

Diseño y Procesamiento de Documentos XML

Segundo Parcial en Modalidad: Trabajo Práctico Especial

1er 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 3 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 NASCAR (o “National Association for Stock Car Auto Racing”) es la categoría automovilística más popular de Estados Unidos; y también es considerada la competición de automóviles de serie más importante del mundo. Sus competiciones suelen correrse en circuitos ovals, y se utilizan vehículos cuyos diseños están basados en autos de fábrica.

SportRadar (<https://developer.sportradar.com/>) es un sitio web que permite consumir datos y estadísticas actualizados de diversos deportes (ie: básquet, automovilismo, hockey, 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 NASCAR (https://developer.sportradar.com/docs/read/racing/NASCAR_v3).

Se utilizarán como base los siguientes documentos XML:

- **drivers_standings.xml**: Se obtiene a partir de llamar al método GET [drivers_standings](#) de la API de SportRadar. Dados un año y un tipo de competición (o división), este método devuelve las estadísticas de los corredores que participaron en esa competencia.
- **drivers_list.xml**: Se obtiene a partir de llamar al método GET [drivers](#) de la API de SportRadar. Dados un año y un tipo de competición, este método devuelve la lista de los corredores que participaron en esa competencia.

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

- a) $2013 \leq \text{year} \leq 2021$
- b) *type* in (*sc*, *xf*, *cw*, *go*, *mc*, *enas*)
- c) Debe incluir un parámetro llamado *api_key* con una Free API KEY de SportRadar. Los pasos para generarla se encuentran en el “Anexo 1”.

Por ejemplo, para obtener el archivo `drivers_list.xml`, teniendo: *type=mc*, *year=2019*, y *api_key* en una variable Bash llamada “SPORTRADAR_API”, se puede utilizar la aplicación curl de la siguiente forma:

```
curl http://api.sportradar.us/nascar-ot3/mc/2019/drivers/list.xml?api_key=${SPORTRADAR_API}
-o drivers_list.xml
```

Y así, generar el siguiente XML de salida:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" charset="UTF-8" href="/xslt/nascar/series-v2.0.xsl"?>
<!-- Generation started @ 2021-06-07 09:21:31 UTC -->
<series xmlns="http://feed.elasticstats.com/schema/nascar/series-v2.0.xsd"
id="3e32047e-4ff3-4e35-a607-1546a2c32214" alias="CUP" name="NASCAR Cup Series">
  <season year="2019" id="d39477c5-3c73-4119-98f1-5b88958b524a">
    <driver id="4f63e2cc-e20e-43a4-ad00-1f54b6f7461d" first_name="Austin"
last_name="Theriault" full_name="Austin Theriault" birthday="1994-01-23" rookie_year="2015"
gender="M" status="ACT" country="UNITED STATES" residence=", , " birth_place="Fort Kent,
Maine, United States" twitter="@AustinTheriault" points_eligible="true">
      <team id="115610f5-11b9-43ff-a1b4-ef5cdab18119" name="Rick Ware Racing">
      </team>
    </driver>
    <driver id="f07507f1-0a84-4b90-929d-7389dc5baab4" first_name="Landon"
last_name="Cassill" full_name="Landon Cassill" height="67" birthday="1989-07-07"
rookie_year="2008" gender="M" status="ACT" country="UNITED STATES" residence=", , "
birth_place="Cedar Rapids, Iowa, United States" twitter="@landoncassill"
points_eligible="false">
      <team id="108ed8c5-066a-47bb-bfbd-651542f7e530" name="StarCom Racing">
      </team>
    </driver>
    <driver id="d259900a-e2e7-4612-b201-178bb80a862c" first_name="Josh" last_name="Bilicki"
full_name="Josh Bilicki" birthday="1995-06-03" rookie_year="2018" gender="M" status="ACT"
country="UNITED STATES" residence=", , " birth_place="Menomonee Falls, Wisconsin, United
States" points_eligible="false">
      <team id="115610f5-11b9-43ff-a1b4-ef5cdab18119" name="Rick Ware Racing">
      </team>
      <car id="52977af9-f68e-4646-a374-43f4bdcb97a6" number="52" sponsors="Wisconsin
Lighting Lab/Insurance King/Junction Fuels/Jacob Companies/BullShooter" status="ACT"
crew_chief="Jason Houghtaling">
        <manufacturer id="12844047-ab59-43ca-ab6c-c10d54931715" name="Ford"/>
        <engine id="2ff1e55b-7433-4e77-9336-07a74cc3194e" name="Ford"/>
        <owner id="41c22ea5-0caf-459c-9e97-0a13d3e6c086" name="Rick Ware"/>
        <team id="115610f5-11b9-43ff-a1b4-ef5cdab18119" name="Rick Ware Racing">
        </team>
      </car>
    </driver>
    <driver id="6e5a779b-11fa-46bf-bfc9-51efedfe66c4" first_name="Gray"
last_name="Gaulding" full_name="Gray Gaulding" birthday="1998-02-10" rookie_year="2014"
gender="M" status="ACT" country="UNITED STATES" residence=", , " birth_place="Colonial
Heights, Virginia, United States" twitter="@graygaulding" points_eligible="false">
      <team id="115610f5-11b9-43ff-a1b4-ef5cdab18119" name="Rick Ware Racing">
      </team>
    </driver>
    ...
```

