

Diseño y Procesamiento de Documentos XML

Segundo Parcial en Modalidad: Trabajo Práctico Especial

2do cuatrimestre 2021

1. Objetivo

El objetivo de este Trabajo Práctico es aplicar las herramientas de consulta y transformación de documentos XML vistas a lo largo del curso.

2. Modalidad

El Trabajo Práctico debe realizarse en grupos formados por 4 integrantes cada uno. Los entregables del trabajo, detallados en el ítem 5, deben ser cargados en la plataforma Campus ITBA, como respuesta al envío de esta actividad.

3. Contexto

En este Trabajo usaremos el sitio web AirLabs (https://airlabs.co/docs/#docs_Introduction), que permite consumir datos y estadísticas en tiempo real respecto de distintos vuelos (con información geográfica, de aeropuertos, etc), en forma de invocaciones a APIs REST (tanto en formato XML como JSON).

Se utilizarán como base los siguientes documentos XML:

- **airports.xml**: Se obtiene a partir de llamar al método GET [airports](#) de la API de AirLabs. Este método devuelve una lista de aeropuertos con su información correspondiente.
- **countries.xml**: Se obtiene a partir de llamar al método GET [countries](#) de la API de AirLabs. Este método devuelve una lista de países con su información.
- **flights.xml**: Se obtiene a partir de llamar al método GET [flights](#) de la API de AirLabs. Este método devuelve una lista en tiempo real de los vuelos en curso/planificados/completados.

Para que las llamadas a la API sean correctas, se debe incluir un parámetro llamado *api_key* con una Free API KEY de AirLabs. Los pasos para generarla se encuentran en el "Anexo 1". Por ejemplo, para obtener el archivo airports.xml, teniendo *api_key* en una variable Bash llamada "AIRLABS_API_KEY", se puede utilizar la aplicación curl de la siguiente forma:

```
curl https://airlabs.co/api/v9/airports.xml?api_key=${AIRLABS_API_KEY} > airports.xml
```

Y así, generar el siguiente XML de salida:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<request>
<lang>en</lang>
<currency>USD</currency>
<time>279</time><id>n9qx7n1gea8</id><server>b</server>
```

```

...
</request>
<response>
<response>
<icao_code>KMIA</icao_code>
<country_code>US</country_code>
<iata_code>MIA</iata_code>
<lng>-80.29</lng>
<name>Miami International Airport</name>
<lat>25.79</lat>
</response>
<response>
<icao_code>KBWI</icao_code>
<country_code>US</country_code>
<iata_code>BWI</iata_code>
<lng>-76.67</lng>
<name>Baltimore-Washington International Airport</name>
<lat>39.18</lat>
</response>
...
...
<response>
<icao_code>YZAN</icao_code>
<country_code>AU</country_code>
<lng>123.57</lng>
<name>Zanthus Airport</name>
<lat>-31.03</lat>
</response>
<response>
<icao_code>YZHN</icao_code>
<country_code>AU</country_code>
<lng>145.35</lng>
<name>Zeehan Airport</name>
<lat>-41.88</lat>
</response>

</response>
<terms>Unauthorized access is prohibited and punishable by law.
Reselling data 'As Is' without AirLabs.Co permission is strictly prohibited.
Full terms on https://airlabs.co/.
Contact us info@airlabs.co</terms></root>

```

Asimismo, para obtener el archivo **countries.xml** utilizando el mismo parámetro, se puede utilizar curl de la siguiente forma:

```
curl https://airlabs.co/api/v9/countries.xml?api_key=${AIRLABS_API_KEY} > countries.xml
```

Y así, generar el siguiente XML de salida:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>

```

```

<request>
<lang>en</lang><currency>USD</currency>
<time>3</time><id>antu4086omg</id>
<server>g</server><host>airlabs.co</host><pid>25711</pid>
...
</request>
<response>
<response>
<code>AD</code>
<code3>AND</code3>
<name>Andorra</name>
</response>
<response>
<code>AE</code>
<code3>ARE</code3>
<name>United Arab Emirates</name>
</response>
<response>
<code>AF</code>
<code3>AFG</code3>
<name>Afghanistan</name>
</response>
...
...
<response>
<code>ZW</code>
<code3>ZWE</code3>
<name>Zimbabwe</name>
</response>
</response>
<terms>Unauthorized access is prohibited and punishable by law.
Reselling data 'As Is' without AirLabs.Co permission is strictly prohibited.
Full terms on https://airlabs.co/.
Contact us info@airlabs.co</terms></root>

