

Diseño y Procesamiento de Documentos XML

Segundo Parcial en Modalidad: Trabajo Práctico Especial

2do cuatrimestre 2023

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

El Rugby es un deporte de evasión y contacto, que se juega en equipo, y que se practica a nivel internacional en todos los continentes.

SportRadar (<https://developer.sportradar.com/>) es un sitio web que permite consumir datos y estadísticas actualizados de diversos deportes (ie: rugby, básquet, etc) en forma de invocaciones a APIs REST. De esa forma y dependiendo del deporte, se puede obtener listados de jugadores, partidos y estadísticas en formato XML o JSON.

En este Trabajo Práctico usaremos la API de SportRadar oficial de Rugby (https://developer.sportradar.com/docs/read/rugby/Rugby_v3), para obtener información relacionada a este deporte.

Se utilizarán como base los siguientes documentos XML:

- **seasons.xml**: Se obtiene a partir de llamar al método GET [seasons](#) de la API de SportRadar. Devuelve la lista de todas las temporadas registradas del deporte.
- **season_info.xml**: Se obtiene a partir de llamar al método GET [season_info](#) de la API de SportRadar. Dado un id de la temporada (ie: season_id), este método devuelve información detallada de la misma, incluyendo fases de competición y competidores.
- **season_lineups.xml**: Se obtiene a partir de llamar al método GET [season_lineups](#) de la API de SportRadar. Dado un id de la temporada (ie: season_id), este método devuelve la lista de todos sus partidos, incluyendo formaciones de equipos (entre otras estadísticas).

Para que las llamadas a la API sean correctas, se debe cumplir que:

- a) Debe incluir un parámetro llamado *api_key* con una Free API KEY de SportRadar. Los pasos para generarla se encuentran en el "Anexo 1".

- b) Para los métodos `season_info` y `season_lineups`: el parámetro `season_id` debe SER EL ID válido para una temporada (o season). La lista de `season_ids` válidos se encuentra en la salida generada con el método `seasons` (ie: `seasons.xml`).

Por ejemplo, para obtener el archivo `seasons.xml`, teniendo: `api_key` en una variable Bash llamada "API_KEY", se puede utilizar la aplicación `curl` de la siguiente forma:

```
curl http://api.sportradar.us/rugby-league/trial/v3/en/seasons.xml?api_key=${API_KEY} -o seasons.xml
```

Y así, generar el siguiente XML de salida:

```
<?xml version="1.0" encoding="UTF-8"?>
<seasons xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
generated_at="2023-11-01T11:45:00+00:00"
xmlns="http://schemas.sportradar.com/sportsapi/rugby-league/v3"
xsi:schemaLocation="http://schemas.sportradar.com/sportsapi/rugby-league/v3
https://schemas.sportradar.com/sportsapi/rugby-league/v3/schemas/seasons.xsd">
  <season id="sr:season:982" name="NRL 2007" start_date="2007-03-15" end_date="2007-10-01"
year="2007" competition_id="sr:competition:294"/>
  <season id="sr:season:1153" name="NRL 2008" start_date="2008-03-01" end_date="2008-10-13"
year="2008" competition_id="sr:competition:294"/>
  <season id="sr:season:1974" name="NRL 2009" start_date="2009-03-13" end_date="2009-10-05"
year="2009" competition_id="sr:competition:294"/>
  ...
  <season id="sr:season:105249" name="State of Origin, Women 2023" start_date="2023-06-01"
end_date="2023-06-22" year="2023" competition_id="sr:competition:39831"/>
  <season id="sr:season:111872" name="Pacific Championship, Women 2024"
start_date="2023-10-14" end_date="2023-11-30" year="2023"
competition_id="sr:competition:41292"/>
</seasons>
```

Asimismo, para obtener el archivo `season_info.xml` utilizando como parámetro `season_id=sr:season:80588`, se puede utilizar `curl` de la siguiente forma:

```
curl
http://api.sportradar.us/rugby-league/trial/v3/en/seasons/sr:season:80588/info.xml?api\_key=
\${API\_KEY} -o season_info.xml
```

