

MTI-476 Sistemas y Aplicaciones Web

Tarea Final: Portal web de compra/venta de instrumentos

María Elena Arellano Bravo, Aníbal Eduardo Corral Serrano
maria.arellano@sansano.usm.cl, anibal.corral@sansano.usm.cl

Carolina Isabel Flores San Martín, Hipólito Fabián Guerrero Castro
carolina.floress@sansano.usm.cl, hipolito.guerrero@sansano.usm.cl

Profesor Jose Emilio Labra Gayo

Viernes, 07 de Julio de 2017

Palabras claves: XML/Json, DTD/XML Schema, Json Schema, Web Semántica, Rdfa.

1. Representación de la información

1.1. Representación de información en ficheros XML y JSON.

- ☐ En la primera parte se representa la información con los archivos XML: establecimiento.xml, inventario.xml, producto.xml y venta.xml.
- ☐ Los archivos Json que representan la información son: establecimiento.json, inventario.json, producto.json y venta.json.

Disponibles en <https://github.com/anibal-corral/SitiInstrumentosMTI/>

1.2. Crear vocabularios mediante DTD, XML Schema para validar ficheros XML y JSON Schema para validar la representación Json.

- ☐ Se crea los archivos establecimiento.dtd y producto.dtd para validar los archivos xml creados en el paso anterior.

Para la validación¹ se puede ejecutar los siguientes comandos:

```
xmllint --noout --valid establecimiento.xml  
xmllint --noout --valid inventario.xml  
xmllint --noout --valid producto.xml  
xmllint --noout --valid venta.xml
```

- ☐ Para la validación de los archivos mediante XML Schema se generaron los archivos establecimiento.xsd, inventario.xsd, producto.xsd y venta.xsd.
- ☐ Para la validación de la información de los archivos Json se generaron los archivos establecimiento.schema.json, inventario.schema.json, producto.schema.json y venta.schema.json.

¹ Software entregado en el curso MTI-476 Sistemas y Aplicaciones Web.

1.3. Realizar comparación entre el formato XML y formato JSON.

XML y JSON son dos formatos de intercambio de datos usados por diversas aplicaciones. Para realizar una comparación entre ambos, es necesario comentar brevemente las principales características de cada uno de ellos.

XML (Lenguaje de Marcado Extensible) es un subconjunto del Lenguaje de Marcado Generalizado Estándar (SGML). Constituye un formato más especializado para describir un árbol compuesto de "nodos" que tienen su propia semántica y pueden ser serializados en JSON [1]. XML es usado para crear anotaciones definidas por el usuario en documentos y esquemas codificados. Los principales usos de XML son las Llamadas a Procedimientos Remotos (RPC) y la serialización de objetos para transferir datos entre aplicaciones [2].

JSON es un formato genérico para describir estructuras de datos hechas de mapas, arreglos y algunos valores de de tipo simple[1]. Está diseñado para ser un lenguaje de intercambio de datos que es legible por los seres humanos y fácil para que las computadoras puedan analizar y utilizar. JSON está directamente soportado dentro de JavaScript y es el formato más adecuado para aplicaciones de este tipo. Esto, proporciona ganancias de rendimiento significativas sobre XML, puesto que este último necesita bibliotecas adicionales para recuperar datos de los objetos DOM (Modelo de Objetos del Documento). Se estima que JSON analiza hasta cien veces más rápido que XML en los navegadores modernos, pero a pesar de estas afirmaciones que muestran un rendimiento notable, los argumentos contra JSON incluyen: la falta de soporte de espacio de nombres, la falta de validación de entrada y algunos inconvenientes de extensibilidad [2].

Las principales similitudes entre ambos son [3]:

- ☐ Tanto JSON como XML son "autodescriptivos", es decir, son legibles para un ser humano.
- ☐ JSON y XML son jerárquicos, por lo que soportan valores dentro de otros valores.
- ☐ Ambos pueden ser analizados y utilizados por muchos lenguajes de programación.
- ☐ Entre las principales diferencias destacan [3]:
- ☐ JSON no utiliza etiquetas en los extremos, XML si lo hace.
- ☐ JSON es más corto que XML.
- ☐ JSON se hace más fácil y rápido para leer y escribir, a diferencia de XML.
- ☐ JSON permite el uso de arreglos a diferencia de XML.

Elegir adecuadamente el formato de intercambio de datos a utilizar puede tener consecuencias significativas, tanto en la tasa de transmisión de datos como en el desempeño global; ambos, aspectos extremadamente importantes a la hora de desarrollar cualquier aplicación [2].

Hoy por hoy, es admitido incluso en aquellas comunidades XML más tradicionalistas, que la habilidad de integrar JSON y HTML5 perfectamente, es la clave para el futuro de las aplicaciones web y la clave para el futuro del ecosistema XML. Por tal motivo, se han realizado muchos esfuerzos para lograr definir una representación estándar de XMD (es un subsistema CDL que importa catálogos que contienen metadatos relativos a textos y composiciones y crea una versión XML de los datos[4]) en JSON, lo que podría ayudar al conocimiento de esta relación [2] y a la implementación de herramientas que faciliten una coexistencia pacífica entre ambos formatos, como así también, entre sus seguidores.

2. Aplicación Web

2.1. Servicio web REST

- ❑ La aplicación web y las indicaciones del proyecto se encuentran disponibles en el siguiente link: <https://github.com/anibal-corral/SitiInstrumentosMTI>.

Para realizar esta implementación se consultaron las fuentes disponibles en el punto 4.2.

