Instituto Tecnológico de Buenos Aires

72.32 - Diseño y Procesamiento de Documentos XML

Trabajo Práctico Especial

Grupo N°18 - Año 2025

Consulta y Transformación de documentos XML

Integrantes:

- Ron Zeiger - 65027
- Lucas Baigorria Puiggros - 64979
- Thomas Tilve - 65862
- Tomas Arce - 64967

Profesores:

- Valeria Inés Soliani
- Teresa Natalia Fontanella De Santis

Introducción

El presente trabajo práctico tiene como objetivo aplicar los conocimientos adquiridos sobre consulta y transformación de documentos XML, utilizando las herramientas y lenguajes vistos durante la cursada: XQuery, XSLT y Bash scripting.

El proyecto consiste en desarrollar un proceso automatizado que permita obtener información desde la API de CEPALSTAT, un servicio público de la Comisión Económica para América Latina y el Caribe (CEPAL) que expone indicadores socioeconómicos, demográficos y ambientales de los países de la región.

La información disponible en la API de CEPALSTAT, si bien es pública y de acceso libre, se encuentra estructurada en formatos técnicos como JSON y XML, los cuales están pensados para el intercambio de datos entre sistemas informáticos y no para su lectura directa por parte de las personas. Esto hace que los datos no puedan visualizarse de forma clara o comprensible desde la página o los archivos originales. Frente a esta limitación, el trabajo desarrollado tuvo como objetivo construir un programa que permita consultar y visualizar dichos datos de manera legible, combinando consultas XQuery, transformaciones XSLT y un script de automatización en Bash. De esta forma, el usuario puede ingresar un prefijo con el nombre de un indicador y obtener como resultado una página Markdown clara y organizada, con toda la información relevante del indicador proveniente de la API de CEPALSTAT.

Desarrollo

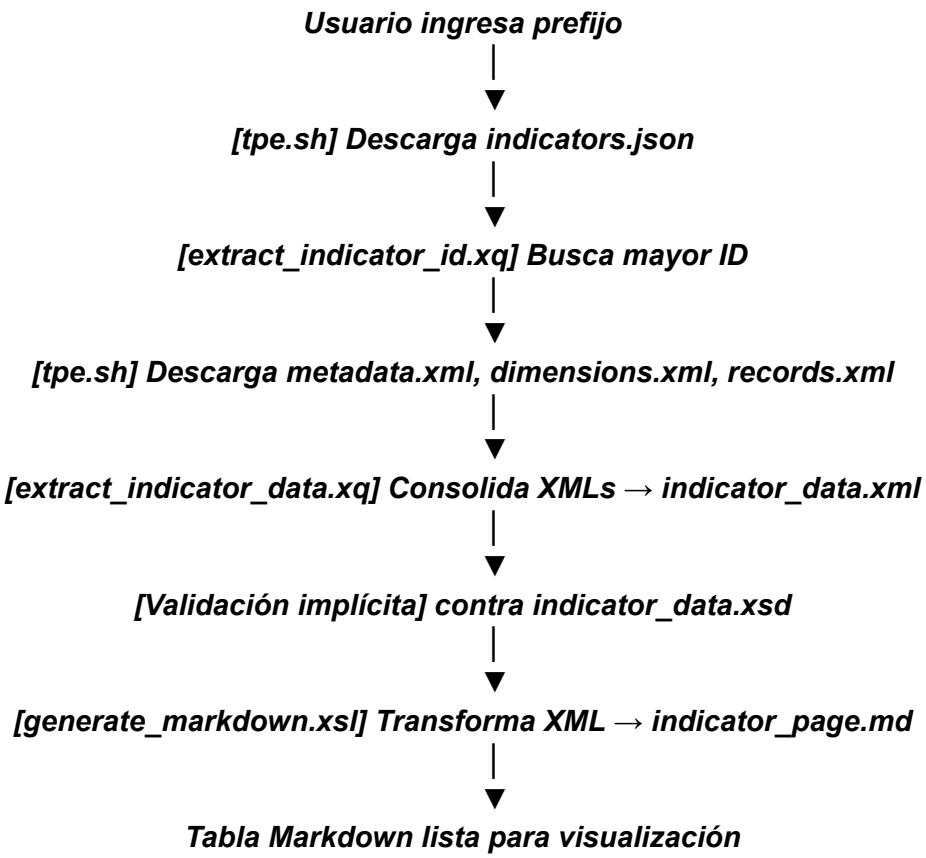
Trabajo realizado

El trabajo consistió en desarrollar un pipeline automatizado para consultar la API REST de CEPALSTAT, extraer datos de un indicador socioeconómico y transformarlos en un documento Markdown legible, utilizando XQuery, XSLT y Bash. Se implementaron cinco archivos principales que cumplen con los requisitos del enunciado:

Archivo	Descripción
extract_indicator_id.xq	XQuery que procesa *indicators.json* (obtenido mediante GET /thematic-tree) y busca el *indicator_id* a partir de un prefijo ingresado por el usuario (ej: “Total population”). Utiliza starts-with() y max() para seleccionar el id más alto entre los coincidentes.

extract_indicator_data.xq	XQuery que combina metadata.xml, dimensions.xml y records.xml. Mapea los IDs de dimensiones a sus nombres y valores reales. Genera indicator_data.xml con la estructura: <details> (metadatos) y <data><record> (filas de datos).
generate_markdown.xsl	XSLT que transforma indicator_data.xml en markdown. Genera: título, metadatos en párrafos, y una tabla dinámica con columnas basadas en distinct-values()
indicator_data.xsd	Esquema XML que define la estructura valida de indicator_data.xml. Incluye <details>, <data> <record><dimension>, y <error> para manejo de fallos.
tpe.sh	Script Bash que automatiza todo el proceso: descarga con *curl*, ejecuta XQuery/XSLT con Saxon HE 12.9, y genera los archivos de salida.