Y así, generar el siguiente XML de salida:

```
<?xml version="1.0" encoding="UTF-8"?>
<season_info xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
generated_at="2023-11-01T12:30:42+00:00"
xmlns="http://schemas.sportradar.com/sportsapi/rugby-league/v3"
xsi:schemaLocation="http://schemas.sportradar.com/sportsapi/rugby-league/v3
https://schemas.sportradar.com/sportsapi/rugby-league/v3/schemas/season\_info.xsd">
```

```

    <season id="sr:season:80588" name="Super League 2021" start_date="2021-03-26"
end_date="2021-10-09" year="2021" competition_id="sr:competition:302">
    <sport id="sr:sport:12" name="Rugby"/>
    <category id="sr:category:83" name="Rugby League"/>
    <competition id="sr:competition:302" name="Super League" gender="men"/>
</season>
<stages>
    <stage order="1" type="league" phase="regular season" start_date="2021-03-26"
end_date="2021-09-19" year="2021">
    <groups>
        <group id="sr:league:54884" name="Super League" max_rounds="29">
            <competitors>
                <competitor id="sr:competitor:4275" name="Warrington Wolves"
abbreviation="WAR" gender="male"/>
                <competitor id="sr:competitor:4234" name="St Helens" abbreviation="STH"
gender="male"/>
                <competitor id="sr:competitor:4273" name="Wakefield Trinity"
abbreviation="WAK" gender="male"/>
                <competitor id="sr:competitor:4268" name="Leeds Rhinos"
abbreviation="LEE" gender="male"/>
                ...
                <competitor id="sr:competitor:7945" name="Catalans Dragons"
abbreviation="CAT" gender="male"/>
            </competitors>
        </group>
    </groups>
</stage>
</stages>
</season_info>

```

Asimismo, para obtener el archivo season_lineups.xml utilizando como parámetro season_id=sr:season:80588 , se puede utilizar curl de la siguiente forma:

```

curl
http://api.sportradar.us/rugby-league/trial/v3/en/seasons/sr:season:80588/info.xml?api\_key=\${API\_KEY} -o season_lineups.xml

```

Y así, generar el siguiente XML de salida:

```

<season_lineups xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.sportradar.com/sportsapi/rugby-league/v3"
generated_at="2023-11-01T12:58:23+00:00"
xsi:schemaLocation="http://schemas.sportradar.com/sportsapi/rugby-league/v3
https://schemas.sportradar.com/sportsapi/rugby-league/v3/schemas/season_lineups.xsd">
    <lineup>
        <sport_event id="sr:sport_event:25974124" start_time="2021-03-26T18:00:00+00:00"
start_time_confirmed="true">
            <sport_event_context>
                <sport id="sr:sport:12" name="Rugby"/>
                <category id="sr:category:83" name="Rugby League"/>
            </sport_event_context>
        </sport_event>
    </lineup>