2.2. Documentación del servicio

- ❑ La documentación de la API en tanto se puede revisar accediendo al link: [API-InstrumentosMTI](#)
- ❑ El despliegue del cliente web está en GCP: <http://35.185.162.103/index.html>
- ❑ El despliegue de la api Node.js está en GCP:
<https://mti476-171817.appspot.com/api/establecimiento/>
<https://mti476-171817.appspot.com/api/producto/>

2.3. Comparación de frameworks web [5] [6] [7]

Tomando una de las características por las que se destaca Node.js, denominado “event loop”, que a groso modo es un único subproceso que realiza las operaciones de I/O de forma asíncrona cada vez que llega una petición, se viene a la mente (como desarrollador de Java) Servlet 3.0 que al igual que Node.js, permiten trabajar aún cuando hay una cantidad de peticiones importantes, así también utilizar múltiples CPU y frameworks como SpringMVC que está basado en las especificaciones de Servlet 3.0. Esta fue una de las características que más destacan la gente de **noders**[8] y creemos que tienen razón en ello.

Non-Blocking I/O de Node.js es otra de las características que más se enfatizan en los artículos/tutoriales leídos/vistos, pero también tiene su símil en tecnologías JAVA a través del estándar NIO, que es una librería de entrada/salida introducida con JDK 1.4[9].

Sin embargo, a pesar de contar con la característica “event loop” el hecho de que todo pase por un único thread genera inquietud en, si se ejecutan grandes tareas, como se harían en aplicaciones a nivel empresarial en las cuales el volumen de transacciones es alto, quedarían todas encoladas y entregaría un bajo desempeño. Otra de las desventajas de Node.js - después de haber investigado en varias fuentes de información y no encontrando información que lo desmienta - es que sólo utiliza un núcleo del procesador, así que si el servidor tiene más de un procesador, estos no serían utilizados por Node.js. Imaginamos que haciendo algún tipo de customización se podrá corregir eso.

Otra de las cuestiones que juegan en contra a Node.js, o mejor dicho hay que manejar con mucho cuidado, es lo referente a los “callbacks”, ya que podría formarse un “espagueti de código” si no se controla bien y luego el seguimiento sería mucho más costoso. De hecho, dentro del ejercicio del sitio de Instrumentos, una de las dificultades fue trabajar con esto.

El uso del NPM para la gestión de paquetes es realmente práctico, a pesar de no haberlo utilizado antes nos pareció bastante sencillo. Sin embargo, a pesar de ser muy práctico también genera la duda de qué tan confiables son los códigos fuentes o módulos que uno instala o las dependencias que necesita cada módulo, ¿se puede asegurar que no hay un robo de información y/o están espiando mis datos?. Fuera de ello, pareció bastante útil y sencillo como ya se mencionó.

Si fuese necesario sólo referirse a frameworks que trabajen en el front-end se podría nombrar, como ejemplo, a Ember.js, AngularJS (que fue el utilizado para el front-end del sitio Instrumentos), Backbone.js, React, cada uno con sus pros y contras en el contexto de front-end, sin embargo, Node.js se

encarga del modelo y el controlador, en otros escenarios se tendría que utilizar diferentes lenguajes para escribir una aplicación web, sin embargo, con Node.js se pueden crear aplicaciones completas en JavaScript sin necesidad de utilizar otro lenguaje como Python o Ruby, lo que hace más útil aún utilizar Node.js.

Otra de las características positivas es la cantidad frameworks que pueden ser utilizados con Node.js para hacer las tareas mucho más sencillas, como puede ser, conectarse a una base de datos por ejemplo, frameworks como Express, Hapi.js, Sails.js, Mojito o Koa hacen este trabajo mucho más sencillo.

En el contexto del trabajo que hubo que desarrollar, creo que utilizar Node.js a modo de prueba o marcha blanca en algún(nos) proyecto(s) de pequeña o mediana envergadura en la empresa sería recomendable, ya que hay temas como los callbacks y el seguimiento del código para las mantenciones que generan un poco de inquietud (teniendo en consideración el gap existente hoy en día del equipo con esta tecnología). No se encontró en la investigación cuadros comparativos de entidades reconocidas como Gartner, pero si existe un sitio interesantes para ahondar en temas relacionados a la comparación de frameworks[10].

En el contexto de desplegar el sistema desarrollado con NodeJS y AngularJS (ya que con estas tecnologías se construyó el trabajo), se estima sin duda que la mejor opción serían plataformas en la nube tales como Google Cloud Platform o Microsoft Azure. ¿Por qué?, porque permiten beneficiarse del uso de los recursos “ilimitados” que ofrece la nube, no es necesario invertir una gran cantidad de dinero en la compra de servidores ni licencias para poder comenzar. En el caso de que el sitio web sea exitoso y requiera de más capacidad, es fácilmente escalable y no es necesario sobre-provisionar capacidad. Al pasar del tiempo, se pueden utilizar los servicios de la nube para realizar labores de Big-Data que ayuden a mejorar aspectos del negocio.

Así también, en este contexto, los datos de los “Instrumentos” que se ofrecen no son de carácter “sensible” por lo tanto, también podrían almacenarse en la nube, no así si fuese una entidad financiera o aseguradora. La idea de microservicios por el momento no se considera ya que la aplicación no es tan compleja, y en el caso de que se complejizara, entonces se podrían tomar los servicios más críticos y moverlos a una arquitectura de microservicios, para posteriormente balancear la carga en aquellos servicios.

Sin entrar en detalle de qué proveedor de la nube es mejor que otro o si Heroku es más intuitivo que Amazon, se estima que hacer uso de la nube en términos de Plataforma como Servicios (PaaS) sería una buena apuesta, más aún cuando algunos proveedores de la nube hoy han invertido en estas nuevas tecnologías como Node.js. El proyecto fue desplegado en la plataforma de Google (GCP) y no existieron mayores inconvenientes, ya que el proveedor entregaba todas las herramientas para realizar esta tarea, como por ejemplo, el SDK desde el cual es posible subir nuevos release del proyecto utilizando comandos simples como **gcloud** desde la consola.