```

    <driver id="986c20d7-178a-4d77-bb46-dde8d453dda5" first_name="Blake" last_name="Jones"
full_name="Blake Jones" birthday="1997-01-12" gender="M" status="ACT" country="UNITED
STATES" residence=", , " birth_place="Sevierville, Tennessee, United States"
points_eligible="false">
    <team id="9f742920-6e44-4581-8a61-e8c38cf21da4" name="Spire Motorsports">
    </team>
    </driver>
  </season>
</series>

```

Asimismo, para obtener el archivo drivers_standings.xml utilizando los mismos parámetros, se puede utilizar curl de la siguiente forma:

```

curl
http://api.sportradar.us/nascar-ot3/mc/2019/standings/drivers.xml?api_key=${SPORTRADAR_API}
-o drivers_standings.xml

```

Y así, generar el siguiente XML de salida:

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" charset="UTF-8" href="/xslt/nascar/standings-v2.0.xsl"?>
<!-- Generation started @ 2021-06-12 19:28:16 UTC -->
<series xmlns="http://feed.elasticstats.com/schema/nascar/standings-v2.0.xsd"
id="3e32047e-4ff3-4e35-a607-1546a2c32214" alias="CUP" name="NASCAR Cup Series">
  <season year="2019" id="d39477c5-3c73-4119-98f1-5b88958b524a">
    <driver rank="1" id="85c06960-3610-4eff-9ef2-18641b84c952" first_name="Kyle"
last_name="Busch" full_name="Kyle Busch" status="ACT" points="5040" starts="36"
wins="5" poles="1" stage_wins="11" chase_bonus="51" chase_wins="1"
chase_stage_wins="1" top_5="17" top_10="27" top_15="30" top_20="32" dnf="2"
laps_led="1582" laps_completed="10167" money="0.0" avg_start_position="12.2778"
avg_finish_position="8.9444" avg_laps_completed="282.4167" laps_led_pct="15.56"
in_chase="true"/>
    <driver rank="2" behind="5" id="4d95ef3a-46cc-43c2-b340-c92cea2ed7f8"
first_name="Martin" last_name="Truex" full_name="Martin Truex Jr" status="ACT"
points="5035" starts="36" wins="7" poles="0" stage_wins="8" chase_bonus="49" chase_wins="3"
chase_stage_wins="5" top_5="15" top_10="24" top_15="27" top_20="31" dnf="2" laps_led="1371"
laps_completed="10152" money="0.0" avg_start_position="13.1111" avg_finish_position="9.75"
avg_laps_completed="282.0" laps_led_pct="13.5" in_chase="true"/>
    <driver rank="3" behind="7" id="9d18716c-fd73-48d4-b3da-44228c767b84"
first_name="Kevin" last_name="Harvick" full_name="Kevin Harvick" status="ACT" points="5033"
starts="36" wins="4" poles="5" stage_wins="6" chase_bonus="34" chase_wins="1"
chase_stage_wins="1" top_5="15" top_10="26" top_15="29" top_20="30" dnf="3" laps_led="953"
laps_completed="9804" money="0.0" avg_start_position="8.5" avg_finish_position="10.0278"
avg_laps_completed="272.3333" laps_led_pct="9.72" in_chase="true"/>
    <driver rank="4" behind="13" id="cb9d6f49-0044-4305-be9a-22d428e52c95"
first_name="Denny" last_name="Hamlin" full_name="Denny Hamlin" status="ACT" points="5027"
starts="36" wins="6" poles="3" stage_wins="6" chase_bonus="43" chase_wins="2"
chase_stage_wins="3" top_5="19" top_10="24" top_15="28" top_20="31" dnf="1" laps_led="922"
laps_completed="10124" money="0.0" avg_start_position="11.0" avg_finish_position="9.4722"
avg_laps_completed="281.2222" laps_led_pct="9.11" in_chase="true"/>
    ...
  