```

```

<competition id="sr:competition:302" name="Super League" gender="men"/>
<season id="sr:season:80588" name="Super League 2021" start_date="2021-03-26"
end_date="2021-10-09" year="2021" competition_id="sr:competition:302"/>
<stage order="1" type="league" phase="regular season" start_date="2021-03-26"
end_date="2021-09-19" year="2021"/>
<round number="1"/>
<groups>
  <group id="sr:league:54884" name="Super League 2021"/>
</groups>
</sport_event_context>
<coverage live="true"/>
<competitors>
  <competitor id="sr:competitor:4234" name="St Helens" abbreviation="STH"
qualifier="home" gender="male"/>
  <competitor id="sr:competitor:4270" name="Salford Red Devils" abbreviation="SAL"
qualifier="away" gender="male"/>
</competitors>
<sport_event_conditions/>
</sport_event>
<sport_event_status status="closed" match_status="ended" home_score="29"
away_score="6" winner_id="sr:competitor:4234">
  <period_scores>
    <period_score home_score="13" away_score="0" type="regular_period" number="1"/>
    <period_score home_score="16" away_score="6" type="regular_period" number="2"/>
  </period_scores>
</sport_event_status>
<lineups>
  <competitors>
    <competitor id="sr:competitor:4270" name="Salford Red Devils" abbreviation="SAL"
qualifier="away" gender="male">
      <players>
        <player id="sr:player:505126" name="Sio, Ken" type="W"
date_of_birth="1990-10-29" nationality="Australia" country_code="AUS"
height="185" weight="95" jersey_number="2" starter="true" played="true"/>
        <player id="sr:player:471440" name="Watkins, Kallum" type="C"
date_of_birth="1991-03-12" nationality="England" country_code="ENG"
height="185" weight="98" jersey_number="3" starter="true" played="true"/>
        ...
        <player id="sr:player:793053" name="Patton, Declan" type="HB"
date_of_birth="1995-05-25" nationality="England" country_code="ENG"
height="180" weight="86"/>
      </players>
    </competitor>
    <competitor id="sr:competitor:4234" name="St Helens" abbreviation="STH"
qualifier="home" gender="male">
      <players>
        <player id="sr:player:953028" name="Bentley, James" type="BR"
date_of_birth="1996-10-19" nationality="Ireland" country_code="IRL"
height="175" weight="95" jersey_number="11" starter="true" played="true"/>
        ...
        <player id="sr:player:1080866" name="Smith, Aaron" type="HO"
date_of_birth="1996-10-12" nationality="England" country_code="ENG"
height="181" weight="85" jersey_number="19"/>
      </players>
    </competitor>
  </competitors>

```

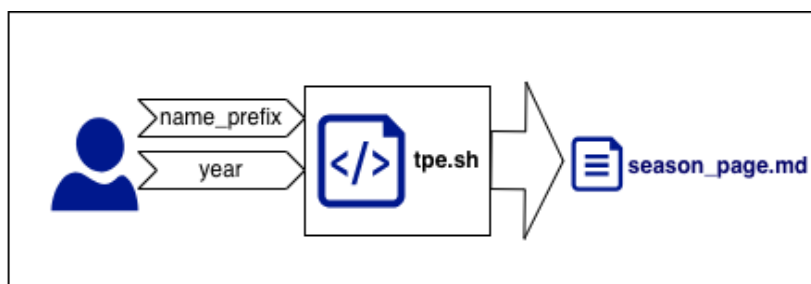
```
</competitors>
</lineups>
</lineup>
...
</season_lineups>
```

4. Procedimiento

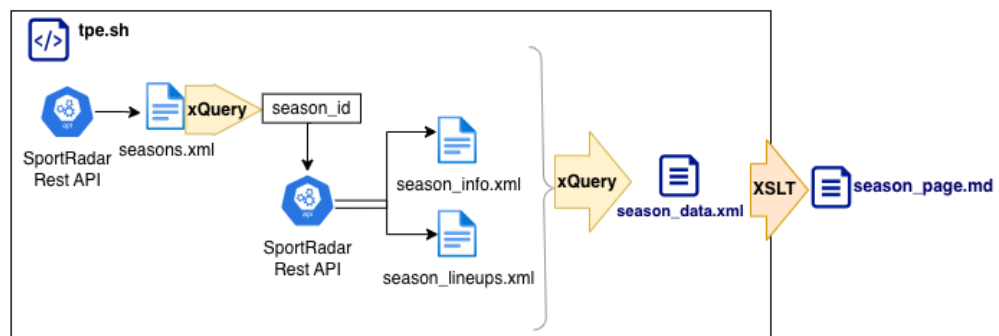
La finalidad del trabajo es obtener, a partir de la información mencionada en el punto 3, la página Markdown **season_page.md**, con la información de la temporada (o season) de Rugby, cuyo nombre comience con un prefijo (definido en un parámetro llamado "name_prefix"), y que pertenezca al año definido con el parámetro "year". La información a mostrar es la siguiente:

- Nombre completo de la temporada.
- Nombre completo de la competición.
- Género de la competición.
- Año de la temporada.
- Fases o grupos de la temporada, con la lista de los competidores en cada uno de ellos.
- Lista de todos los equipos de la temporada, con sus jugadores (indicando nombre, rol, fecha de nacimiento, nacionalidad, cantidad de partidos jugados, etc).