3. Web semántica

3.1. Conversión de XML a RDF

Representar la información de establecimientos y productos musicales de la primer tarea en formato [RDF](#). Opcionalmente, se podrá utilizar [ShEx](#) o [SHACL](#) para validar los ficheros creados (una posible herramienta sería para la validación sería [RDFShape](#) o [Shaclex](#)).

Para llevar a cabo el siguiente punto, se han utilizado los archivos establecimiento.xml y producto.xml, los cuales fueron generados anteriormente, para lo cual se utilizó el editor SUBLIME TEXT y la validación se hizo en el siguiente link: <https://www.w3.org/RDF/Validator/>, en el mismo link se generó el grafo resultante del archivo rdf/turtle.

- ❑ Link establecimiento.xml original:
<https://github.com/anibal-corral/SitiInstrumentosMTI/blob/master/entrega/parte%201/establecimiento.xml>
- ❑ Link archivo rdf.establecimiento.xml:
<https://github.com/anibal-corral/SitiInstrumentosMTI/blob/master/entrega/parte%203/rdf.establecimiento.xml>

Código archivo rdf.establecimiento.xml

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:ex="http://www.example.org/store"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:sch="http://schema.org/">
  <rdf:Description rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/">

    <sch:Place>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/establecimiento">
        <sch:name>Tienda USM</sch:name>
        <sch:Thing>
          <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/rut">
            <sch:Number>98765000</sch:Number>
            <sch:Number>0</sch:Number>
          </rdf:Description>
        </sch:Thing>

        <sch:PostalAddress>
          <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/direccion">
            <sch:addressCountry>CHILE</sch:addressCountry>
            <sch:addressRegion>Metropolitana</sch:addressRegion>
            <sch:addressLocality>SANTIAGO</sch:addressLocality>
            <sch:addressLocality>Vitacura</sch:addressLocality>
            <sch:streetAddress>Avenida Santa Maria</sch:streetAddress>
            <sch:PostalCode>6400</sch:PostalCode>
            <sch:GeoCoordinates>
              <rdf:Description rdf:about="http://schema.org/GeoCoordinates">
```



```

<sch:longitude>-70.5774312</sch:longitude>
<sch:latitude>-33.3775228</sch:latitude>
</rdf:Description>
</sch:GeoCoordinates>
</rdf:Description>
</sch:PostalAddress>

