



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Aplicación WaterML

Dennis Gabriela Coy Calderón

Universidad de Oviedo
Escuela de Ingeniería Informática
Software y Estándares para la Web

Descripción

El objetivo de este documento es explicar la aplicación que se realizó en Python con un archivo de entrada derivado de XML, el cual, según la finalización de mi UO, que es 6, es un archivo WaterML.

Estructura del archivo WaterML

WaterML 2.0 del OGC, es un estándar abierto del OGC, es decir está disponible libremente para que cualquiera lo utilice con la finalidad de mejorar el intercambio de datos en el mundo. WaterML 2.0 es una versión actualizada de WaterML, se basa en el estándar O&M de OGC. La principal diferencia entre WaterML y WaterML 2.0 es en la forma de estructuración y la sintaxis. Además, WaterML 1.0 (1.1) utiliza como servicio web WaterOneFlow y WaterML 2.0 puede utilizar como servicio web el estándar SOS de OGC, así como otros servicios que proveen datos como Web Feature Service (WFS).

WaterML 2.0 se estructura en partes; la primera parte del estándar WaterML2.0: Timeseries se centra en un modelo de información estándar y una codificación XML derivada de este modelo para series temporales de observaciones hidrológicas. La segunda parte de WaterML 2.0: Ratings, Gaugings and Sections describe un modelo de información para intercambiar algunos indicadores y medidas observacionales muy específicas como por ejemplo secciones de ríos. Otras partes del estándar WaterML 2.0 se extenderán a otras áreas de datos hidrológicos, tales como la descripción de calificación de curvas, información de calibración, diccionarios (controled vocabularies) etc.

Esta aplicación tiene como objetivo visualizar las series de tiempo que aparecen en la primera parte de la estructura del WaterML

La estructura de los elementos a graficar es la siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<wml2:Collection xmlns:wml2="http://www.opengis.net/waterml/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:om="http://www.opengis.net/om/2.0"
xmlns:sa="http://www.opengis.net/sampling/2.0"
xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
```

```

xsi:schemaLocation="http://www.opengis.net/waterml/2.0
http://www.opengis.net/waterml/2.0/waterml2.xsd" gml:id="Ki.Col.1">
...
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>1990-09-01T00:00:00.000+01:00</wml2:time>
<wml2:value>193.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
...
</wml2:Collection>

```

Teniendo en cuenta la estructura anterior, se extraen los datos de tipo MeasurementTVP, para tomar sus elementos time y value y graficarlos.

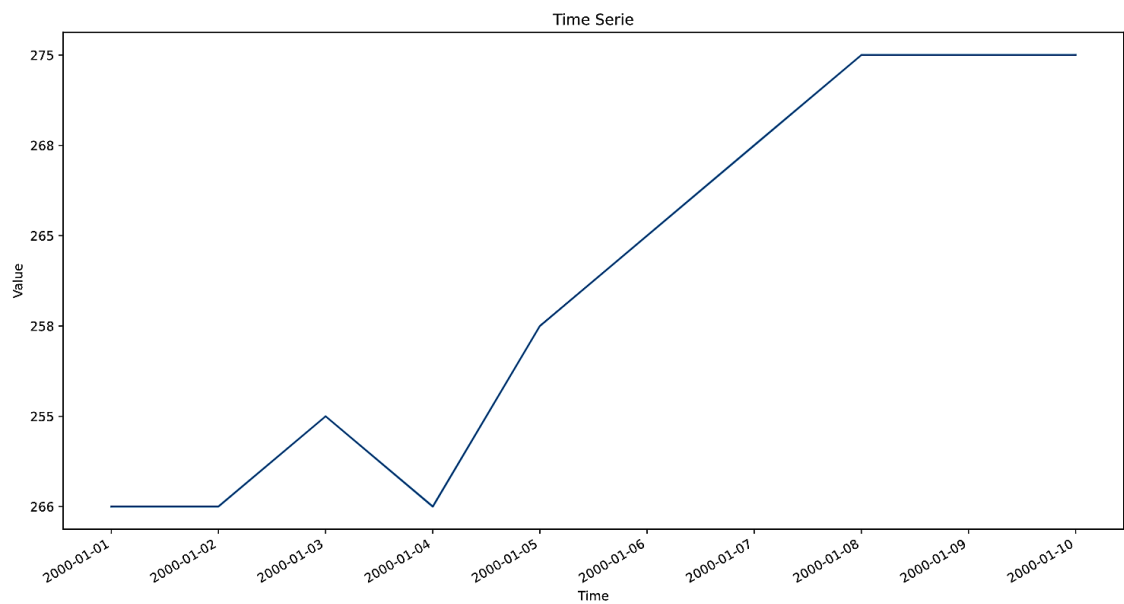
Construcción del código en Python

Para esta aplicación se hizo uso de Python 3.8 mediante el entorno de desarrollo Spyder. El código wmlTimeSerie.py utiliza la librería xml.etree.ElementTree para la conversión del archivo XML en su árbol y así poder recorrer sus elementos como nodos, la librería matplotlib.pyplot para la gráfica y dateutil.parser para convertir la cadena de caracteres en la que viene dado el elemento de time a un tipo de dato date.

Se compone de la función timeSerieWML(file) la cual a partir del archivo que tiene como parámetro crea un árbol con la librería mencionada anteriormente, busca los nodos que tienen por nombre MeasurementTVP con la función iter(), encuentra los valores time y value con la función find(), hace el cambio de tipo de dato del elemento time, crea un gráfico a partir de los datos de time (eje x) y value (eje y) y lo guarda como un archivo svg.

Los archivos utilizados como ejemplo para el uso de esta aplicación son “measurement-timeseries-discharge.wml” y “KiWIS-WML2-Example.wml” obtenidos a partir de [OGC® WaterML | OGC](http://www.opengis.net/OGC/6/100).

SVG generado del archivo “measurement-timeseries-discharge.wml”



SVG generado del archivo “KiWIS-WML2-Example.wml”