Los valores de *name_prefix* y *year* serán ingresados como parámetro.



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



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

- **seasons.xml**: Generado a partir de la API de SportRadar con el método “get seasons” (explicado en el punto 3). Se le deberá remover su XML Schema Namespace para que pueda ser utilizado correctamente en los pasos siguientes.
- **season_info.xml**: Generado a partir de la API de SportRadar con el método “get season_info” (explicado en el punto 3). Se le deberá remover su XML Schema Namespace para que pueda ser utilizado correctamente en los pasos siguientes.
- **season_lineups.xml**: Generado a partir de la API de SportRadar con el método “get season_lineups” (explicado en el punto 3). Se le deberá remover su XML Schema Namespace para que pueda ser utilizado correctamente en los pasos siguientes.

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

Además, por consola se ingresan los valores de los parámetros: **name_prefix** y **year**; siendo **name_prefix** el comienzo del nombre de una temporada (o season), y **year** el año en el que se desarrolló dicha temporada.

- ❖ Utilizando el archivo **seasons.xml** (descrito previamente), se pide crear la consulta XQuery **extract_season_id.xq** que reciba dos parámetros: *season_prefix* y *season_year*, y que produzca como salida solamente el texto de la temporada cuyo nombre comience con *season_prefix* y cuyo año sea *season_year*. En caso de haber más de un id que cumpla esa condición, se debe devolver únicamente el primer id (en base a su posición). Por ejemplo, teniendo como parámetros *season_prefix*="Super League" y *season_year*="2021", la salida correspondiente debe ser: "sr:season:80588".
- ❖ Utilizando los archivos descritos en a), se pide crear la consulta XQuery **extract_season_data.xq**, que debe producir como salida el documento XML que contenga el año, el nombre de la competencia, y el listado de sus corredores. El XML **season_data.xml** debe tener la estructura indicada en el siguiente XML Schema (llamado **season_data.xsd**).

season_data.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="season_data" type="resultT"/>
  <xsd:complexType name="resultT">
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="season" type="seasonT"/>
        <xsd:element name="stages">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="stage" type="stageT"
minOccurs="1" maxOccurs="unbounded" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="competitors">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="competitor" type="competitorT"
minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:element name="error" type="xsd:string" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="seasonT">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="competition">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="gender" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="date">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="start" type="xsd:string"/>
            <xsd:element name="end" type="xsd:string"/>
            <xsd:element name="year" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="stageT">

```

```
<xsd:sequence>
  <xsd:element name="groups">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="group" type="groupT" minOccurs="1"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
<xsd:attribute name="phase" type="xsd:string"/>
<xsd:attribute name="start_date" type="xsd:string"/>
<xsd:attribute name="end_date" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="groupT">
  <xsd:sequence>
    <xsd:element name="competitor" minOccurs="1" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="abbreviation" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="scoreT">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="score" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="competitorT">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="players">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="player" minOccurs="0"
maxOccurs="unbounded" type="playerT"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="playerT">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="type" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```



```
<xsd:element name="date_of_birth" type="xsd:string"/>
<xsd:element name="nationality" type="xsd:string"/>
<xsd:element name="events_played" type="xsd:int"/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>
```

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

- *season/name*
- *season/competition*
- *season/date*
- *season/year*
- *season_info/stages*

La información de estadísticas de los jugadores (name, type, date_of_birth, nationality y events_played) se deben extraer del XML **season_lineups.xml**. Para calcular el elemento events_played se deben contar aquellos partidos para los cuales el atributo @played es igual a "true".

❖ Crear la transformación XSLT **generate_markdown.xsl** que tomando los datos de **season_data.xml** produzca una página Markdown llamada **season_page.md** con las siguientes características.