<sch:ContactPoint>
  <rdf:Description rdf:about="http://schema.org/telefono">
    <sch:areaServed>56</sch:areaServed>
    <sch:areaServed>32</sch:areaServed>
    <sch:telephone>2654678</sch:telephone>
  </rdf:Description>
</sch:ContactPoint>
<sch:openingHours>Sin información</sch:openingHours>
</rdf:Description>
</sch:Place>
</rdf:Description>
</rdf:RDF>

```

Luego se ha validado el archivo en <https://www.w3.org/RDF/Validator/>

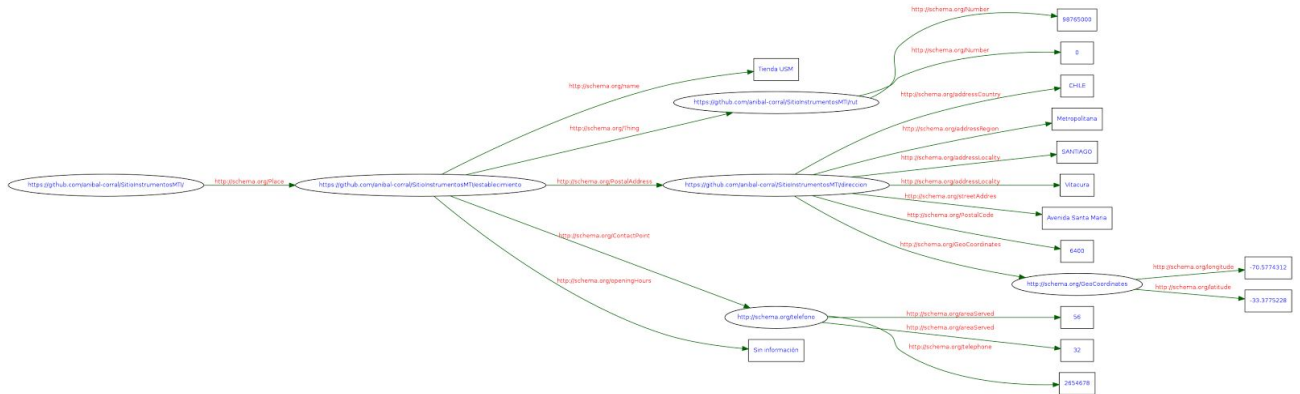
Validation Results

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	https://github.com/anibal-corral/SitioInstrumentosMTI/	http://schema.org/Place	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento
2	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento	http://schema.org/name	"Tienda USM"
3	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento	http://schema.org/Thing	https://github.com/anibal-corral/SitioInstrumentosMTI/rut
4	https://github.com/anibal-corral/SitioInstrumentosMTI/rut	http://schema.org/Number	"98765000"
5	https://github.com/anibal-corral/SitioInstrumentosMTI/rut	http://schema.org/Number	"0"
6	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento	http://schema.org/PostalAddress	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion
7	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/addressCountry	"CHILE"
8	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/addressRegion	"Metropolitana"
9	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/addressLocality	"SANTIAGO"
10	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/addressLocality	"Vitacura"
11	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/streetAddress	"Avenida Santa Maria"
12	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/PostalCode	"6400"
13	https://github.com/anibal-corral/SitioInstrumentosMTI/direccion	http://schema.org/GeoCoordinates	http://schema.org/GeoCoordinates
14	http://schema.org/GeoCoordinates	http://schema.org/longitude	"-70.5774312"
15	http://schema.org/GeoCoordinates	http://schema.org/latitude	"-33.3775228"
16	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento	http://schema.org/ContactPoint	http://schema.org/telefono
17	http://schema.org/telefono	http://schema.org/areaServed	"56"
18	http://schema.org/telefono	http://schema.org/areaServed	"32"
19	http://schema.org/telefono	http://schema.org/telephone	"2654678"
20	https://github.com/anibal-corral/SitioInstrumentosMTI/establecimiento	http://schema.org/openingHours	"Sin información"

y finalmente se muestra el grafo resultante del archivo rdf/turtle:



- ❑ Link producto.xml original:
<https://github.com/anibal-corral/SitiolInstrumentosMTI/blob/master/entrega/parte%201/producto.xml>
- ❑ Link archivo rdf.producto.xml:
<https://github.com/anibal-corral/SitiolInstrumentosMTI/blob/master/entrega/parte%203/rdf.producto.xml>

Código archivo rdf.producto.xml

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:ex="http://www.example.org/store"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:sch="http://schema.org/">
  <rdf:Description rdf:about="https://github.com/anibal-corral/SitiolInstrumentosMTI/">
    <sch:Product>
      <rdf:Description
        rdf:about="https://github.com/anibal-corral/SitiolInstrumentosMTI/producto">
        <sch:number>737383</sch:number>
        <sch:Name>Guitarra eléctrica GRX22 color candy apple (CA)
Ibanez</sch:Name>
        <sch:description>Guitarra eléctrica Marca Ibanez modelo
GRX22. Una guitarra no tiene que costar una fortuna para tener un buen sonido, es el caso
de la serie GRX de Ibanez, fue desarrollada para los músicos que desean la calidad de esta
marca, pero a un precio asequible.</sch:description>
        <sch:brand>Ibanez</sch:brand>
        <sch:model>GRX22</sch:model>
        <sch:category>
          <rdf:Description
            rdf:about="https://github.com/anibal-corral/SitiolInstrumentosMTI/producto">
            <sch:place>Guitarra eléctrica</sch:place>
            </rdf:Description>
          </sch:category>
```



```

    <sch:ProductModel>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/caracteristicas">
        <sch:name>ESCALA</sch:name>
          <sch:ProductModel>25.5
pulgadas</sch:ProductModel>
        </rdf:Description>
      </sch:ProductModel>

    <sch:ProductModel>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/caracteristicas">
        <sch:name>NÚMERO DE CUERDAS</sch:name>
          <sch:number>6</sch:number>
        </rdf:Description>
      </sch:ProductModel>

    <sch:ProductModel>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/caracteristicas">
        <sch:name>COLOR</sch:name>
          <sch:ProductModel>Candy Apple
(CA)</sch:ProductModel>
        </rdf:Description>
      </sch:ProductModel>

    <sch:ProductModel>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/caracteristicas">
        <sch:name>MATERIAL NECK</sch:name>
        <sch:ProductModel>Maple</sch:ProductModel>
        </rdf:Description>
      </sch:ProductModel>

    <sch:ProductModel>
      <rdf:Description
rdf:about="https://github.com/anibal-corral/SitiInstrumentosMTI/caracteristicas">
        <sch:name>MATERIAL CUERPO</sch:name>
        <sch:ProductModel>Basswood</sch:ProductModel>
        </rdf:Description>
      </sch:ProductModel>
    </rdf:Description>
  </sch:Product>
</rdf:Description>
</rdf:RDF>

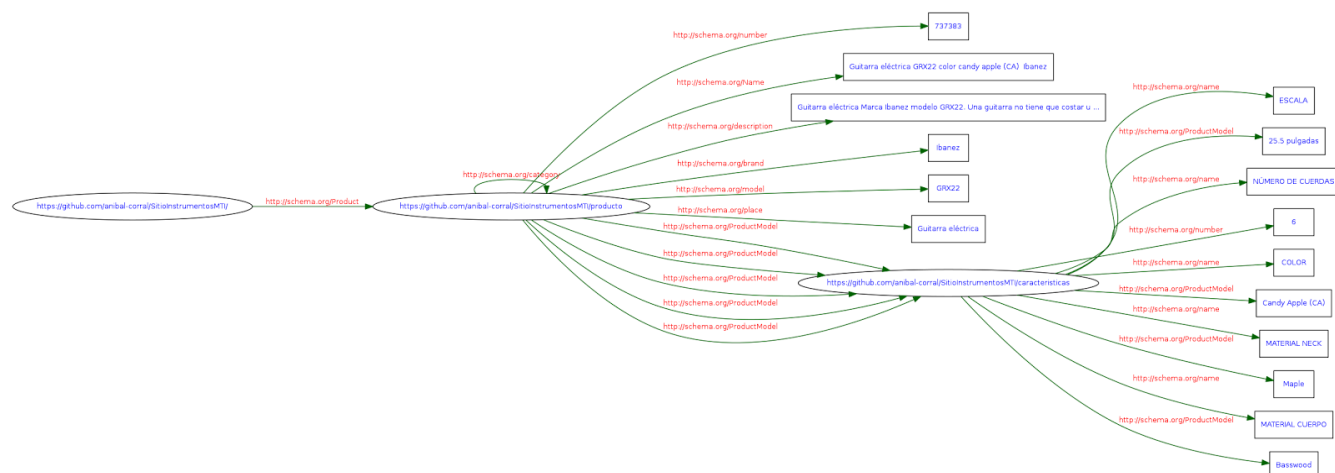
```


