline podemos crear la “**capa de aplicación**” o “**proxy**”.



\* Si no quieres exponer el pipeline de forma automáticas, desmarca la casilla de arriba.

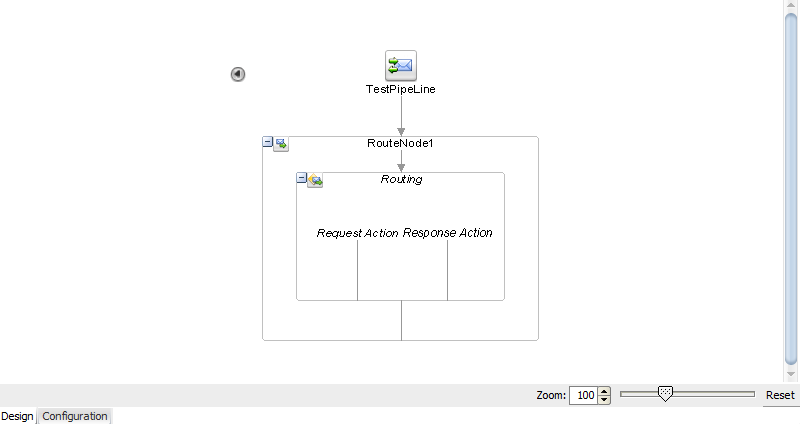
* + 1. “**Proxy Name**”: El nombre que recibirá el proxy. De ser posible incluye el nombre “**proxy**” para identificar el componente.
    2. “**Proxy Location**”: Donde se va a guardar el elemento una vez creado. Para mayor organización guárdalo en la carpeta proxy (depende del proyecto donde estés trabajando)
    3. “**Proxy Transport**”: Forma de exponer el “**pipeline**”. Existen varias opciones, pero las que nos interesan ahora son dos:
       1. “**http**”: Nos permite dar acceso externo a nuestro servicio.
       2. “**sb**”: En caso que necesitemos el servicio para su uso dentro de otro y no queramos exponerlo al usuario, usamos esta opción.
  1. Presionamos “**Finish**” y se nos creará el pipeline conectado al “**proxy**”

1. Por defecto, cuando se crea el pipeline, este no viene conectado con el servicio de base de datos por lo que presionamos en la flecha verde () que sale de pipeline y la unimos con el bus de servicio.

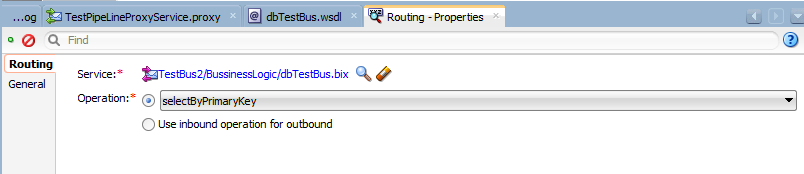


\* Ahora vamos al paso de configurar el método a utilizar por el pipeline. Como se mencionó anteriormente, si en la creación del pipeline, en el campo “**operation**” pusimos exactamente el nombre del método a utilizar para conseguir los datos del servicio; no tenemos que hacer esta configuración.

1. Damos doble clic sobre el icono de pipeline para acceder a su configuración. Nos saldrá una pestaña parecida a la siguiente:



1. Seleccionamos el que dice “**Routing**” que es el encargado de aceptar los datos de entrada y devolverlos al usuario. Nos saldrá debajo o en la misma pestaña (depende de cómo lo tengas configurado) las propiedades de ruteo.



1. En este caso, el nombre puesto a la operación, no coincide con el del servicio, por lo que debemos seleccionar la primera opción del radio “**Operation**” y en el desplegable, seleccionar el método a utilizar.
2. Ya con esto hecho terminamos y solo nos queda desplegar para que se suban todos los cambios al “**weblogic**”.

\* Si queremos exponer por REST el pipeline ya creado con XML, podemos dar clic derecho sobre él en el “**composite**” y decir que lo exponga por “**REST**”.