- Heading nivel 1 con el texto: "Season {season_name}" (donde {season_name} es el nombre de la temporada).
- Heading nivel 3 con el siguiente texto: "Competition: {competition}" (siendo {competition} el contenido del elemento *season/competition/name* del XML).
- Párrafo con el siguiente texto: "Gender: {gender}" (siendo {gender} el contenido del elemento *season/competition/gender* del XML).
- Heading nivel 4 con el texto: "Year {year}. From {start_date} to {end_date}" (siendo {year}, {start_date} y {end_date} los elementos *season/date/year*, *season/date/start*, *season/date/end* respectivamente). Luego tiene que aparecer una línea divisoria.
- Por cada fase (o *stage*) debe aparecer:
 - Una línea divisoria
 - Heading nivel 4 con el texto: "{phase}. From {start_date} to {end_date}" (siendo {phase}, {start_date} y {end_date} los elementos *stage/@phase*, *stage/@start_date* y *stage/@end_date* del XML respectivamente).
 - Por cada grupo de dicha fase, se deben mostrar:
 - Línea divisoria
 - Heading nivel 4 con el texto "Competitors:"
 - Enumeración de los competidores, mostrando por cada línea el texto "{name} {abbreviation}" (siendo {name} y {abbreviation} los elementos *stage/competitor/name* y *stage/competitor/abbreviation* del XML respectivamente).
- Heading nivel 4 con el texto: "Teams:"
- Por cada competidor (elemento *stage/competitors/competitor*), se debe mostrar:

- o Heading nivel 4 con el texto: “{name}” (donde {name} es el elemento *stage//competitor/name* del XML).
- o Heading nivel 5 con el texto “Players:”
- o Ordenando los jugadores por el elemento *events_played* de forma descendente, se debe mostrar una tabla con las siguientes columnas:
 - Name
 - Type
 - Date of birth
 - Nationality
 - Events played
- Al final de toda la información de un competidor se debe poner otra línea, a modo de separación.

Season Super League 2021

Competition: Super League

Gender: men

Year: 2021. From 2021-03-26 to 2021-10-09

regular season. From 2021-03-26 to 2021-09-19

Competitors:

- Warrington Wolves (WAR)
- St Helens (STH)
- Wakefield Trinity (WAK)
- Leeds Rhinos (LEE)
- Wigan Warriors (WGW)
- Castleford Tigers (CAS)
- Salford Red Devils (SAL)
- Hull FC (HFC)
- Hull Kingston Rovers (HKR)
- Huddersfield Giants (HFG)
- Leigh Leopards (LEL)
- Catalans Dragons (CAT)

Teams**St Helens****Players:**

Name	Type	Date of birth	Nationality	Events played
McCarthy-Scarsbrook, Louie	PR	1986-01-14	Ireland	9
Bentley, James	BR	1996-10-19	Ireland	8
Simm, Josh	C	2001-02-27	England	5
Eaves, Josh	HO	1997-10-20	England	4
Davies, Ben	C	2000-04-21	England	3
Naiqama, Kevin	FB	1989-02-04	Fiji	21
Roby, James	HO	1985-11-22	England	20
Walmsley, Alex	UF	1990-04-10	England	20
Amor, Kyle	PR	1987-05-26	Ireland	2
Paasi, Agnatius	PR	1991-11-30	Tonga	2
Smith, Aaron	HO	1996-10-12	England	2
Wingfield, Jake	BR	2001-08-01	England	2
Grace, Regan	W	1996-12-12	Wales	19
Lomax, Jonny	FB	1990-09-04	England	19

Fages, Theo	HB	1994-08-23	France	10
Batchelor, Joe	BR	1994-10-28	England	10
Royle, Sam	BR	2000-02-12	England	1
Bennison, Jonathan	W	2002-12-01	England	1
Martin, Shay			England	1
Dan Norman	UF	1997-09-08	Ireland	0

...

Warrington Wolves

Players:

Name	Type	Date of birth	Nationality	Events played
Williams, George	HB	1994-10-31	England	8
Walker, Danny	HO	1999-06-29	England	6
Wrench, Connor	BR	2001-10-04	England	6
Ashton, Matty	W	1998-07-28	England	4
Robson, Ellis	BR	1998-09-14	England	4
Clark, Jason	PR	1989-06-28	Australia	3
Butler, Rob	PR	1998-05-15	England	3

...