Luego se ha validado el archivo en <https://www.w3.org/RDF/Validator/>

Triples of the Data Model

Number	Subject	Predicate	Object
1	https://github.com/anibal-corral/SitioInstrumentosMTI/	http://schema.org/Product	https://github.com/anibal-corral/SitioInstrumentosMTI/producto
2	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/number	"737383"
3	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/name	"Guitarra eléctrica GRX22 color candy apple (CA) Ibanez"
4	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/description	"Guitarra eléctrica Marca Ibanez modelo GRX22. Una guitarra no tiene que costar una fortuna para tener un buen sonido, es el caso de la serie GRX de Ibanez, fue desarrollada para los músicos que desean la calidad de esta marca, pero a un precio asequible."
5	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/brand	"Ibanez"
6	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/model	"GRX22"
7	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/category	https://github.com/anibal-corral/SitioInstrumentosMTI/producto
8	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/place	"Guitarra eléctrica"
9	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/ProductModel	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas
10	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/name	"ESCALA"
11	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/ProductModel	"25.5 pulgadas"
12	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/ProductModel	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas
13	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/name	"NÚMERO DE CUERDAS"
14	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/number	"6"
15	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/ProductModel	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas
16	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/name	"COLOR"
17	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/ProductModel	"Candy Apple (CA)"
18	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/ProductModel	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas
19	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/name	"MATERIAL NECK"
20	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/ProductModel	"Maple"
21	https://github.com/anibal-corral/SitioInstrumentosMTI/producto	http://schema.org/ProductModel	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas
22	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/name	"MATERIAL CUERPO"
23	https://github.com/anibal-corral/SitioInstrumentosMTI/caracteristicas	http://schema.org/ProductModel	"Basswood"

y finalmente se muestra el grafo resultante del archivo rdf/turtle:



3.2. Incorporar Metadatos

Existen 2 tecnologías que permiten incorporar metadatos en páginas Web: [microdatos](#) y [RDFa](#). Enriquecer las páginas Web de establecimientos y productos musicales con microdatos ó con RDFa para que sus contenidos puedan ser indexados por buscadores. Utilizar para ello elementos del vocabulario [Schema.org](#).

Para el siguiente punto, se han utilizado los archivos instrumentos.html y establecimiento.html, archivos generados en el sitio web, para lo cual se utilizó el editor SUBLIME TEXT y la validación se hizo en el siguiente link: <https://www.w3.org/2012/pyRdfa/Validator.html>

3.2.1 Instrumentos:

- ❑ Link para adquirir código instrumentos.html original, para agregar metadatos.
<http://35.185.162.103/index.html#/instrumento>
- ❑ Link archivo rdfa.instrumentos.html:
<https://github.com/anibal-corral/SitiInstrumentosMTI/blob/master/entrega/parte%203/rdfa.instrumentos.html>

Código rdfa.instrumentos.html

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>

<div ng-view="" class="ng-scope"><div class="panel panel-default ng-scope">
<a href="#/addinstrumento" class="btn btn-default btn-block">
  Registrar nuevo instrumento
</a>
<div class="panel-heading">Listado de Instrumentos</div>
<!-- <button ng-click="obtenerInstrumentos()"> hola instrumentos</button> -->
  <table class="table">
    <tbody><tr><th></th>
      <th>Nombre</th>
      <th>Precio</th>
      <th>Descripción</th>
      <th>Marca</th>
      <th>Modelo</th>
      <th>Origen</th>
      <th>Establecimiento</th>
      <th>Acción</th>
    </tr>
    <!-- ngRepeat: i in instrumentos -->

      <tr ng-repeat="i in instrumentos" class="ng-scope" typeof="schema:Thing">
        <td class="ng-binding" property="schema:identifier">0 </td>
        <td class="ng-binding" property="schema:name">INSTRUMENTO
1</td>

        <td class="ng-binding" property="schema:number">20</td>
        <td class="ng-binding" property="schema:description">EL
PRIMER INSTRUMENTO DE PRUEBA</td>
        <td class="ng-binding" property="schema:Brand">MARCA
TEST</td>
        <td class="ng-binding" property="schema:ProductModel">MODEL
TEST</td>
        <td class="ng-binding" property="schema:Place">CHINO 1</td>
```

```

1</td>
                                <td class="ng-binding" property="schema:Place">Establecimiento
                                <td>
                                <button class="btn btn-danger btn-xs pull-right"
ng-click="retirar($index,i)" type="button">Retirar</button>
                                <a href="#/instrumento/595035365e59020010f2485e" class="btn
btn-default btn-xs pull-right" style="margin-right:10px;">Editar</a>
                                <!--
                                <a href="#/instrumento/{{ i._id }}" class="btn btn-default btn-xs
pull-right" style="margin-right:10px;">Editar</a>
                                -->
                                </td>

                                </tr><!-- end ngRepeat: i in instrumentos -->
                                </tbody></table>
</div></div>
</body>
</html>

```

Luego el código generado se valida en <https://www.w3.org/2012/pyRdfa/Validator.html> (RDFa 1.1 Validator), en donde nos genera el resultado de validación RFDa con vocabulario Shema.org

Documento Rdfa en formato turtle:

```

@prefix schema: <http://schema.org/> .