```

Por otro lado, para obtener el archivo **flights.xml** utilizando el mismo parámetro, se puede utilizar curl de la siguiente forma:

```
curl https://airlabs.co/api/v9/flights.xml?api_key=${AIRLABS_API_KEY} > flights.xml
```

Y así, generar el siguiente XML de salida:

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
<request>
<lang>en</lang><currency>USD</currency><time>7</time><id>cjdi4i08h6g</id><server>g</server>
<host>airlabs.co</host>
...
</request>

<response>

```

```

<response>
<hex>842350</hex>
<flag>JP</flag>
<lat>36.38887</lat>
<lng>134.859066</lng>
<dir>236.7</dir>
<dep_iata>CTS</dep_iata>
<arr_iata>FUK</arr_iata>
<aircraft_icao>A320</aircraft_icao>
<flight_iata>MM456</flight_iata>
<flight_number>456</flight_number>
<airline_iata>MM</airline_iata><alt>11582</alt><speed>687</speed><v_speed>0</v_speed>
<squawk>2375</squawk><status>en-route</status>
</response>
...
...

<response>
<hex>FEDF6E</hex>
<flag>HU</flag>
<lat>47.308334</lat>
<lng>18.9995</lng>
<dir>32</dir>
<aircraft_icao>GLID</aircraft_icao>
<alt>397</alt>
<speed>135</speed>
<status>en-route</status>
</response>
</response>
<terms>Unauthorized access is prohibited and punishable by law.
Reselling data 'As Is' without AirLabs.Co permission is strictly prohibited.
Full terms on https://airlabs.co/.
Contact us info@airlabs.co</terms>
</root>

