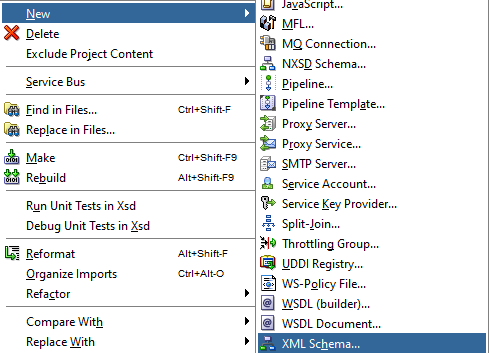
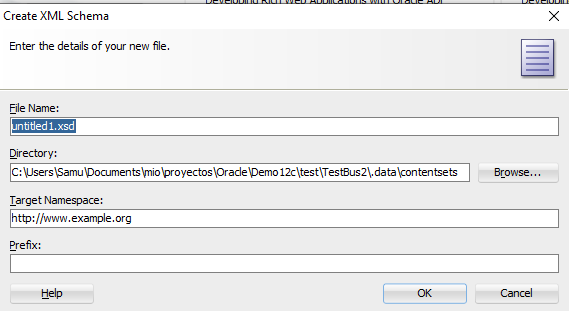
Crear archivo XSD

\* Para este tutorial, vamos a crear un archivo de tipo “**Schema**”

1. Creamos nuestra carpeta (**ver extra - 1**) con el nombre “**XSD”**, “**Schema”** o como se defina en el proyecto.
2. Nos paramos sobre la carpeta y damos **clic derecho > New > XML Schema**



1. Esto nos abrirá una ventana para ingresar los datos básicos del “**Schema**”:

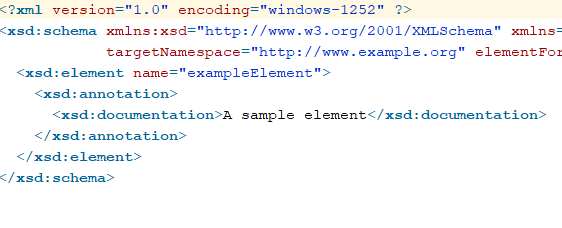


* 1. **File Name**: Nombre para identificar el archivo dentro de la jerarquía de elementos.
  2. **Directory**: Directorio donde se va a guardar el archivo “**XSD**” a crear.
     1. Para cambiar la ubicación damos en [] y seleccionamos donde queremos guardarlo.
  3. **Target Namespace**: El valor del nombre del dominio para el archivo “**XSD**”
  4. **Prefix**: Si queremos que los datos internos creados comiencen con un prefijo como “**imp**” por ejemplo.

1. Una vez creado se nos van a presentar dos vistas .
   1. **Design**: Es la encargada de proporcionar una interfaz gráfica para construir el esquema mediante una estructura de árbol.

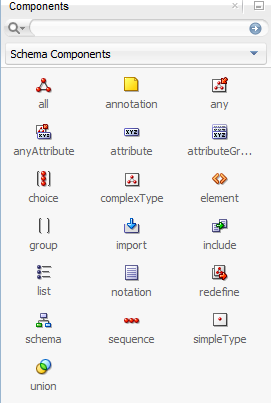


* 1. Soruce: Es la representación en código XML de la interfaz visual.

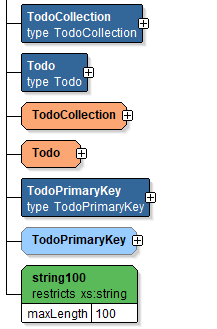


* 1. Cualquier cambio que se haga en una, se ve reflejado en la otra, por lo que podemos trabajar indistintamente entre ellas.

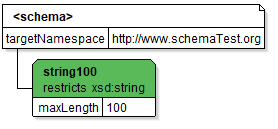
1. En la zona de la derecha, podemos ver el menú de componentes que podemos utilizar mediante la vista de diseño:



1. Para este ejemplo, vamos a crear un “**Schema**” similar al creado mediante el adaptador de base de datos:



* 1. Para conocer los componentes a utilizar mire el tutorial **extra - 2**

1. Primero vamos a crear el “**Simple Type**” para declarar el tipo de dato “**string100**”.
   1. Añadimos un “sipmle type” [] as esquema principal:
   2. 
   3. Para agregar el tipo de dato presionamos **clic derecho > set base type > string**



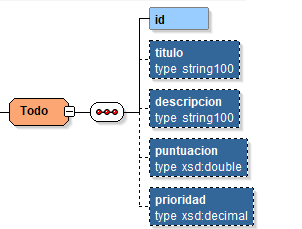
* 1. Para decir que tiene una longitud máxima de 100 caracteres clic derecho > Insert inside simpleType > maxLength.
     1. Nos abrirá una ventana y llenamos el campo que nos aparece.



1. Ya con esto tenemos un tipo simple. Ahora vamos a crear un tipo complejo para:
   1. **Todo** -> Similar a un objeto.
   2. **Todo Collection** -> Lista de objeto todo.
   3. **Todo Primary Key** -> Para crear un dato de entrada que sea solo el id.
2. **TODO**:
   1. Añadimos un “**Complex Type**” al “**Schema**” mediante [].



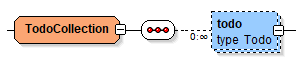
* 1. Ahora vamos a añadir elementos [], unidos por [] con el “Complex Type”, para conformar los atributos de nuestro objeto.



1. **Todo Collection**
   1. Ahora vamos a añadir un “**Complex Type**” para manejar listas mediante [].
      1. Primero creamos el “**Complex Type**”



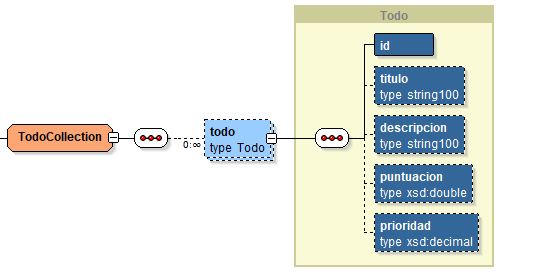
* + 1. Agregamos un “**Sequence**” seguido por un “**Element**”.



* + 1. Para agregar la cardinalidad vamos a “**properties**” del “**element**” **todo.**
       1. Dentro configuramos las opciones de (**minOccur**) a 0 y (**maxOccur**) a “**unbounded”**

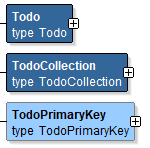


* + 1. Al terminar tendremos creado un tipo complejo de la siguiente forma:



1. **Todo Primary Key**
   1. Este es un tipo complejo que solo va a retener el criterio que vamos a usar para la búsqueda por llave primaria.
   2. En caso que queramos cambiarlo en algún momento, no tendríamos que ir de elemento en elemento. Con cambiarlo en el tipo complejo, los demás serán actualizados de forma automática.
   3. Quedaría de la siguiente forma:



1. Ya con los tipos complejos creados, podemos comenzar a declarar los elementos a ser utilizados para las llamadas del pipeline mediante el “**WSDL**”.
   1. Sería crear varios “**element**” con un “**Complex Type**” como tipo:
   2. 
   3. De esta forma ya tenemos los elementos bases para usar en archivos “**WSDL**”, “**Xquery**” y “**XSL**”

Ya tenemos creado nuestro archivo de “**Schema**” para trabajar.