[] a schema:Thing;
  schema:Brand "MARCA TEST";
  schema:Place "CHINO 1",
    "Establecimiento 1";
  schema:ProductModel "MODEL TEST";
  schema:description "EL PRIMER INSTRUMENTO DE PRUEBA";
  schema:identifier "0 ";
  schema:name "INSTRUMENTO 1";
  schema:number "20" .

```

3.2.2 Establecimiento:

- ❑ Link para adquirir código establecimiento.html original, para agregar metadatos.
<http://35.185.162.103/index.html#/establecimiento>
- ❑ Link archivo rdfa.establecimiento.html:
<https://github.com/anibal-corral/SitiInstrumentosMTI/blob/master/entrega/parte%203/rdfa.establecimiento.html>

Código rdfa.establecimiento.html

```

</head>
<body>

<div ng-view="" class="ng-scope"><div class="panel panel-default ng-scope">
<a href="#/addestablecimiento" class="btn btn-default btn-block">
  Registrar nuevo establecimiento
</a>
<div class="panel-heading">Listado de Establecimientos</div>
  <table class="table">
    <tbody><tr>

      <th>Nombre</th>
      <th>Rut</th>
      <th>Ubicación</th>
      <th>Teléfono</th>
      <th>Horario</th>
      <th>Acción</th>
    </tr>

    <!-- ngRepeat: e in establecimientos -->

    <tr ng-repeat="e in establecimientos" class="ng-scope"
typeof="schema:Place">
      <td class="ng-binding" property="schema:name">Establecimiento 1</td>
      <td class="ng-binding" property="schema:number"> 1-9</td>
      <td class="ng-binding" property="schema:PostalAddress">Manchester</td>
      <td class="ng-binding" property="schema:number">+56985819635</td>
      <td class="ng-binding" property="schema:openingHours">Lunes a
Viernes</td>

      <td>
        <a href="#/establecimiento/595034f25e59020010f2485d" class="btn
btn-default btn-xs pull-right" style="margin-right:10px;">Editar</a>
      </td>
    </tr><!-- end ngRepeat: e in establecimientos -->

    <tr ng-repeat="e in establecimientos" class="ng-scope"
typeof="schema:Place">
      <td class="ng-binding" property="schema:name">Establecimiento 2</td>
      <td class="ng-binding" property="schema:number">654541126-k</td>
      <td class="ng-binding" property="schema:PostalAddress">Huerfanos

```



```

422</td>
        <td class="ng-binding" property="schema:number">+56437843443</td>
        <td class="ng-binding" property="schema:openingHours">9:00 - 18:30</td>

        <td>
        <a href="#/establecimiento/595b11c28a9fc50010f032e6" class="btn
btn-default btn-xs pull-right" style="margin-right:10px;">Editar</a>
        </td>
        </tr><!-- end ngRepeat: e in establecimientos -->

        <tr ng-repeat="e in establecimientos" class="ng-scope"
typeof="schema:Place">
        <td class="ng-binding" property="schema:name">Establecimiento 3</td>
        <td class="ng-binding" property="schema:number">74545854-5</td>
        <td class="ng-binding" property="schema:PostalAddress">alameda
623</td>
        <td class="ng-binding" property="schema:number">+56 3332333</td>
        <td class="ng-binding" property="schema:openingHours">9:00 - 18:30</td>

        <td>
        <a href="#/establecimiento/595b11d88a9fc50010f032e7" class="btn
btn-default btn-xs pull-right" style="margin-right:10px;">Editar</a>
        </td>
        </tr><!-- end ngRepeat: e in establecimientos -->

    </tbody></table>

</div></div>

</body>
</html>

```

Luego el código generado se valida en <https://www.w3.org/2012/pyRdfa/Validator.html>, (RDFa 1.1 Validator) en donde nos genera el resultado de validación RFDa con vocabulario Shema.org ,

Documento Rdfa en formato turtle:

```

@prefix schema: <http://schema.org/> .

[] a schema:Place;
  schema:PostalAddress "Huerfanos 422";
  schema:name "Establecimiento 2";
  schema:number "+56437843443",
    "654541126-k";
  schema:openingHours "9:00 - 18:30" .

[] a schema:Place;
  schema:PostalAddress "Manchester";
  schema:name "Establecimiento 1";
  schema:number "1-9",
    "+56985819635";
  schema:openingHours "Lunes a Viernes" .