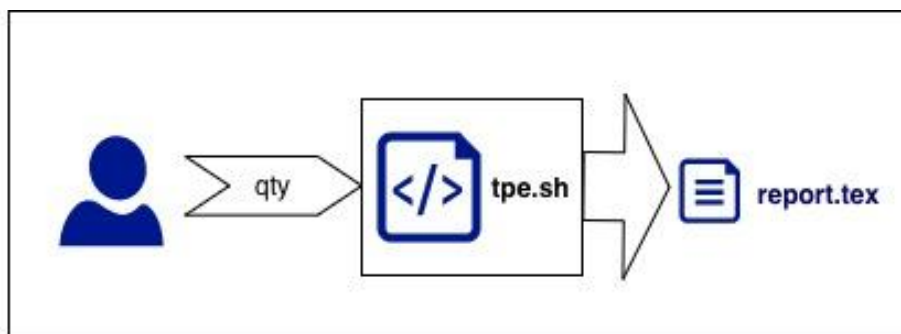
```

4. Procedimiento

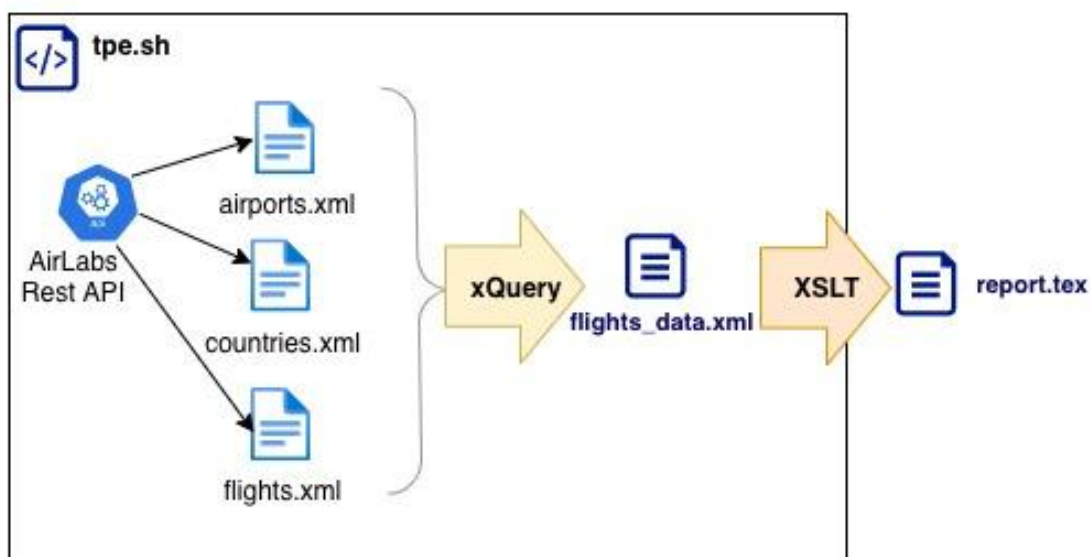
La finalidad del trabajo es obtener, a partir de la información mencionada en el punto 3, un documento LáTex con la información de los vuelos, ordenados alfabéticamente por su identificador. La información a mostrar, para cada uno de ellos, es la siguiente:

- Id
- País
- Posición geográfica
- Estado del vuelo
- Aeropuerto de salida
- Aeropuerto de llegada

Además, se puede ingresar el parámetro opcional **qty** que, en caso de aparecer, indica el límite máximo de vuelos que se pueden mostrar en el reporte.



Previo a la generación del reporte LáTeX, se debe crear un documento XML (**flights_data.xml**) que contenga los datos que serán mostrados en la página. Gráficamente así sería el flujo del proceso.



A. Tal como se mencionó en el ítem 3, se deben generar los siguientes archivos como fuentes de datos:

- **airports.xml**: Generado a partir de la API de AirLabs con el método “get airports” (explicado en el punto 3).
- **countries.xml**: Generado a partir de la API de AirLabs con el método “get countries” (explicado en el punto 3).
- **flights.xml**: Generado a partir de la API de AirLabs con el método “get flights” (explicado en el punto 3).

Es función de los alumnos analizar la estructura de los documentos fuentes en base a los archivos provistos.

Además, por consola se puede ingresar el parámetro opcional **qty**, que indica el límite máximo de vuelos que se pueden mostrar en el reporte. En caso de no aparecer dicho parámetro, se deben mostrar todos los vuelos (sin límite máximo).

- B. Utilizando los archivos descritos en a), se pide crear la consulta XQuery **extract_data.xq**, que debe producir como salida el documento XML que contenga el listado de vuelos, con su correspondiente información de país, aeropuertos de origen y destino, estado actual del viaje y posición geográfica. El XML **flights_data.xml** debe tener la estructura indicada en el siguiente XML Schema (llamado **flights_data.xsd**).

flights_data.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="flights_data" type="resultT"/>
<xsd:complexType name="resultT">
  <xsd:choice>
    <xsd:element name="flight" type="flightT" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="error" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="flightT">
  <xsd:sequence>
    <xsd:element name="country" minOccurs="0" type="xsd:string"/>
    <xsd:element name="position" type="posT"/>
    <xsd:element name="status" type="xsd:string"/>
    <xsd:element name="departure_airport" minOccurs="0" type="airportT"/>
    <xsd:element name="arrival_airport" minOccurs="0" type="airportT"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="airportT">
  <xsd:sequence>
    <xsd:element name="country" type="xsd:string"/>
    <xsd:element name="name" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="posT">
  <xsd:sequence>
    <xsd:element name="lat" type="xsd:float"/>
    <xsd:element name="lng" type="xsd:float"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Los campos que deben extraerse del archivo **flights.xml** son los siguientes:

- *id* (que es el atributo *hex* del elemento *response*)
- *position* (compuesto por los elementos *lat* y *lng* del elemento *response*)
- *status*
- *country* (integrándolo por el code en *countries.xml*)

La información relacionada con los aeropuertos debe extraerse de los XML **airports.xml** y **countries.xml** (utilizando los elementos *dep_iata/arr_iata* de los vuelos con el elemento *iata_code* de los aeropuertos).

Los elementos *departure_airport* y *arrival_airport*, no deben aparecer si no se encuentra su respectiva información (ya sea su nombre como el país en donde se ubica dicho aeropuerto). Si llega a haber más de un aeropuerto posible para un mismo vuelo (ya sea de salida o de llegada), se debe incluir la primera referencia encontrada.

