Grado en Ingeniería Informática del Software

Software y estándares para la Web

P2. TECNOLOGÍAS XML

Escuela de Ingeniería Informática
Universidad de Oviedo

Tabla de contenidos

Los archivos se deben subir al Campus Virtual en un único archivo empaquetado con el nombre UOXXXXXX (siendo el identificador de cada estudiante en la Universidad de Oviedo).

El archivo empaquetado tendrá una estructura de directorios denominados Ejercicio-1, Ejercicio-2, etc.

En su interior deben estar las carpetas Tarea-1, Tarea-2, etc.

En el interior de estas carpetas deberán estar los archivos solicitados.

Es **OBLIGATORIO** presentar todos los ejercicios y tareas

Objetivos generales

En esta práctica se va a realizar:

- La creación de un documento XML (Teoría "Introducción a XML")
- La validación del documento XML con un DTD (Teoría "DTD")
- La validación del documentos XML con un XML Schema (Teoría "Schemas")
- La transformación del documento XML usando XPATH (Teoría "XPath")
- El procesamiento del documento XML y la generación de nuevos documentos XML (Teoría "Procesamiento y generación de XML")

Ejercicio 1

En este ejercicio se va a realizar la creación de un documento XML (Teoría "Introducción a XML"), la validación del documento XML con DTD (Teoría "DTD") y la validación del documento XML con XML Schema (Teoría "Schemas").

Tarea 1

Construir un documento XML bien formado para contener rutas turísticas (3 rutas mínimo).

Las rutas turísticas serán en función de la terminación del identificador UOXXXXXX.

- UOXXXXXX terminados en 0: rutas de senderismo
- UOXXXXXX terminados en 1: rutas en bicicleta
- UOXXXXXX terminados en 2: rutas en moto
- UOXXXXXX terminados en 3: rutas en coche
- UOXXXXXX terminados en 4: rutas en tren
- UOXXXXXX terminados en 5: rutas en avión
- UOXXXXXX terminados en 6: rutas de drones
- UOXXXXXX terminados en 7: rutas en quad
- UOXXXXXX terminados en 8: rutas en auto-caravana
- UOXXXXXX terminados en 9: rutas a caballo

Las rutas deberán tener los siguientes requisitos mínimos:

- Nombre de la ruta turística (por ejemplo "Ruta por el Naranco")
- Tipo de ruta (por ejemplo "Arquitectura y monumentos", "Gastronómica", "Paisajística", "Mixta tapas y monumentos", etc.)
- Dificultad de la ruta (por ejemplo "Fácil", "Solo para expertos", "Difícil", "Media", etc.)
- Fecha de inicio de la ruta (opcional)
- Hora de inicio de la ruta (opcional)
- Tiempo de duración de la ruta (por ejemplo "2 horas", "3 días", "2 semanas","3 meses")

- Agencia que gestiona la ruta (por ejemplo “Sin agencia”, “NaturAller”)
- Descripción de la ruta
- Personas adecuadas para la ruta (por ejemplo “Se puede ir con niños”, “Personas en buena forma física”, “tercera edad”, etc.)
- Lugar de inicio de la ruta (por ejemplo “Oviedo”)
- Dirección de inicio de la ruta (por ejemplo “calle Foncalada”)
- Coordenadas geográficas de inicio de la ruta: longitud, latitud y altitud
- Referencias y bibliografía con información de la ruta (mínimo 3). Si es posible poner enlaces.
    - Referencia 1: por ejemplo https://es.wikipedia.org/wiki/Foncalada
    - Referencia 2: http://prerromanicoasturiano.es/
    - Referencia 3. etc.
- Recomendación de la ruta de 0 a 10 (por ejemplo “7”)
- Hitos de la ruta (mínimo 3 hitos):
    - Nombre del sitio
    - Descripción del sitio
    - Coordenadas geográficas del sitio: longitud, latitud, altitud
    - Distancia desde el hito anterior (las unidades se expresarán como atributos)
    - Tiempo desde el hito anterior (opcional)
    - Galería de fotografías del hito (mínimo 1, máximo 5). Son enlaces.
        - Fotografía 1: Por ejemplo Monumento.jpg
        - Fotografía 2: Por ejemplo Panorama.jpg
        - Fotografía 3: etc…
    - Galería de vídeos del hito (opcional). Mínimo 0 y máximo 3. Son enlaces.
        - Video 1: Por ejemplo Paisaje360.mpeg o enlace a YouTube, Vimeo, etc.
        - Video 2: Por ejemplo Modelo3D.mpeg
        - Video 3: etc.

Se debe comprobar que los documentos están bien formados en los 4 navegadores de referencia de la asignatura: Chrome, Firefox, Microsoft Edge y Opera.

