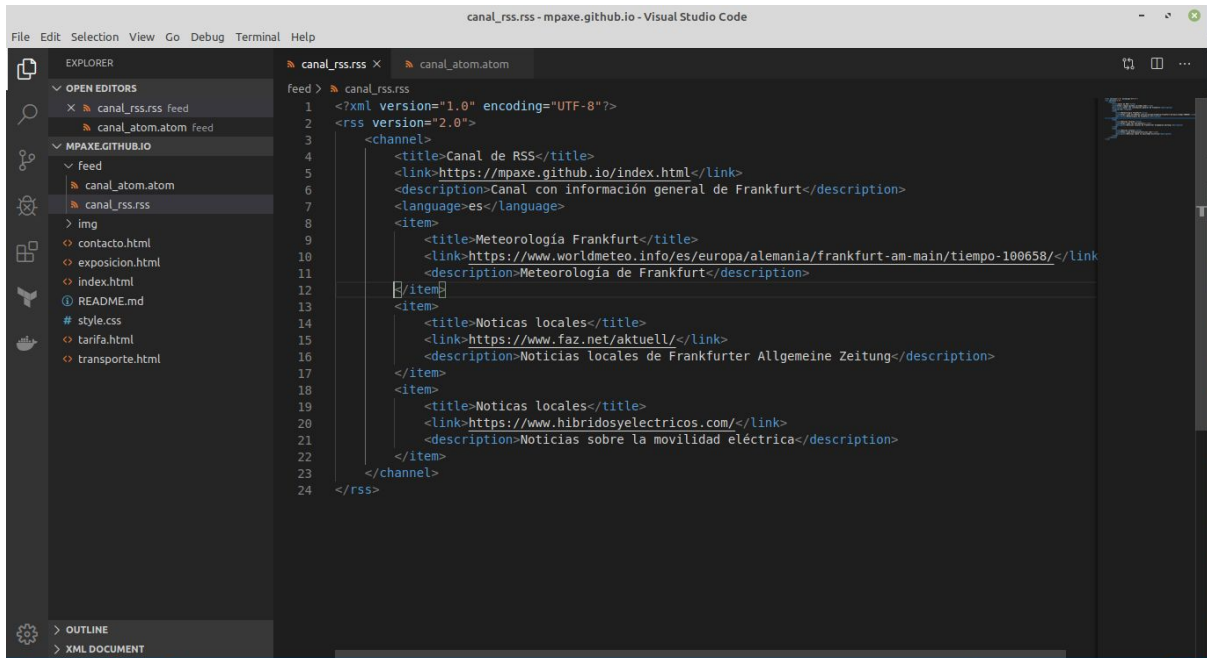


# Tarea 03 de LMSGI

## RSS

1. Creamos los ficheros .rss, yo he usado como en la actividad anterior la herramienta Visual Studio Code



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rss version="2.0">
3   <channel>
4     <title>Canal de RSS</title>
5     <link>https://mpaxe.github.io/index.html</link>
6     <description>Canal con información general de Frankfurt</description>
7     <language>es</language>
8     <item>
9       <title>Meteorología Frankfurt</title>
10      <link>https://www.worldmeteo.info/es/europa/alemania/frankfurt-am-main/tiempo-100658/</link>
11      <description>Meteorología de Frankfurt</description>
12    </item>
13    <item>
14      <title>Noticias locales</title>
15      <link>https://www.faz.net/aktuell/</link>
16      <description>Noticias locales de Frankfurter Allgemeine Zeitung</description>
17    </item>
18    <item>
19      <title>Noticias locales</title>
20      <link>https://www.hibridosyelectricos.com/</link>
21      <description>Noticias sobre la movilidad eléctrica</description>
22    </item>
23   </channel>
24 </rss>
```

2. Una vez creados el archivo vamos definiendo sus etiquetas, lo primero es la versión del rss, luego creamos un canal al que le ponemos como mínimo las etiquetas <title>, <link> y <description>, la de <language> es opcional. Por último la etiqueta <item> que donde irán las diferentes secciones.
3. dentro de las etiquetas de <item> van otro elementos que son <title>, <link> y <description>, sí lo mismos que usamos para definir el canal, pero esta vez lo usamos para los items.  
Tendremos que usar tantos items como secciones queramos mostrar. En este caso son tres.
4. Una vez terminado lo validamos, podríamos haberlo hecho de muchas formas, yo he elegido validarlo en [https://validator.w3.org/feed/#validate\\_by\\_input](https://validator.w3.org/feed/#validate_by_input) directamente, pero podría valer de otra forma, pero teniendo internet me parece la forma más rápida y fácil de hacerlo.  
Nos dan algunas recomendaciones, como incluir un elemento <guid> en los elementos <item> pero esa etiqueta ni siquiera viene en el temario, y como no es obligatoria vamos a obviar la.



## Feed Validation Service

Check the syntax of Atom or RSS feeds

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Canal de RSS</title>
    <link>https://mpaxe.github.io/index.html</link>
    <description>Canal con información general de
Frankfurt</description>
    <language>es</language>
    <item>
      <title>Meteorología Frankfurt</title>
      <link>https://www.worldmeteo.info/es/europa/alemania/fr
ankfurt-am-main/tiempo-100658</link>
      <description>Meteorología de
Frankfurt</description>
    </item>
  </channel>
</rss>
```