[] a schema:Place;
  schema:PostalAddress "alameda 623";
  schema:name "Establecimiento 3";
  schema:number "+56 3332333",
    "74545854-5";
  schema:openingHours "9:00 - 18:30" .

```

3.3. Ventajas de agregar RDFa o microdatos

Analizar la solución presentada, así como las ventajas e inconvenientes de enriquecer portales Web con tecnologías semánticas. Este análisis puede llevarse a cabo utilizando uno o más casos de uso propuestos por los estudiantes. Se valorarán casos de uso reales o en los que los estudiantes tengan experiencia previa.

En relación a las ventajas, cabe destacar que debido a la inmensa cantidad de recursos disponibles actualmente en Internet, es necesario incorporar elementos que ayuden a mejorar los resultados obtenidos en las búsquedas de información. Para esto, una forma es incorporar RDFa o microdatos a las aplicaciones web, con el objetivo de mejorar la interpretación de la información de parte de las máquinas al aplicar etiquetas en los datos que ayuden a desambiguar el sentido de la información relacionada a un determinado elemento, además, son indispensables para la preservación de los recursos de información. Finalmente, al agregar microdatos o RDFa a los recursos se añade valor, que permiten el intercambio de información sin afectar a los recursos.

Puntualizando, las ventajas de la web semántica son principalmente:

- ☐ Incorporación de contenido semántico a los sitios web, lo que permite una mejor organización de la información y asegura búsquedas mucho más precisas por significado y no sólo por el contenido textual de los recursos.
- ☐ Permite a las computadoras y robots gestionar el conocimiento, algo que hasta antes de la web semántica, era una tarea reservada sólo a las personas [11].

En cuanto a las desventajas, destacan los problemas de estandarización de la semántica de contenidos web, ya que actualmente la iniciativa schema.org (de los principales buscadores de la web: Google, Bing, Yahoo y Yandex) es el mayor intento de estandarización, sin embargo no es un estándar oficial y único de uso para este tipo de tecnologías. Otras desventajas son:

- ☐ El costo y gran trabajo que implica adaptar los documentos para poder ser procesados de forma semántica.
- ☐ Problemas con los diversos idiomas.
- ☐ La dificultad para lograr una unificación de los estándares semánticos y proveer relaciones de equivalencia entre conceptos [11].

Sin embargo, la web semántica permite moverse de una web basada en documentos a una red de datos enlazados que permite ofrecer diversas funcionalidades en base al potencial presente en combinar e interrelacionar datos de diversa índole y procedencia, permitiendo que ya no sólo los humanos puedan procesar información por su significado, sino también las máquinas, realizando procesamiento de datos de forma automatizada [12].

Por ejemplo un caso de uso con Rdfa, al buscar la palabra “zapato”, se muestra el sitio “www.falabella.com” como primera opción, siendo una de las ventajas de usar vocabulario shema (tiene varias clases y propiedades predefinidas que permiten definir dicho vocabulario), lo cual ayuda a que los sitios tengan la opción de posicionarse de los primeros, pudiendo lograr mejores resultados de búsqueda.



zapato mujer



[Todo](#) [Imágenes](#) [Noticias](#) [Videos](#) [Maps](#) [Más](#) [Preferencias](#) [Herramientas](#)

Cerca de 4.950.000 resultados (0,48 segundos)

- ✓ **Si Tienes Zapatos En Mente - Encuéntralos En Falabella.com**
[Anuncio](#) www.falabella.com/ ▼
Disfruta Más De 200 Marcas y 7.000 Productos Solo En Falabella.com
Marcas: CONVERSE, DC, JACK & JONES@...
Has visitado falabella.com dos veces. Última visita: hoy
[Zapatos de Mujer](#) · [Moda de Mujer](#) · [Botines](#) · [Botas de Mujer](#) · [Zapatos de Fiesta](#) · [Sandalias](#)
- ✓ **Zapatos de Mujer - kliper.cl**
[Anuncio](#) www.kliper.cl/ ▼
Compra las mejores marcas sólo en Kliper.cl. Conócenos aquí.
[Envío a Todo Chile](#) · [Envío Gratis sobre \\$50.000](#) · [Cambios Sin Costo](#) · [6 cuotas Sin Interés](#).
[Calzado Mujer](#) · [Vestuario Mujer](#) · [Accesorios Mujer](#) · [Descuentos hasta 50%](#)
- ✓ **Zapato Mujer - Dafiti.cl**
[Anuncio](#) www.zapatos.dafiti.cl/ ▼ (2) 2656 9840
¡La Mayor Variedad de Zapatos Aquí! Envío Gratis sobre \$50.000.
[Devolución Gratis y Fácil](#) · [Ganador eCommerce Award](#) · [Site Confiable](#) - E-bit
Tipos: [Zapatillas](#), [Botas](#), [Botines](#), [Alpargatas](#)
[Póleras 4 x \\$20.990](#) · [Botines 40% Off 2a Unidad](#) · [Zapatos 40% Off 2a Unidad](#)
- ✓ **Zapato Mujer - shopcaterpillar.cl**
www.shopcaterpillar.cl/zapato-mujer ▼
[Zapatos CL](#) · [Login / Registro](#) · [0 Ver Carro 0](#) · [Tu Carro está vacío! 0 Ver Carro 0](#) ... [Productos de limpieza](#) · [Sale](#) · [Inicio](#) » [Zapato Mujer](#). [Zapato Mujer](#). [Mostrar: 12](#)

Parte del código html del sitio que contiene metadatos:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:fb="http://www.facebook.com/2008/fbml"
  xmlns:og="http://ogp.me/ns#">
<script language="JavaScript" type="text/javascript"
  src="/static/script/common/jquery-1.7.2.min.js?vid=1904"></script>
<script language="JavaScript" type="text/javascript"
  src="/static/script/common/jquery.colorbox-min.js?vid=1904"></script>
<script type="text/javascript">
  globalFMTMessages = {
    'addToCart.error' : 'Lo sentimos, se ha producido un error mientras tratabas de
agregar un producto a tu bolsa. Por favor, comunícate con nuestro Servicio al Cliente al
223907910 o escríbenos a contacto@falabella.cl',
    'addItemToCart.hasOtherSkutypelnCart' : 'Tiene un tipo sku otros en el
carrito.',
    'installment.error' : 'En este momento no es posible calcular el valor de tu
cuota CMR. Por favor, inténtalo más tarde.',
    'order.commit.cancel.outofstock.oneofmanyproducts' : 'Algunos productos de
tu compra ya no están disponibles, por favor selecciona una de las siguientes alternativas.',
    'order.commit.quota.overBurned' : 'La fecha de entrega que escogiste ya no se
encuentra disponible.Te rediregiremos a la Bolsa de Compras para que puedas elegir una
nueva fecha de entrega.[Ir a Bolsa de Compras]/falabella-cl/cart/cart.jsp'
  }
</script>
<head>

<script type="text/javascript" src="/static/script/common/adrum.js?vid=1904"></script>
<meta http-equiv="X-UA-Compatible" content="IE=Edge">

<meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type" />
<link rel="alternate" hreflang="x-default"
  href="http://www.falabella.com/" />
<link rel="alternate" href="http://www.falabella.com/falabella-cl/"
  hreflang="es-cl" />
<link rel="alternate" href="http://www.falabella.com.ar/falabella-ar/"
  hreflang="es-ar" />
<link rel="alternate" href="http://www.falabella.com.pe/falabella-pe/"
  hreflang="es-pe" />
<link rel="alternate" href="http://www.falabella.com.co/falabella-co/"
  hreflang="es-co" />
<script
src="//assets.adobedtm.com/6fa5b5cd2babc0bd10785a36afb0992edfac60c2/satelliteLib-be88
883ee5e0094ffd47a4acb38dd2a6669499e8.js"></script><meta content="text/html;
charset=iso-8859-1" http-equiv="Content-Type" />
<title>Zapatos - Falabella.com</title>
<meta content="" name="Keywords" />
<meta content="" name="Description" />
<meta name="language" content="ES">
<meta name="og:country-name" content="CL" />

<meta property="og:type" content="website" />
<meta property="og:title" content="Zapatos - Falabella.com" />
```



```
<meta property="og:image"
content="http://www.falabella.com/static/site/common/Falabella-Chile.jpg" />
<meta property="og:description" content="Zapatos - Falabella.com" />

<meta property="og:url"
  content="http://www.falabella.com/falabella-cl/category/cat700256/Zapatos" />

<meta property="og:site_name" content="Falabella.com" />
  <meta http-equiv="cache-control" content="max-age=600"/>
    <meta http-equiv="expires" content="never"/>
      <meta http-equiv="pragma" content="cache"/>
<link href="/static/styles/reset.css?vid=1904" rel="stylesheet" type="text/css" />
<link href="/static/styles/plugins.css?vid=1904" rel="stylesheet" type="text/css" />
<link href="/static/styles/base_miscellaneous.css?vid=1904" rel="stylesheet"
type="text/css" />
<link href="/static/styles/base_miscellaneous-R0Extra.css?vid=1904" rel="stylesheet"
type="text/css" />
<!--[if !IE]><!-->
```