Tarea 2

Construir un DTD que permita validar el documento XML bien formado.

Los DTD's tienen limitaciones que les impide comprobar rangos y tipos. Estas comprobaciones se realizarán con los XML Schema.

Comprobar la validez con alguna de las herramientas enumeradas en las clases de teoría u otras herramientas.

Tarea 3

Representar en SVG el árbol n-ario del archivo XML utilizando el programa xml2svg.exe que se encuentra en el Campus Virtual.

Se recuerda que xml2svg.exe se utiliza en modo consola y los archivos SVG se visualizan con los navegadores.

Tarea 4

Utilizando Visual Studio 2019 generar automáticamente un XML Schema del documento XML bien formado a partir del DTD. Comprobar que el documento XML bien formado con el XSD generado es válido.

Tarea 5

Modificar el archivo .xsd generado automáticamente para ajustar los tipos de datos a la riqueza de tipos de XML Schema y comprobar los rangos.

También debe comprobarse que la codificación es "UTF-8".

Además, pueden hacerse otros tipos de ajuste aprovechando la mayor potencia del XML Schema sobre los DTD.

Comprobar que el documento XML bien formado con el XSD modificado es válido.

Tarea 6

Representar en un archivo SVG el árbol n-ario del archivo .xsd utilizando el programa *xml2svg.exe* que se encuentra en el Campus Virtual.

Tarea 7

Escribir un informe en un documento de texto indicando las principales diferencias entre el XML Schema construido manualmente y el generado automáticamente.

Ejercicio 2

En este ejercicio el objetivo es procesar y generar documentos XML.

El procesamiento y generación de documentos XML se puede hacer en cualquier lenguaje de programación (Teoría "XPath" y "Procesamiento y generación de XML"). Se deja la libertad al estudiante la elección del lenguaje de programación.

Tarea 1

Se supone que se tiene el documento XML bien formado y válido de rutas turísticas obtenido en el ejercicio 1.

Se debe realizar una **transformación automática del archivo XML a un archivo HTML5** utilizando un lenguaje de programación (se deja libre al estudiante la elección del lenguaje de programación).

El archivo HTML5 debe referenciar a un archivo CSS (construido manualmente).

El archivo HTML5 generado debe mostrar toda la información del documento XML.

El formato de presentación es de libre diseño por parte del estudiante.

El código generado debe comprobar la validez de los documentos generados HTML5 y CSS con los validadores del W3C.

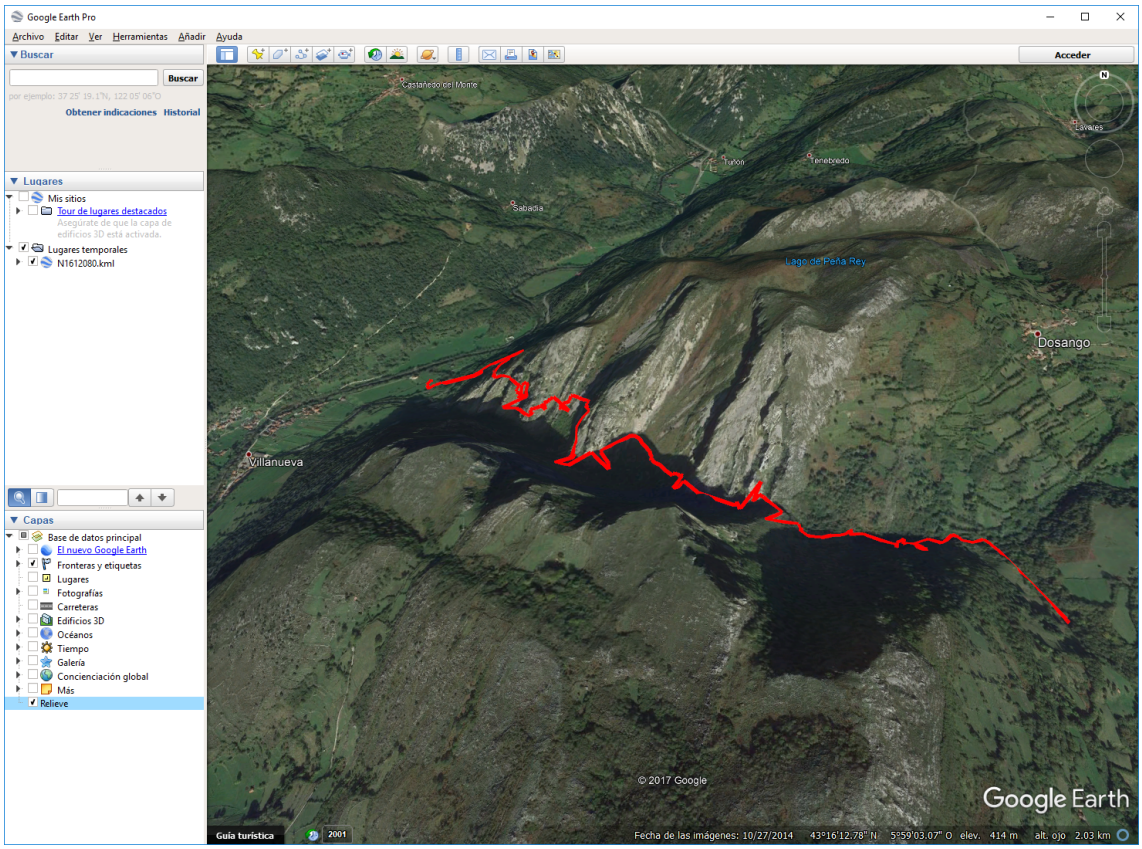
Se deben entregar:

- Archivo XML
- Código fuente del programa que realiza la transformación
- Archivo HTML5 generado
- Archivo CSS

Tarea 2

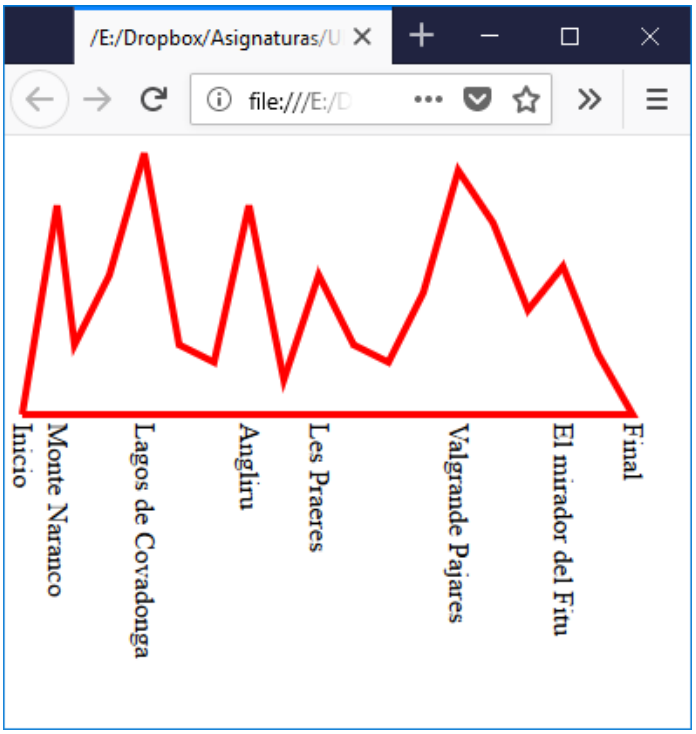
Se debe realizar una transformación del archivo XML a un **archivo KML** que debe mostrar las rutas turísticas cuando se abre en Google Earth.

La transformación puede hacerse con cualquier lenguaje de programación.



Tarea 3

Se debe realizar una transformación del archivo XML a varios **archivos SVG que muestren el perfil altimétrico de las rutas**. El diseño se deja libre a cada estudiante, el aspecto mínimo es el que se muestra en el siguiente gráfico:



Ejercicio 3

Diseñar y construir una aplicación cuya **entrada es un archivo de un lenguaje derivado de XML**

- El lenguaje derivado de XML será elegido por el estudiante según la terminación de su UOXXXXXX
  - UOXXXXXX terminados en 0: archivos .docx
  - UOXXXXXX terminados en 1: archivos .pptx
  - UOXXXXXX terminados en 2: archivos .xslx
  - UOXXXXXX terminados en 3: archivos .ePub
  - UOXXXXXX terminados en 4: archivos .GPX
  - UOXXXXXX terminados en 5: archivos MathML
  - UOXXXXXX terminados en 6: archivos SMILE
  - UOXXXXXX terminados en 7: archivos X3D
  - UOXXXXXX terminados en 8: archivos UPnP
  - UOXXXXXX terminados en 9: archivos TTML
- El diseño de la aplicación se deja libre al estudiante
- El lenguaje de programación a utilizar se deja a libre elección del estudiante
- La entrada al programa debe ser obligatoriamente un archivo de un lenguaje derivado de XML
- Se valorará la originalidad y la utilidad de la aplicación desarrollada

Se debe presentar:

- Código fuente y ejecutable (si lo hubiera)
- Archivo **léeme.txt** indicando el lenguaje de programación utilizado, la versión del compilador o intérprete usada y los archivos de prueba utilizados. También se incluirán las instrucciones de uso.
- Archivos XML de prueba utilizados
- Archivos generados (si los hubiera)