Validate

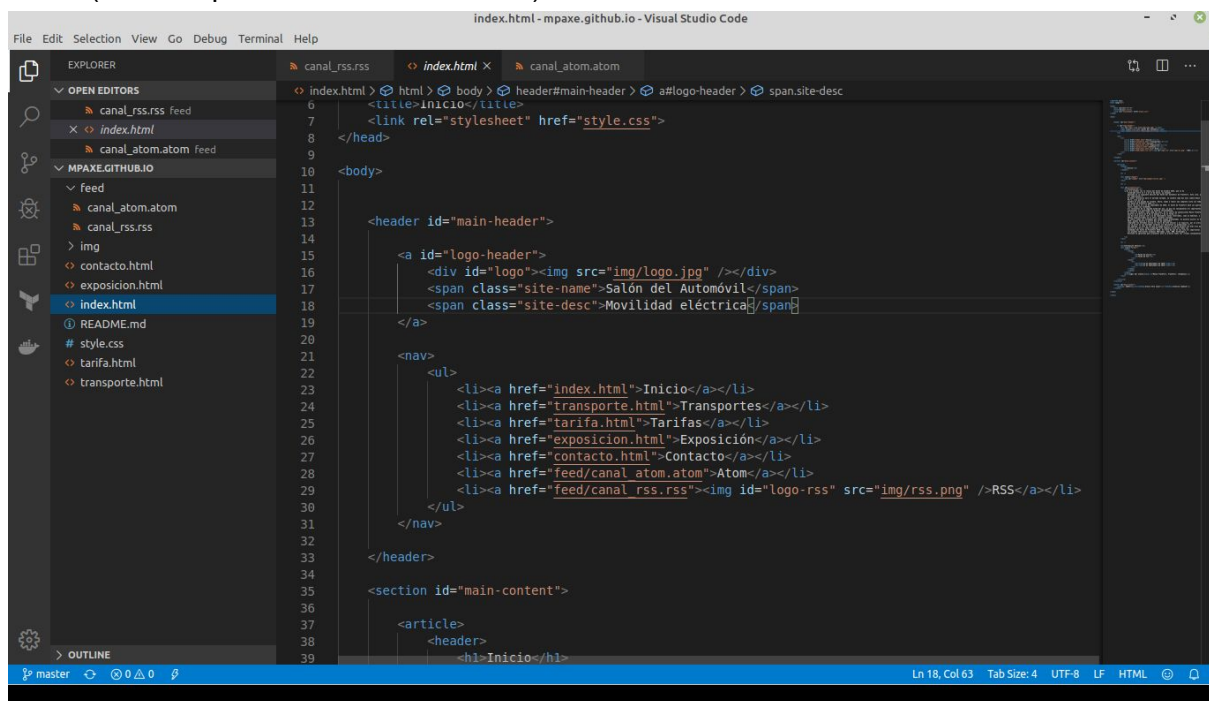
Congratulations!



This is a valid RSS feed.

### Recommendations

- Una vez validado el archivo lo incluimos en su carpeta correspondiente en el proyecto, normalmente feed pero podía ser cualquier otra, y añadimos el enlace a nuestro canal rss (esto tampoco viene en el temario).



- Por último es poner nuestra web en un servidor web, aquí también tenemos varias opciones, yo he optado por usar la página que nos provee GitHub y otra web que nos da un servidor web gratuito para páginas estáticas <https://infinityfree.net/>, estas son la direcciones:

GitHub: <https://mpaxe.github.io/index.html>

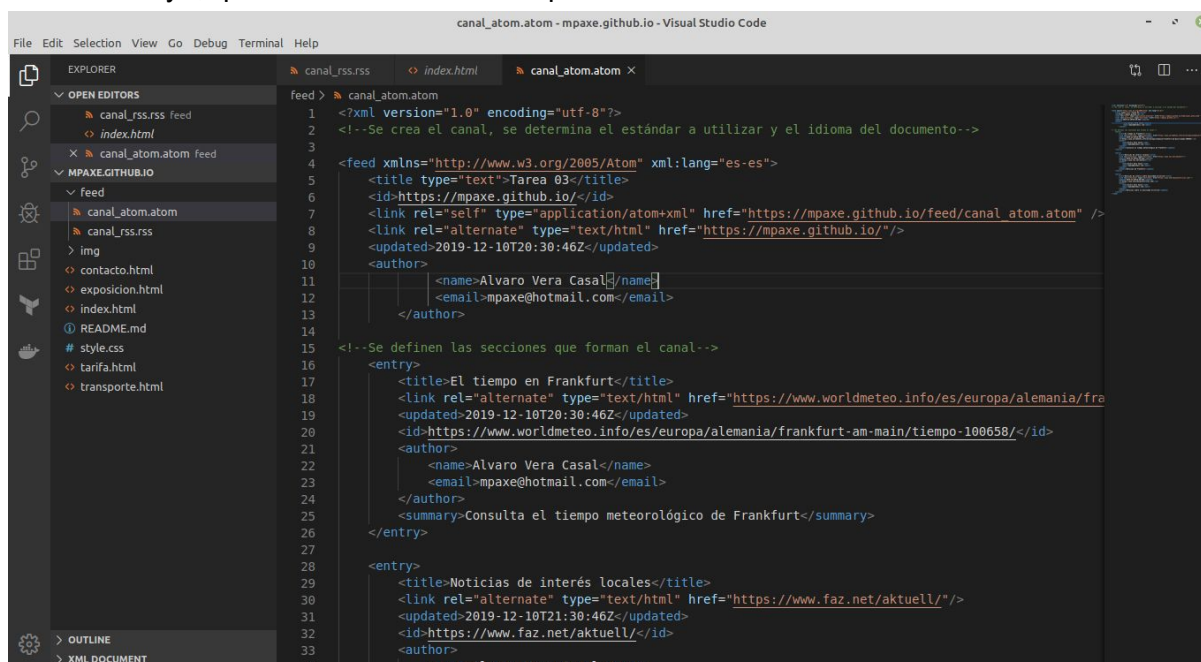
InfinityFree: <http://mpaxe.epizy.com/mpaxe.github.io/index.html>

# ATOM

1. Para el estándar de atom el proceso es muy similar, lo único que varía es la estructura del fichero y como no me parece necesario comentar todos los pasos anteriores ya que son idénticos solo vamos a centrarnos en la estructura del fichero.

Las etiquetas necesarias son <feed> donde definimos el estándar y en su interior van <title>, <id>, <link>, <updated>, <author>, y <entry> que viene a ser la etiqueta de <item> en rss.

Dentro de los entrys no pasa igual que en anterior estándar (rss) que tenemos las mismas etiquetas que fuera de esta, solo que esta vez añadimos una nueva que es <summary>, que es donde va la descripción.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--Se crea el canal, se determina el estándar a utilizar y el idioma del documento-->
3
4 <feed xmlns="http://www.w3.org/2005/Atom" xml:lang="es-es">
5   <title type="text">Tarea 03</title>
6   <id>https://mpaxe.github.io/</id>
7   <link rel="self" type="application/atom+xml" href="https://mpaxe.github.io/feed/canal_atom.atom" />
8   <link rel="alternate" type="text/html" href="https://mpaxe.github.io/" />
9   <updated>2019-12-10T20:30:46Z</updated>
10  <author>
11    <name>Alvaro Vera Casal</name>
12    <email>mpaxe@hotmail.com</email>
13  </author>
14
15  <!--Se definen las secciones que forman el canal-->
16  <entry>
17    <title>El tiempo en Frankfurt</title>
18    <link rel="alternate" type="text/html" href="https://www.worldmeteo.info/es/europa/alemania/fra
19    <updated>2019-12-10T20:30:46Z</updated>
20    <id>https://www.worldmeteo.info/es/europa/alemania/frankfurt-am-main/tiempo-100658/</id>
21    <author>
22      <name>Alvaro Vera Casal</name>
23      <email>mpaxe@hotmail.com</email>
24    </author>
25    <summary>Consulta el tiempo meteorológico de Frankfurt</summary>
26  </entry>
27
28  <entry>
29    <title>Noticias de interés locales</title>
30    <link rel="alternate" type="text/html" href="https://www.faz.net/aktuell/" />
31    <updated>2019-12-10T21:30:46Z</updated>
32    <id>https://www.faz.net/aktuell/</id>
33    <author>
34      <name>Alvaro Vera Casal</name>
```

2. Luego seguimos los paso que realizamos en el ejercicio anterior, validamos y colocamos el enlace y listo!!