4. Bibliografía

4.1. Referencias

- [1] Van Der Vlist, E.. "JSON and XML: a new perspective". In *XML London 2014 Conference Proceedings*. 2014. Disponible en Internet: <http://xmllondon.com/2014/xmllondon-2014-proceedings.pdf#page=157>
- [2] Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C.. "Comparison of JSON and XML Data Interchange Formats: A Case Study". *Scenario*. 2009, Vol. 59715, pp. 1-3. Disponible en Internet: <http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf>
- [3] w3schools. "JSON vs XML". 2017. Disponible en Internet: https://www.w3schools.com/js/js_json_xml.asp
- [4] Tinney, Steve. "XMD: XML Metadata". *Oracc*. 2014. Disponible en Internet: <http://oracc.museum.upenn.edu/ns/xmd/1.0/>
- [5] Shelat, Mihir. "Node.js for Enterprise applications!! Are you kidding?". 2016. Disponible en Internet: <https://www.linkedin.com/pulse/nodejs-enterprise-applications-you-kidding-mihir-shelat>
- [6] Habib, Omel. "A practical guide to popular Node.js MVC frameworks". *APP Dynamics*, Cisco. 2015. Disponible en Internet: <https://blog.appdynamics.com/engineering/a-practical-guide-to-popular-node-js-mvc-frameworks/>
- [7] Habib, Omel. "Battle of the PaaS Node.js apps in the Cloud". *APP Dynamics*, Cisco. 2016. Disponible en Internet: <https://blog.appdynamics.com/product/battle-of-the-paas-node-js-apps-in-the-cloud/>
- [8] Noders. Comunidad global alrededor de JavaScript, NodeJS y su ecosistema en Latinoamérica y el Mundo. Disponible en Internet: <http://noders.com>
- [9] Travis, Greg. "Getting started with new I/O (NIO)". *IBM Java development*. 2003. Disponible en internet: <https://www.ibm.com/developerworks/java/tutorials/j-nio/j-nio.html>
- [10] Best Web Frameworks. Disponible en Internet: <http://www.bestwebframeworks.com/>
- [11] Salazar Argonza, Javier. "Estado actual de la Web 3.0 o Web Semánticas". *Revista Digital Universitaria* [en línea]. 1 de noviembre de 2011, Vol. 12, No. 11. Disponible en Internet: <http://www.revista.unam.mx/vol.12/num11/art108/#a>
- [12] Shorlemmer, M. "Diez años construyendo una web semántica". *Lychnos: Cuadernos de La Fundación General CSIC* [en línea]. 2011, Vol. 7, pp. 26-32. Disponible en Internet: http://www.fgcsic.es/lychnos/es_es/articulos/construyendo_una_web_semantica

4.2. Fuentes Consultadas

- [1] Expertos en Servicios de Consultoría Exes. "Manual de XML". Disponible en Internet: <http://www.mundolinux.info/herramientas-xml.htm>
- [2] w3school.com. "DTD Tutorial". Disponible en Internet: http://www.w3schools.com/xml/xml_dtd_intro.asp
- [3] Nogales, Tomás. "Editores XML". 2013. Disponible en Internet: <http://www.bib.uc3m.es/~nogales/cursos/editxml.html>
- [4] Países en XML. Disponible en Internet: <http://www.asiutn.com.ar/sysacad/paises.xml>
- [5] Ramírez, Manuel. "Seguridad en Servicios Web REST". *Information System Security Association*. 2015. Disponible en Internet: http://www.issachile.cl/Segurida_API_REST
- [6] Conde, Jesús. "Qué es REST y RESTful". *Serie ¿Qué es?*. 2015. Material videográfico disponible en Internet: <https://www.youtube.com/watch?v=pVAMOielOJQ>
- [7] Desarrollo web.com. "Crear una API REST en 5 minutos con json-server" *DesarrolloWeb.com*. 2015. Material videográfico disponible en Internet: <https://www.youtube.com/watch?v=Z6FH1vGn94k>
- [8] Sarmiento, KR. "¿Qué es RDF y para qué es bueno?". *Semantizando la Web*. 2011. Disponible en Internet: <https://semantizandolaweb.wordpress.com/2011/11/07/que-es-rdf-y-para-que-es-bueno/>