El esquema de todo el proceso se podría visualizar de la siguiente manera:



Dificultades encontradas

En primer lugar, se nos presentó un problema al intentar usar *Xerces* en nuestro proyecto, motivo por el cual decidimos realizar este proyecto con el parser *Saxon 12.9*. Una vez instalado el parser, el primer problema que tuvimos fue con el archivo *extract_id* que se supone que nos debía devolver un número (el ID del campo en formato texto). En su lugar nos devolvía el ID correcto pero con su etiqueta, lo cual nos rompía el flujo de nuestro proyecto ya que esperábamos el número solo y así poder utilizarlo en otra parte del código. Al imprimir el ID con su etiqueta, cuando lo utilizabamos para los distintos curl dentro de los url estos arrojaban un error, ya que solo esperaban un número en ese campo.

El siguiente problema que tuvimos fue con el Saxon. Una vez probado y ya con el código correcto para el ID, queríamos poder probar los archivos individualmente para tener el control de cada parte del proyecto. En esta parte, tuvimos algunos problemas a la hora de tratar de ejecutar por comandos cada archivo. Investigando, logramos armar los comandos, que luego implementaríamos en el *tpe.sh*.

Durante las pruebas realizadas en medio del proyecto, uno de los errores que tuvimos fue que, aunque estábamos recibiendo el archivo de salida *indicator_data.xml*, el cuerpo del mismo se encontraba vacío. Eso significaba que a la hora de procesar los datos, algo no se estaba realizando correctamente. Luego de un par de ajustes en la lógica del procesamiento, pudimos solucionarlo, y ahora sí, generaba el cuerpo correctamente dentro del archivo.

Una vez que el tpe.sh lograba devolvernos los archivos de salida, notamos un pequeño error que nos impedía obtener el producto final. El problema era que faltaba *metadata.xml* dentro de todos los archivos esperados. Por lo que tuvimos que revisar y modificar nuestro código para poder recibirlo correctamente.

En un principio a la hora de usar los comandos curl para descargar metadata.xml, dimensions.xml y records.xml desde la api de cepalstat, utilizabamos url mal formados. El error estaba en la forma de pasar el id. Era simplemente un error sintáctico, pero esto hacía que los url se carguen mal y tire un error.

Roles de los integrantes

Aunque para la organización del trabajo se asignaron roles específicos, es importante destacar que todos los integrantes participaron activamente en cada etapa del proyecto y colaboraron para resolver los problemas que surgían.

Responsables de la consulta XQuery (L. Baigorria y T. Tilve): Garantizaron que las consultas devolvieran los datos correctos en el formato esperado para la plantilla XSLT. Sin embargo, todos revisamos y probamos las consultas juntos, corrigiendo errores y ajustando la lógica según era necesario.

Responsable de la plantilla XSLT (Tomas Arce): Se encargó de que la transformación a Markdown funcionara y que se imprimiera correctamente. Nuevamente, cada integrante participó en la prueba y validación del resultado, aportando sugerencias y detectando casos problemáticos.

Responsable del funcionamiento global (Ron Zeiguer): Supervisó la integración de todas las partes, asegurando que el flujo completo funcionara de punta a punta y de armar el archivo *tpe.sh* que unía todas las partes. El resto del equipo colaboró en la verificación de compatibilidad y en la resolución de errores que aparecían en distintos archivos.

Responsable de la presentación (L. Baigorria): Coordinó la redacción y revisión del informe final, cuidando la presentación y consistencia del documento. No obstante, la revisión del contenido y la corrección de detalles fue un esfuerzo conjunto del equipo.

Conclusiones

El presente Trabajo Práctico nos permitió aprender a extraer datos desde una API externa y transformarlos a distintos formatos, recorriendo todo el flujo desde los datos crudos hasta la presentación final. Durante la experiencia, trabajamos con *JSON* y *XML*, aplicando consultas *XQuery* para navegar por estructuras complejas, filtrar información por condiciones específicas y convertir los datos a un formato adecuado para su uso posterior. En particular, la lógica implementada para obtener el ID del indicador mostró cómo es posible seleccionar la información relevante y preparar los datos para los pasos siguientes del flujo.

Por otro lado, la transformación *XSLT*, con *generate_markdown.xsl,* nos permitió convertir los datos procesados en un documento Markdown legible y bien estructurado, con tablas y secciones que facilitan la interpretación de los resultados. Esto nos enseñó cómo XSLT puede adaptarse para presentar información de distintas maneras, según las necesidades de visualización.

La integración de todo el proceso mediante un script *Bash* nos permitió automatizar la descarga de datos, la ejecución de las consultas *XQuery* y la transformación *XSLT*, logrando que el flujo completo se ejecute de manera sistemática y repetible. Esta automatización es clave cuando se trabaja con grandes volúmenes de datos, ya que asegura que los resultados sean consistentes y confiables.

En conclusión, este Trabajo Práctico nos permitió poner en práctica los conceptos de extracción, transformación y presentación de datos, consolidando habilidades que van más allá de la teoría y que resultan esenciales para el desarrollo de soluciones informáticas modernas. Nos brindó una visión completa de cómo manejar datos desde su obtención hasta su presentación final, y reforzó la importancia de pensar en procesos automatizados y estructurados para trabajar con información de manera eficiente.