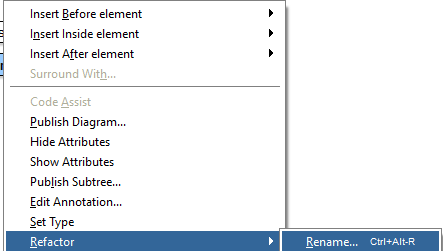
Componentes del XSD

\* En este documento veremos el uso de los principales componentes del diseñador visual del “**Schema**”.

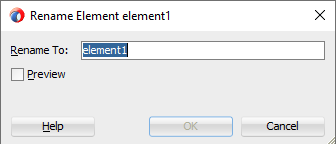
**Element**

****

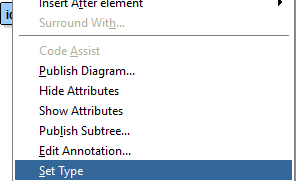
* Representa un dato básico de la estructura.
* Es comparable a una variable en cualquier lenguaje de programación.
* Si damos clic derecho sobre ella, podremos abrir un menú que nos permitirá realizar varios tipos de acciones.
  + Ahora nos interesa solo cambiarle el nombre y decir que tipo de datos es:
  + Cambiar Nombre (**Refactor > Rename**):



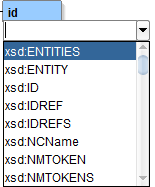
* + - Esto nos inicializa una ventana donde podremos cambiarle el nombre al componente.



* + - Damos [OK] y terminamos
  + Asignarle un tipo de dato:
    - **Clic derecho > Set Type**



* + - Esto nos habilita un [**Select**] debajo de elemento para seleccionar el tipo de dato.



* + - Lo seleccionamos y ya está, completamos la creación de un elemento

**Complex Type**



* Los tipos de datos complejos nos permiten crear objetos como en lenguajes de programación.
* Por si solo no hace nada, por lo que debemos agregarle más componentes para que pueda trabajar.

\* Si quieres simular la herencia, podemos utilizar la opción “Set Base Type” del menú contextual al dar clic derecho sobre el componente.

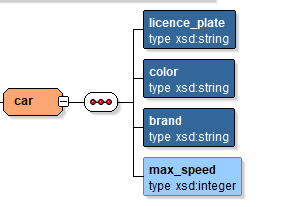
**Sequence**



* El componente de secuencia permite conectar diferentes componentes entre ellos, realizando una operación similar a la generalización.
* Un ejemplo que como quedaría acoplada a un **Complex Type**.



**Complex Type + Sequence + Element**

****

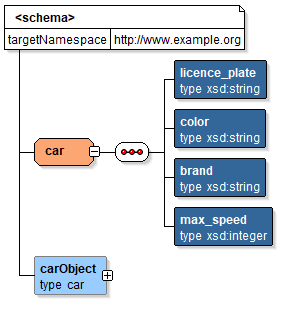
* Aquí simulamos la creación de una “Clase” llamada “**car**” que dentro tiene los atributos correspondientes a él.
* La creación de un tipo complejo, nos permite reutilizar “Tipos de datos” durante todo nuestro “**Schema**”

**Element with complex type**

* Creamos un “element” como se vio anteriormente y esta vez se le asigna el tipo de dato complejo creado.



* Sumando las dos partes, quedaría de la siguiente manera:



**Simple Types**

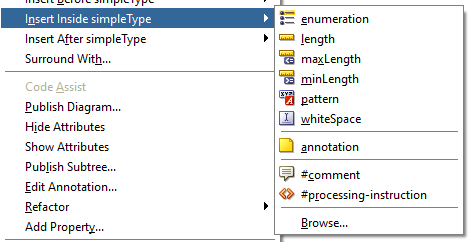
* Si podemos crear tipos complejos, significa que también se pueden crear simples.
* En este caso, arrastramos [****] hacia el “**Schema**”.
* Este, quedaría representado de la siguiente forma:



* En este caso vamos a decir que va a ser un objecto de tipo string con tamaño 100.
* Primero le ponemos un tipo de dato como lo hicimos anteriormente:



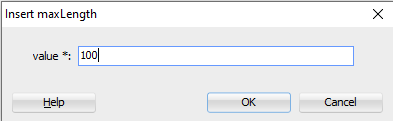
* Ahora vamos a añadirle la restricción de la cantidad de caracteres.
* **Clic derecho sobre el components > Insert Inside simpleType > maxLength**



\* Como vimos anteriormente, también se puede usar el menú contextual para añadir nuevos componentes partiendo desde el que tenemos seleccionado:



* Al seleccionar el valor “maxLength” nos saldrá una ventana para definir el tamaño deseado.



* Presionamos [**OK**] y el resultado final es el siguiente:



* Para asignar este tipo de dato a cualquier elemento seguimos los mismos pasos que en “**Complex type**”

**Lista de elementos mediante Complex Type**

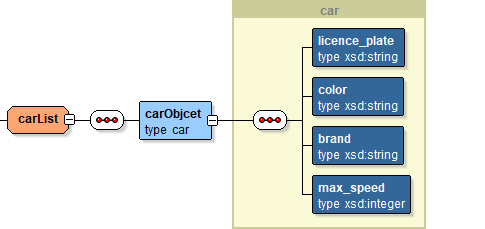
* Vamos a crear una “**Complex Type**” que almacene una lista de objetos. En este caso una lista de carros.
* Primero creamos el “**Complex Type**” de la misma forma que se hizo anteriormente.



* Ahora le asignamos el componente de “**Sequence**”.



* Le agregamos el elemento que será el valor a almacenar en la lista. En este caso creamos otro elemento de tipo “**Car**”.



* Ahora presionamos sobre **“carObject”** y vamos al apartado de propiedades. Dentro buscamos el apartado “**XML**”. Llenamos los campos (**maxOccurs**) y (**minOccurs**)



* Para no poner tope al máximo de cantidad de objeto, podemos desplegar el **[Select]** y seleccionamos la opción **“unbouded”.**
* Además podemos especificar la mínima cantidad de ocurrencia que queramos.



* Para convertir un elemento en nulleable, vamos a sus propiedades y dentro buscamos **(nillable)** y en el [**Select**] ponemos **true**. Y el campo (**minOccurs**) en 0 para permitir que no existan datos en ese campo.



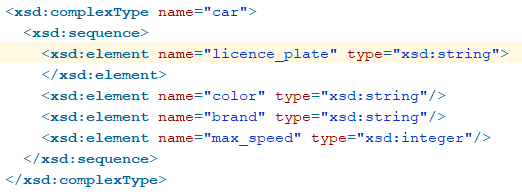
Ya con esto se tiene una base para conocer algunos de los componentes para crear los archivos **XSD**.

Todo lo que se hizo hasta el momento, se puede realizar mediante código. Ahora mostraremos por componente como se verían.

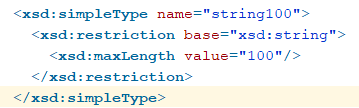
**Element**



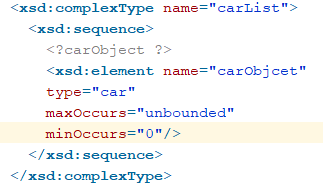
**Complex Type + Sequence + Complex Type**



**Simple Type**



**Lista**

****