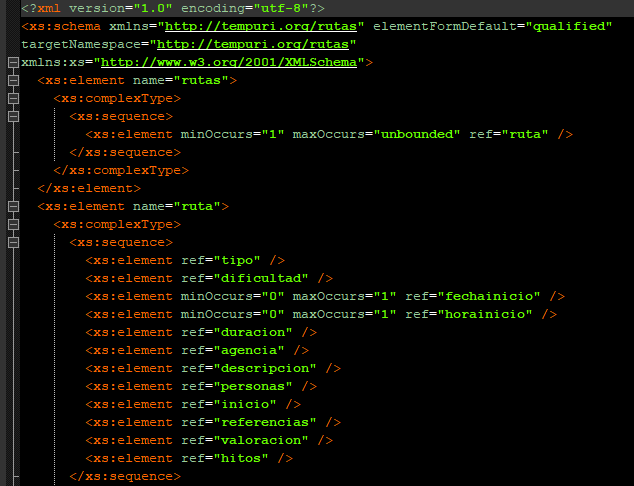
**Ejercicio 1 - Tarea 7**

**Software y Estándares para la Web**

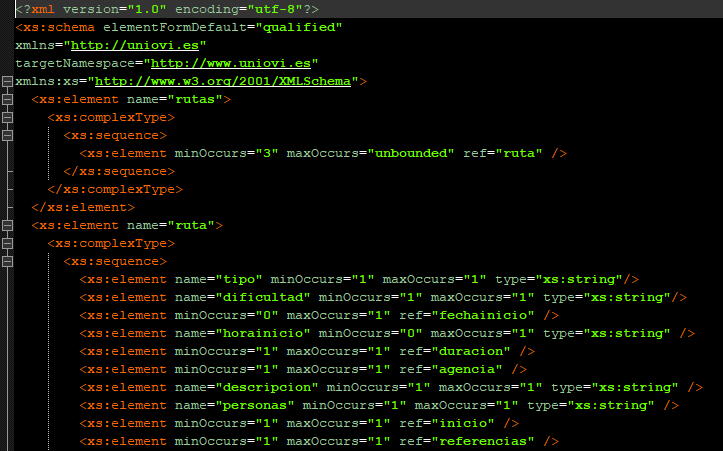
En este documento, se mostrarán algunas **imágenes** del *XML Schema* generado durante la **Tarea 4**, así como del *XML Schema* elaborado y modificado a partir del mencionado anteriormente en la **Tarea 5**, con el objetivo de comentar y dar a conocer las principales **diferencias**, dentro del dominio de nuestro problema.

Cabe recordar, para mi caso concreto, que el dominio del problema consiste en la validación de documentos XML bien formados para especificar **información relevante sobre rutas en moto**. Esto implica, por tanto, que se trabajará con **datos de diferentes tipos**: por ejemplo, con nombres y unidades de distancia; y por otro lado, con distancias y coordenadas para los diferentes hitos y ciudades recorridas.

A continuación, se compara el primer fragmento entre ambos *XML Schemas*:



Tarea 4

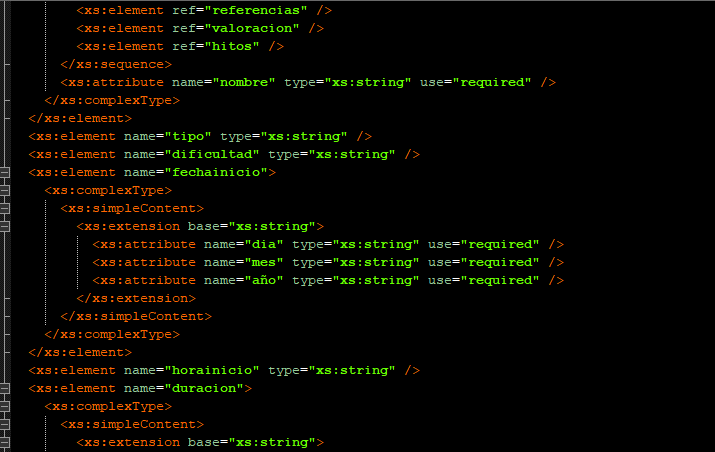
La primera diferencia notable, además de la especificación del xmlns y del targetNamespace, es que el *XML Schema* **generado automáticamente** **no es capaz de inferir el mínimo ni el máximo de ocurrencias** de cada elemento de forma ajustada a la especificación de nuestro dominio; algo que resulta totalmente lógico, dado que **fue generado a partir de la declaración de un DTD**, donde no es posible, por ejemplo, especificar un número mínimo de ocurrencias superior a 1 para el número de rutas integradas en el documento XML.

Tarea 5

Otra diferencia curiosa entre el generado y el modificado, es que **el XML Schema generado automáticamente tiende a declarar todos los elementos por separado**; es decir, los referencia en su primera aparición y los declara más adelante, independientemente de que tenga subelementos o atributos. Esto supone la formación de un *XML Schema* con más líneas de código, pero el mismo significado.

Otra desventaja del DTD frente a los *XML Schema*, es su **escasa flexibilidad de tipos** de datos. Al generar el archivo XSD a partir del DTD, lo más probable es que todos los tipos de los elementos y atributos aparezcan reflejados como xs:string.

Esto, como se mencionó anteriormente, **no es interesante** para la resolución de algunos problemas. En nuestro caso**, necesitamos algunos elementos numéricos** para poder trabajar con la información de forma adecuada e impedir la introducción de información inválida para su respectivo uso.



Tarea 4



Tarea 5