```

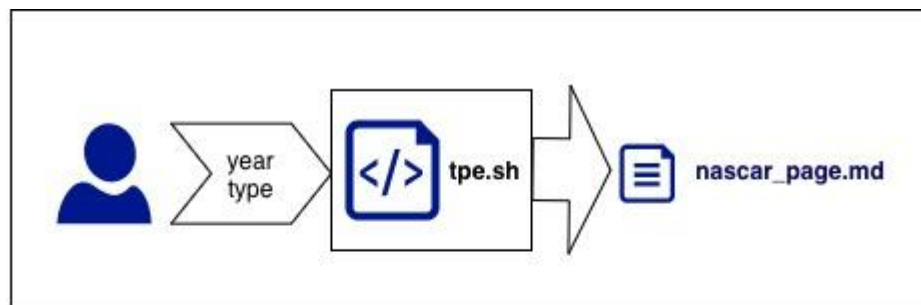
```
<driver      rank="64"      behind="-5040"      id="6e5a779b-11fa-46bf-bfc9-51efedfe66c4"
first_name="Gray" last_name="Gaulding" full_name="Gray Gaulding" status="ACT" points="0"
starts="1"      wins="0"      poles="0"      stage_wins="0"      chase_bonus="0"      chase_wins="0"
chase_stage_wins="0" top_5="0" top_10="0" top_15="0" top_20="0" dnf="1" laps_led="0"
laps_completed="142" money="0.0" avg_start_position="33.0" avg_finish_position="36.0"
avg_laps_completed="142.0" laps_led_pct="0.0" in_chase="false"/>
</season>
</series>
<!-- Generation ended @ 2021-06-12 19:28:16 UTC -->
```

4. Procedimiento

La finalidad del trabajo es obtener, a partir de la información mencionada en el punto 3, una página Markdown con la información de los corredores de la competición (o división) NASCAR de tipo “type”, para el año “year”, ordenados alfabéticamente según su nombre. La información a mostrar, para cada una de ellos, es la siguiente:

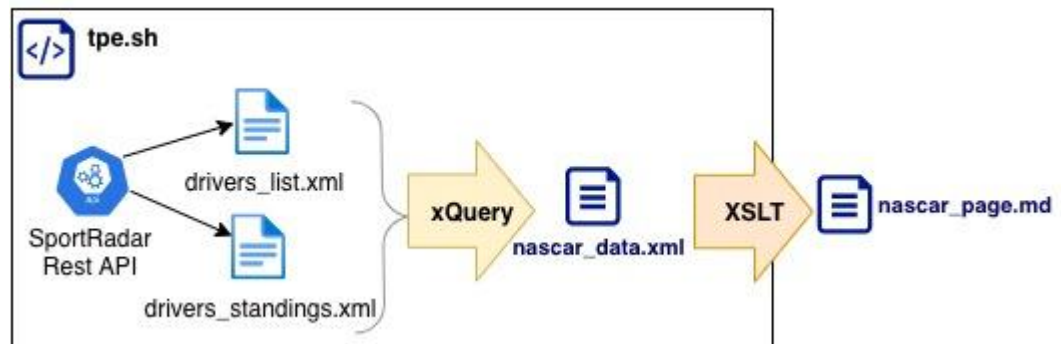
- Nombre completo del corredor.
- País.
- Lugar de Nacimiento.
- Fecha de Nacimiento.
- Nombre del fabricante al que representa.
- Puesto en la clasificación (si existe).
- Estadísticas: puntos obtenidos; carreras ganadas, poles, carreras sin finalizar, vueltas completadas.

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



Previo a la generación de la página Markdown, se debe crear un documento XML (**nascar_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:

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

- ❖ Utilizando los archivos descritos en a), se pide crear la consulta XQuery **extract_nascar_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 **nascar_data.xml** debe tener la estructura indicada en el siguiente XML Schema (llamado **nascar_data.xsd**).

nascar_data.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="nascar_data" type="resultT"/>
<xsd:complexType name="resultT">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="year" type="xsd:int"/>
      <xsd:element name="serie_type" type="xsd:string"/>
      <xsd:element name="drivers">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="driver" type="driverT" minOccurs="1"
maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>

```

```

        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
    <xsd:element name="error" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
</xsd:choice>
</xsd:complexType>