C. Crear la transformación XSLT **generate_report.xsl** que tomando los datos de **flights_data.xml** produzca un documento LáTex llamado **report.tex** con las siguientes características.

- Debe ser un documento en hoja A4, con letra de tamaño 10 puntos.
- En la primera página debe aparecer un título con el siguiente contenido: "Flights Report". Luego en la línea siguiente debe aparecer el texto: "XML Group XX" (donde XX debe ser reemplazado por el número de Grupo en dos dígitos). Posteriormente, debe aparecer la fecha actual..
- A partir de la segunda página, se mostrará una tabla con un encabezado fijo compuesto por 6 columnas: "Flight Id", "Country", "Position", "Status", "Departure Airport", "Arrival Airport". Luego se insertan dos líneas horizontales divisorias. Y por cada fila, se debe mostrar la siguiente información de los vuelos:
 - Flight Id
 - Country
 - Position: latitud y longitud encerrados con paréntesis y separados por comas.
 - Status
 - Departure Airport
 - Arrival Airport
- Al final de la tabla, insertar otra línea horizontal de separación.

Tener en cuenta que la lista de los vuelos debe estar **ordenada alfabéticamente** por el id del vuelo. Asimismo, la plantilla XSLT debe poder recibir un parámetro opcional llamado **qty** que, de estar presente, limita el número máximo de vuelos al valor de dicho parámetro. Si su valor es nulo, se deben mostrar todos los vuelos (sin límite máximo).

El documento pdf obtenido del documento LáTex debería tener un contenido similar al siguiente ejemplo:

Flight Report

XML Group XX

October 27, 2021

Flight Id	Country	Position	Status	Departure Airport	Arrival Airport
001102		(35.4667 , 32.7519)	en-route		
00403A	Zimbabwe	(-24.52 , 28.58)	en-route	O.R. Tambo Inter- national Airport	Robert Gabriel Mu- gabe International Airport
0082D2	Zimbabwe	(-26.07 , 28.26)	en-route		
008756	South Africa	(-25.942291 , 27.555576)	en-route	Lanseria Interna- tional Airport	
008DE1	South Africa	(-33.261337 , 19.947437)	en-route	Cape Town Interna- tional Airport	O.R. Tambo Inter- national Airport
008DE2	South Africa	(-25.92691 , 27.783025)	en-route	Cape Town Interna- tional Airport	Lanseria Interna- tional Airport
009993	South Africa	(-32.05 , 21.04)	en-route	O.R. Tambo Inter- national Airport	Cape Town Interna- tional Airport
00999B	South Africa	(-25.96 , 28.39)	en-route	East London Air- port	O.R. Tambo Inter- national Airport
00A517	United Kingdom	(-26.132355 , 28.234695)	landed	O.R. Tambo Inter- national Airport	King Shaka Inter- national Airport
00B026	South Africa	(-28.333879 , 25.491137)	en-route	O.R. Tambo Inter- national Airport	Cape Town Interna- tional Airport

Queda a entera decisión del equipo el diseño del documento LáTex (siguiendo las características principales solicitadas en este punto).

- D. Para que todo el proceso se ejecute automáticamente se debe crear el script bash **tpe.sh** que genere el documento LáTex **report.tex** de salida, a partir de las transformaciones aplicadas en los puntos **4.a)**, **4.b)** y **4.c)**. Y puede recibir como parámetros de entrada *qty*, que es la cantidad máxima de vuelos mostrados en el reporte LáTex (debe ser un número entero positivo), y que, en caso de aparecer, debe ser utilizado en la llamada de la plantilla XSLT.

Para realizar la llamada a la API de AirLabs, la API Key debe estar definida en la **variable de entorno** `AIRLABS_API_KEY`.

Por ejemplo: ejecutando **tpe.sh** con el parámetro: `qty=10`, debería generarse como salida final el archivo **report.tex** mostrado en el punto **4.c)**.

Queda a criterio de los alumnos el diseño del script `tpe.sh`, pero cualquier texto que se imprima por pantalla debe estar en Inglés.

- E. **Errores:** Se deben manejar los errores posibles e informar las decisiones tomadas. Los errores deben aparecer en el archivo **flights_data.xml** en uno o varios `<error>` dentro del tag `<flights_data>`. Por ejemplo, en caso de que se ingrese un parámetro incorrecto, la salida de la consulta XQuery debe producir algo similar a lo siguiente:

```
<flights_data>
  <error>Unknown error when retrieving data</error>
</flights_data>
```

Luego, el reporte LáTex resultante, deberá contener sólo dicha leyenda de error.

- F. Dado que éste es un trabajo en equipo, todos sus integrantes deben tener uno o varios roles asignados (incluso, un rol puede repartirse entre dos integrantes):
- Responsable del funcionamiento de la consulta xQuery: Debe garantizar que la consulta no lance excepciones, que devuelva el formato necesario para que lo consuma la plantilla XSLT y que devuelva los datos correctos.
 - Responsable del funcionamiento de la plantilla XSLT: Debe garantizar que la plantilla no lance excepciones, y que origine el documento LaTeX correspondiente.
 - Responsable del funcionamiento global del proyecto: Debe garantizar que todo el trabajo funcione bien “de punta a punta”, es decir, que las distintas partes del proyecto estén bien integradas entre sí (desde el XML origen hasta el archivo de salida final). Puede ser útil verificar que el proyecto funcione en las máquinas locales de todos los integrantes del equipo (especialmente si se usan distintos sistemas operativos).
 - Responsable de la presentación: Debe garantizar que el informe esté bien presentado (ie: sin errores de ortografía, etc.) y que represente el trabajo desplegado por todo el equipo. No necesariamente el responsable de la presentación debe escribir la totalidad del informe.

5. Entregables

Entregar en un archivo llamado **GrupoXX.zip** (donde **XX** es el número de grupo con dos dígitos), el siguiente material:

- Archivos de código:
 - `extract_data.xq` (Explicado en **4.b**).
 - `generate_report.xsl` (Explicado en **4.c**).
 - `tpe.sh` (Explicado en **4.d**).
 - Todo otro archivo que el grupo considere necesario para el correcto funcionamiento del trabajo.
- Informe:

El informe del trabajo debe estar dentro del .zip y debe contener:

 - Una carátula con el número de Grupo y sus integrantes.
 - Una Introducción.
 - Un Desarrollo explicando:
 - a) El trabajo realizado (con la definición de los errores, los archivos generados, etc).
 - b) Las dificultades encontradas (tanto técnicas como organizacionales).
 - c) Los aspectos que debieron investigarse (lo que no se vió en clase).
 - d) Los roles de cada uno de los integrantes.
 - Una sección de Conclusiones.

El informe se entregará en formato **PDF** y con una extensión no mayor a **3 páginas** (sin incluir carátula).

6. Evaluación

El Trabajo Práctico deberá entregarse por medio de Campus en la fecha de entrega establecida. Posteriormente, cada equipo hará una **defensa** frente a las docentes, en una

fecha a convenir. En ella, se preguntarán aspectos del trabajo y decisiones tomadas, así como se pedirá hacer corridas del programa.

La evaluación del trabajo se llevará a cabo teniendo en cuenta los parámetros establecidos en la rúbrica asociada a la actividad en Campus.

También se tendrá en cuenta que las consultas, más allá del funcionamiento (lo cual es fundamental), sean lo más genéricas posibles.

Las docentes ejecutarán las consultas con los parsers vistos en clase.

Se comprobará que **flights_data.xml** tenga la estructura solicitada en **4 b**).

El informe deberá estar completo y sin faltas de ortografía.

En caso de que el trabajo no cumpliera los requisitos básicos para ser aprobado, los alumnos serán citados en la fecha de recuperatorio para defenderlo y corregir los errores detectados.

No se aceptarán preguntas técnicas durante la semana de realización del Trabajo Práctico. Solamente se responderán preguntas estrictas del enunciado.

Anexo 1: Crear una nueva API Key de AirLabs

- 1) Crear una nueva cuenta gratuita en el sitio AirLabs (<https://airlabs.co/signup>)
- 2) Una vez creada la cuenta, loguearse y utilizar la API Key indicada en la página <https://airlabs.co/account>.
- 3) Para utilizar esa Key generada como variable de entorno, se puede ejecutar el siguiente comando: `export ENV_VARIABLE="<api_key_generada>"`

Anexo 2: Introducción/Repaso a L^AT_EX

L^AT_EX es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica.

Para realizar tablas extensas, se puede utilizar el package [longtable](#).

Se puede encontrar más información sobre cómo instalar L^AT_EX [aquí](#).

Una vez creado el archivo de L^AT_EX, se lo puede transformar a pdf utilizando un editor online como [Overleaf](#) o bien usando pdflatex (disponible con la distribución TeX).

A continuación se presenta un ejemplo de un documento L^AT_EX con una tabla:

```
\documentclass[a4paper,10pt]{article}
\usepackage{longtable}
\begin{document}
\title{Document title}
\date{\today}
\maketitle
\newpage
\begin{longtable}{| p{.12\textwidth}| p{.14\textwidth} |}
\hline
Id& Item & \\
\hline
001102&First Product & \\
00403A&l Second Product & \\
\hline
\end{longtable}
\end{document}
```