

Unidad 4. Schemas XSD de validación de ficheros XML

Como examen de esta unidad debes realizar un fichero XSD que valide perfectamente al fichero XML entregado por el profesor. El fichero XSD debe cumplir con una serie de características que se desglosan en el resto de este texto. En función de las características que logres llevar a cabo recibirás tu puntuación. La entrega del examen se realizará a través de la plataforma Google Classroom. No se admitirá la entrega de exámenes una vez transcurrido el tiempo que se establezca para su realización.

A continuación, encuentras un desglose del sistema de puntuación que se utilizará para evaluar tu prueba.

Piezas de un coche

- ~~1. (0.49 puntos) Contenido Vacío — Los elementos “/piezasCoche/coche” deben ser elementos vacíos.~~
- ~~2. (0.49 puntos) Elemento mixto — Los elementos “/piezasCoche/piezas/motor” deben ser elementos con contenido mixto.~~
- ~~3. (0.98 puntos) Elemento mixto con contenido validado con patrón — Los elementos “/piezasCoche/piezas/motor/modelo” deben validar su contenido texto, siendo necesario que ese contenido contenga al menos 2 caracteres cualesquiera.~~
- ~~4. (1.22 puntos) Elemento con atributos — Los elementos con atributos deben tener declarados todos los atributos que se observan en el fichero.~~
- ~~5. (0.49 puntos) Uso de <xs:sequence> — <xs:sequence>. Los elementos contenidos directamente dentro del elemento “/piezasCoche” deben declararse con el uso de <xs:sequence>. Indica, con un comentario en XSD, encima de <xs:sequence>, las ventajas e inconvenientes del uso de <xs:sequence>~~
- ~~6. (0.49 puntos) Uso de <xs:choice> — Los elementos “/piezasCoche/piezas/materialDeChasis” deben tener en su interior uno de los siguientes elementos vacíos: <aluminio>, <titanio> o <tungsteno>.~~
- ~~7. (0.49 puntos) Uso de <xs:all> — Los elementos que aparecen dentro “/piezasCoche/piezas” deben definirse con <xs:all>. Indica, con un comentario en XSD, encima de <xs:all>, las ventajas e inconvenientes del uso de <xs:all>.~~
- ~~8. (0.98 puntos) Uso de pattern — Los elementos “/piezasCoche/piezas/carrocería” deben tener un contenido textual que valide con el siguiente patrón:~~
 - ~~a. Se debe comenzar con la cadena de texto: “Seat” o “Volkswagen”.~~
 - ~~b. A continuación se debe incorporar un guión (-).~~
 - ~~c. Después debe aparecer un conjunto de letras en mayúscula con un mínimo de 2 caracteres.~~
 - ~~d. A continuación se incorpora un número con una cantidad de dígitos entre 1 y 4.~~
- ~~9. (0.98 puntos) Uso de <xs:union> — Debes crear un tipo de datos simple que, con una enumeración, permita varios valores, que comienzan por un número y continúan por “pulgadas” (ejemplo: 35pulgadas). Debes crear otro tipo de datos similar al anterior, pero cuyos valores terminen en “cm”. Realiza una unión de los dos tipos anteriores y asignasela al contenido de los elementos “/piezasCoche/piezas/parabrisas”.~~
- ~~10. (0.98 puntos) Uso de <xs:list> — Los elementos “/piezasCoche/disponibilidad” deben de poder tener los valores separados que se muestran en el fichero, haciendo uso de <xs:list>.~~
- ~~11. (1.22 puntos) Derivación de tipos de datos simples — Sobre los atributos de los elementos “/piezasCoche/piezas/neumáticos”, haz la siguiente creación de tipos de datos:~~
 - ~~a. “anchura” admitirá valores numéricos con un máximo de tres dígitos.~~
 - ~~b. “altura” tendrá un tipo de datos derivado del anterior, que restringirá sus valores a dos dígitos.~~
 - ~~c. “diámetro” tendrá un tipo de datos derivado del usado en el atributo “anchura”. En este caso debemos incorporar un número cualquiera seguido del texto “pulgadas” (el texto tiene un espacio en blanco al principio).~~
- ~~12. (1.22 puntos) Derivación de tipos de datos complejos — Entre los elementos “/piezasCoche/piezas/asientos” y “/piezasCoche/piezas/tapicería” debes tener tipos de datos que extiendan entre ellos, siendo uno de ellos el tipo primitivo y otro el derivado.~~



¡Suerte!.