<xsd:complexType name="driverT">
    <xsd:sequence>
        <xsd:element name="full_name" type="xsd:string"/>
        <xsd:element name="country" type="xsd:string"/>
        <xsd:element name="birth_date" type="xsd:string"/>
        <xsd:element name="birth_place" type="xsd:string"/>
        <xsd:element name="rank" type="xsd:string"/>
        <xsd:element name="car" type="xsd:string" minOccurs="0"/>
        <xsd:element name="statistics" type="statisticsT"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="statisticsT">
    <xsd:sequence>
        <xsd:element name="season_points" type="xsd:string"/>
        <xsd:element name="wins" type="xsd:string"/>
        <xsd:element name="poles" type="xsd:string"/>
        <xsd:element name="races_not_finished" type="xsd:string"/>
        <xsd:element name="laps_completed" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

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

- *year*
- *serie_type* (que es el atributo *name* del elemento *series*)
- *rank* (para cada uno de los corredores)
- Aquellos que se encuentran dentro de "statistics" (*season_points*, *wins*, *poles*, *races_not_finished* y *laps_completed*)

La información personal de los corredores (*full_name*, *country*, *birth_date*, *birth_place* y *car*) se deben extraer del XML **drivers_list.xml**.

El elemento *car* debe aparecer solamente si el corredor tiene por lo menos un auto; y su contenido debe ser el nombre del fabricante (o *manufacturer*) del primer auto que aparezca (en el XML) que le pertenece.

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

- Heading nivel 1 con el siguiente contenido: "Drivers for [serie_type] for [year] season" (siendo *year* y *serie_type* valores extraídos del XML). Luego tienen que aparecer dos líneas divisorias.

- Por cada corredor, debe aparecer un heading nivel 3 indicando su nombre completo. Seguido de ello, se mostrará una lista ordenada de los siguientes ítems:
 - Country
 - Birth date
 - Birthplace
 - Car manufacturer: Valor definido en *car*. Si el elemento no existe, debe mostrarse un guión (“-”)
 - Rank: Valor definido en *rank*. Si el elemento está vacío, debe mostrarse un guión (“-”)
- Si el elemento *rank* no está vacío, se debe mostrar, además, un heading nivel 5 llamado “Statistics”, seguido de una lista no ordenada de la siguiente información:
 - Season points
 - Wins
 - Poles
 - Races not finished
 - Laps completed
- Al final de toda la información de un corredor se debe poner otra línea, a modo de separación.

Tener en cuenta que la lista de corredores debe estar **ordenada alfabéticamente** por el nombre completo del corredor. La página Markdown debería tener un contenido similar al siguiente ejemplo (en este caso se muestran solamente las primeras líneas del documento final a modo de guía):

Drivers for NASCAR Cup Series for 2019 season

Alex Bowman

1. Country: UNITED STATES
2. Birth date: 1993-04-25
3. Birthplace: Tucson, Arizona, United States
4. Car manufacturer: Chevrolet
5. Rank: 12

Statistics

- Season points: 2257
- Wins: 1
- Poles: 0
- Races not finished: 2
- Laps completed: 2

Andy Seuss

1. Country: UNITED STATES
2. Birth date: 1987-03-15
3. Birthplace: Hempstead, New Hampshire, United States
4. Car manufacturer: -
5. Rank: 36

Statistics

- Season points: 9
- Wins: 0
- Poles: 0
- Races not finished: 0
- Laps completed: 0

Aric Almirola

1. Country: UNITED STATES
2. Birth date: 1984-03-14
3. Birthplace: Tampa, Florida, United States
4. Car manufacturer: Ford
5. Rank: 14

Statistics

- Season points: 2234
- Wins: 0
- Poles: 1
- Races not finished: 3
- Laps completed: 3

Austin Cindric

1. Country: UNITED STATES
2. Birth date: 1998-09-02
3. Birthplace: Mooresville, North Carolina, United States
4. Car manufacturer: Ford
5. Rank: -

Queda a entera decisión del equipo el diseño de la página Markdown (siguiendo las características principales solicitadas en este punto).

- ❖ Para que todo el proceso se ejecute automáticamente se debe crear el script bash ***tpe.sh*** que genere la página Markdown ***nascar_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:
 - **year**: Año de la competencia. En un entero entre 2013 y 2021 inclusive.
 - **type**: Indica el tipo de competencia. Puede ser alguno de los siguientes valores: *sc*, *xf*, *cw*, *go*, *mc* y *enas*.

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

Por ejemplo: ejecutando `tpe.sh` con los parámetros: `year=2` y `type=mc`, debería generarse como salida final el archivo `nascar_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 `nascar_data.xml` en uno o varios `<error>` dentro del tag `<nascar_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:

```
<nascar_data>
  <error>Year must not be empty</error>
</nascar_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_nascar_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. 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 **nascar_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 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 “NASCAR 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
- Lista ordenada de ítems:
 1. Ítem 1
 2. Ítem 2

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/>