Davis, Matt	BR	1996-07-06	England	10
Mulhern, Rob	C	1994-10-18	Ireland	1
Longstaff, Ellis	BR	2002-07-05	England	1
Whitehead, Tom			England	1
Dean, Riley	HB	2001-08-10	England	0
Doro, Eribe	UF	2001-03-26	England	0
Burgess, Morgan			England	0

Se pide que la página Markdown respete la estructura mostrada anteriormente.

- ❖ Para que todo el proceso se ejecute automáticamente se debe crear el script bash ***tpe.sh*** que genere la página Markdown ***season_page.md*** de salida, a partir de las transformaciones aplicadas en los puntos **4.a)**, **4.b)** y **4.c)**. Y debe recibir como parámetros de entrada:
 - name_prefix: comienzo del nombre de la temporada a buscar (es case sensitive). Debe ser un string no vacío.
 - year: Año de la competencia. Es un número entero mayor o igual a 2007.

Para realizar la llamada a la API de SportRadar, la API Key debe estar definida en la **variable de entorno** `API_KEY`. Y para generar los archivos `season_info.xml` y `season_lineups.xml` se debe utilizar como `season_id` la salida del archivo `extract_season_id.xq` (que conviene almacenarla en una variable local).

Por ejemplo: ejecutando `tpe.sh` con los parámetros: `name_prefix="Super League"` y `year=2021`, debería generarse como salida final el archivo `season_page.md` 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.

- ❖ **Errores:** Se deben manejar los errores posibles e informar las decisiones tomadas. Los errores deben aparecer en el archivo `season_data.xml` en uno o varios `<error>` dentro del tag `<season_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:

```
<season_data>
  <error>Year must not be empty</error>
</season_data>
```

Luego, la página Markdown resultante, deberá contener sólo dicha leyenda de error.

- ❖ 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):
 - i. 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.
 - ii. Responsable del funcionamiento de la plantilla XSLT: Debe garantizar que la plantilla no lance excepciones, y que origine el archivo Markdown correspondiente.
 - iii. 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).
 - iv. 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_season_id.xq` (Explicado en 4.b)).

- extract_season_data.xq (Explicado en **4.b**).
- generate_markdown.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, de forma virtual (por medio de Campus). En ella, se preguntarán aspectos del trabajo y decisiones tomadas, así como se pedirá hacer corridas del programa. Durante la misma, se le pedirá a cada integrante correr al menos una vez el trabajo práctico (en caso de no poder hacerlo, ese integrante deberá ir a recuperatorio).

Para la evaluación del trabajo, 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 **season_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, por medio del mail: xml@itba.edu.ar y por medio del Foro de Campus.

Anexo 1: Crear una nueva API Key de SportRadar

- 1) Crear una nueva cuenta gratuita en el sitio Developers SportRadar (<https://developer.sportradar.com/member/register>)
- 2) Una vez creada la cuenta, crear una nueva “application”, con formato XML y luego crear una API Key para “Rugby v3”.
- 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 a Markdown

Markdown es un lenguaje de markup ligero, con el cual podemos crear texto formateado usando un editor de texto. Se suele utilizar en mensajería instantánea, blogs, software colaborativo, páginas de documentación, archivos README, entre otras aplicaciones.

Los archivos Markdown suelen tener la extensión: “.md”, y son archivos de texto que pueden ser visualizados con Plugins o también editores de texto online.

Ejemplos de sintaxis:

- Encabezado de nivel 1: # Heading 1
- Encabezado de nivel 2: ## Heading 2
- Encabezado de nivel 3: ### Heading 3
- Encabezado de nivel 4: #### Heading 4
- Encabezado de nivel 5: ##### Heading 5
- Lista de ítems:
 - Ítem 1
 - Ítem 2
 - Ítem 3
- Línea divisoria: ---
- Tabla:

Col1 Col2 Col3
---- ----- -----
row1 row1 row3

Visualizadores de Markdown:

- Plugins para editores de texto (como Sublime, Atom, entre otros). Markdown Preview
- Visualizadores Markdown online: <https://dillinger.io/>

Para una guía más detallada, pueden consultar: <https://www.markdownguide.org/basic-syntax/>