



Instituto Tecnológico
de Buenos Aires

Trabajo Práctico Especial

Primer cuatrimestre de 2021

Grupo 7

Integrantes:

Chayer, Iván	61360
Di Toro, Camila	62576
Baron, Maria Mercedes	61187

Introducción

A lo largo de este curso, hemos visto diferentes maneras de diseñar y procesar documentos XML. El objetivo de este trabajo práctico es poner en práctica los conocimientos adquiridos.

El proyecto que hemos diseñado, permite procesar información de la competición de automóviles que es conocida mundialmente como NASCAR(National Association for Stock Car Auto Racing), a partir de un año y un tipo de competición los cuales deben ser válidos.

A continuación, detallamos el trabajo realizado junto con los aspectos que debieron investigarse, los archivos generados, las dificultades encontradas y sus correspondientes errores.

Desarrollo

1. Contenido del ZIP Grupo07.zip

En el zip se encontrarán los siguientes archivos:

- `tpe.sh`
- `full_validations.sh`
- `extract_nascar_data.xq`
- `generate_markdown.xsl`
- `errors.xml`
- `nascar_data.xsd`

* El archivos *tpe.sh* trabaja a la par con el archivo *full_validations.sh* para ejecutar las líneas de código necesarias y generar la página Markdown *nascar_page.md* de salida. Para obtener esta salida se chequea que los parámetros sean correctos. De no serlos, el Markdown de salida mostrará el error que se produjo en la ejecución del programa. Caso contrario, se llama a la API como forma de obtener los XML *drivers_list.xml* y *drivers_standing.xml* que permitirá realizar la consulta *extract_nascar_data.xq* y la transformación *generate_markdown.xsl* para obtener el resultado final.

* El archivo *full_validations.sh* se encarga de hacer las validaciones necesarias para comprobar que los parámetros recibidos por entrada estándar sean los que correspondan y estén en el orden correcto.

* El archivo *extract_nascar_data.xq* produce como salida el documento XML *nascar_data.xml* que valida contra el XML schema del enunciado del TPE llamado *nascar_data.xsd*.

* El archivo *generate_markdown.xsl* toma los datos de *nascar_data.xml* y produce la página Markdown llamada *nascar_page.md*.

* El archivo *errors.xml* contiene los diferentes errores que pueden producirse a la hora de validar los parámetros ingresados por entrada estándar. Éste será llamado por *extract_nascar_data.xq* si es que alguna de las validaciones hechas en *full_validations.sh* produjera algún error.

2. Ejecución¹

Para que el programa se ejecute correctamente, se debe hacer desde la terminal y, en el directorio donde se encuentren los archivos contenidos en el zip mencionado en el ítem 1, se debe ejecutar la siguiente línea de código: `./tpe.sh year type`. Donde *year* es un año en el intervalo natural [2013, 2021] y *type* es uno de los siguientes tipos de competición: *sc*, *xf*, *cw*, *go*, *mc* y *enas*. Para la categoría *enas*, el único año válido es el 2020. Si no se envían parámetros, o se envían parámetros incorrectos, el programa producirá un Markdown de salida explicando el error producido (ver ítem 3 para mayor comprensión de los posibles errores). Caso contrario, se retorna un Markdown con la información de la competición y el año en cuestión.

Se debe setear la variable de entorno SPORTRADAR_API.

3. Tipos de errores

Estos son los errores que pueden surgir al correr el programa:

- **Error, two parameters are required: year and type.** Si la cantidad de parámetros recibidos por entrada estándar es distinta de dos.
- **Error, year should be a number between 2013 and 2021.** Si el tipo de competición es válido y el año no es un número o no está en el intervalo natural [2013 - 2021].
- **Error, invalid type. Type should be one of this options: sc, xf, cw, go, mc, enas.** Si el tipo de competición es inválido y el año es válido.
- **Error, invalid year for type enas. The only valid year for type enas is 2020.** Si el tipo de competición es *enas* pero el año es distinto de 2020.
- **Error, invalid type and year.**
Year should be a number between 2013 and 2021.
Type should be one of this options: sc, xf, cw, go, mc, enas.
Parameters could be in the wrong order. The first parameter must be year and type must be the second one, not otherwise. Si el tipo de competición es inválido y el año no es número o el año no corresponde al intervalo natural [2013 - 2021] o si los parámetros ingresados están en orden incorrecto.

4. Investigaciones

- Revisión del manejo de las funciones en Bash como base para avanzar en la programación de los archivos *tpe.sh* y *full_validations.sh*.
- Lenguaje Markdown.
- Investigación de como pasar por parámetro desde Bash, dos parámetros externos a un XML.
- Incorporación de una mayor cantidad de contenidos vistos en clase para conocer mejor sus correctas aplicaciones. Por ejemplo, funciones de xQuery, entre otras.
- Breve investigación del funcionamiento de la API de SportRadar.

¹ **Nota:** El código fue comentado de la mejor forma posible para que se pueda hacer un seguimiento ordenado de nuestro razonamiento para resolver la consigna. Los comentarios hechos en el código no llevan tildes dado que podrían entrar en conflicto con el UTF-8.

5. Dificultades encontradas

Debemos admitir que al principio el grupo dudó cómo encarar adecuadamente el proyecto. Por ejemplo, ver los lenguajes intermedios que podíamos utilizar para resolver algunos problemas (Java o Bash).

Notamos que los XML obtenidos de la API eran inconsistentes. Varios de los corredores, por ejemplo, tienen en el campo birthplace “ESTADOS UNIDOS” o “Estados Unidos”. Decidimos conservar el estado original del XML y no modificarlo.

No supimos cómo prototipar las funciones en el archivo xQuery para que quedara más prolijo el archivo.

También nos costó encontrar la forma de que la función *get_car* definida en el xQuery no tuviera un valor de retorno en caso de que el driver no tuviera un tag <car>. Además en esa misma función, se nos dificultó bastante cómo hacer para quedarnos con el primer elemento, si es que una expresión XPATH nos devuelve más de un elemento.

6. Roles

A continuación detallamos los roles asignados en este trabajo, cabe aclarar que todo el grupo colaboró en las distintas partes del proyecto:

Responsable del funcionamiento global del proyecto: Camila Di Toro.

Responsable de la presentación: Iván Chayer, Maria Mercedes Baron.

Responsable del funcionamiento de la consulta xQuery: Camila Di Toro.

Responsable del funcionamiento de la plantilla XSLT: Camila Di Toro.

Responsables del funcionamiento de los archivos .sh: Iván Chayer y Maria Mercedes Baron.

Conclusiones

Con este proyecto, hemos arribado a la conclusión de que, un archivo diseñado en XML no sólo es entendible debido a que sus partes pueden diferenciarse fácilmente, sino que además, los documentos creados con este formato, son fáciles de procesar. Estos beneficios resultan ser de gran utilidad al momento de modificar o bien de extraer datos de un documento, ya que gracias a su estructura, se localiza con facilidad lo que se está buscando.

Además, corresponde agregar lo ventajoso que es el recurso del lenguaje Markdown, dado que permite dar formato a un texto sin necesidad de recordar las complicadas etiquetas de HTML.

Por último, como grupo hemos notado que algunos de los conocimientos adquiridos a lo largo de esta carrera (Bash) y especialmente de esta materia, se requirieron ampliamente a lo largo de este proyecto. Se pudo ver la manera en la que éstos se emplean y funcionan simultáneamente. Este proceso nos entusiasmó debido a que tuvimos la oportunidad de comenzar a relacionar y experimentar con los conocimientos que fuimos cultivando.