María Fernández Rojo

uo263595@uniovi.es

Bitácora

En esta bitácora se recogen todos los sucesos relacionados con el desarrollo del proyecto correspondiente a la convocatoria extraordinaria de Software y Estándares para la Web.

software y estándares para la web

Convocatoria Extraordinaria

**Contenido**

[Creación del proyecto 3](#_Toc75102203)

[Creación de un repositorio en GitHub 3](#_Toc75102204)

[Contenido HTML5 (no generado) 3](#_Toc75102205)

[Estructura HTML5 (no generado) 4](#_Toc75102206)

[Fichero peliculas.xml 4](#_Toc75102207)

[Documento XML bien formado 4](#_Toc75102208)

[Google Chrome 5](#_Toc75102209)

[Mozilla Firefox 5](#_Toc75102210)

[Microsoft Edge 5](#_Toc75102211)

[Opera 6](#_Toc75102212)

[Validación con DTD 6](#_Toc75102213)

[Notepad++ 6](#_Toc75102214)

[Code Beauty 6](#_Toc75102215)

[Validación con XML Schema 7](#_Toc75102216)

[Notepad++ 7](#_Toc75102217)

[Utilities-Online 7](#_Toc75102218)

[CoreFiling 7](#_Toc75102219)

[FreeFormatter 8](#_Toc75102220)

[Procesamiento de XML 8](#_Toc75102221)

[Problema con enlaces 8](#_Toc75102222)

[Descripción alternativa de imágenes 8](#_Toc75102223)

[API de Mediastack 9](#_Toc75102224)

[API de Marketstack 9](#_Toc75102225)

[Problema con las APIs 9](#_Toc75102226)

[Intento de solución 10](#_Toc75102227)

[Solución final 10](#_Toc75102228)

[Advertencia contenido mixto 10](#_Toc75102229)

[API de Weatherbit 11](#_Toc75102230)

[Descargar cartelera 11](#_Toc75102231)

[API File de HTML5 12](#_Toc75102232)

[Modificaciones base de datos 12](#_Toc75102233)

[Carga de datos 13](#_Toc75102234)

[Diseño del apartado de información deportiva 13](#_Toc75102235)

[Búsqueda sin resultados 14](#_Toc75102236)

[Búsqueda parcial 14](#_Toc75102237)

[Todos los resultados 14](#_Toc75102238)

[Mensaje de error en las APIs 15](#_Toc75102239)

[Funcionamiento en máquina local 15](#_Toc75102240)

**[01/06/2021 - 9:00]**

## Creación del proyecto

Inicialmente empecé el proyecto con una estructura básica de HTML, dividiendo el acceso a las distintas páginas a través de un navegador. También añadí un fichero CSS para darle estilo a las páginas.

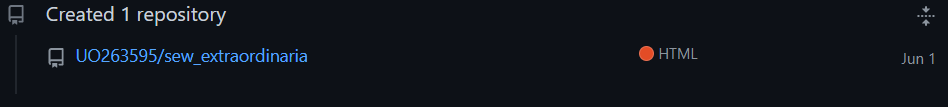


navegador inicial

**[01/06/2021 - 10:20]**

## Creación de un repositorio en GitHub

Para la visualización general de la página web cree un repositorio que fui actualizando a lo largo del desarrollo de la misma con las nuevas ampliaciones.



creación del repositorio

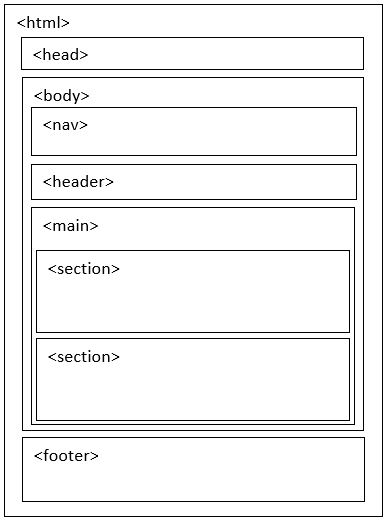
**[01/06/2021 - 10:25]**

## Contenido HTML5 (no generado)

La página web contiene tres documentos de HTML5 no generados, con el siguiente contenido: en el primero vemos una explicación sobre la página y algo de información relacionada con los periódicos digitales (el contexto y sus ventajas) además de un vídeo explicativo de la digitalización de los medios; en el segundo se encuentran las noticias más relevantes del 2021 y un audio resumen de las noticias del 2020, cada una de estas noticias ocupa una sección; el último de estos documentos contiene información sobre el equipo que hipotéticamente desarrolla este periódico, en él se habla del equipo directivo y el personal que compone los distintos ámbitos (política, internacional y desalambre, edición, portada, redes y boletines y economía), además como este periódico también tiene ediciones asociadas que se corresponden con diferentes provincias españolas, se introduce una tabla con los directores y el personal del equipo de redacción de cada una de ellas.

**[01/06/2021 - 10:40]**

## Estructura HTML5 (no generado)

Tal como se indicó en la propuesta el esquema general que siguen todos estos documentos es la que se puede observar en la imagen.

Cada uno de ellos contendrá obligatoriamente estos elementos, siendo de especial importancia la barra de navegación, que permitirá moverse entre los distintos documentos, también el footer va a contener enlaces a otras páginas, que se implementarán a continuación.

esquema general

**[]**

## Fichero peliculas.xml

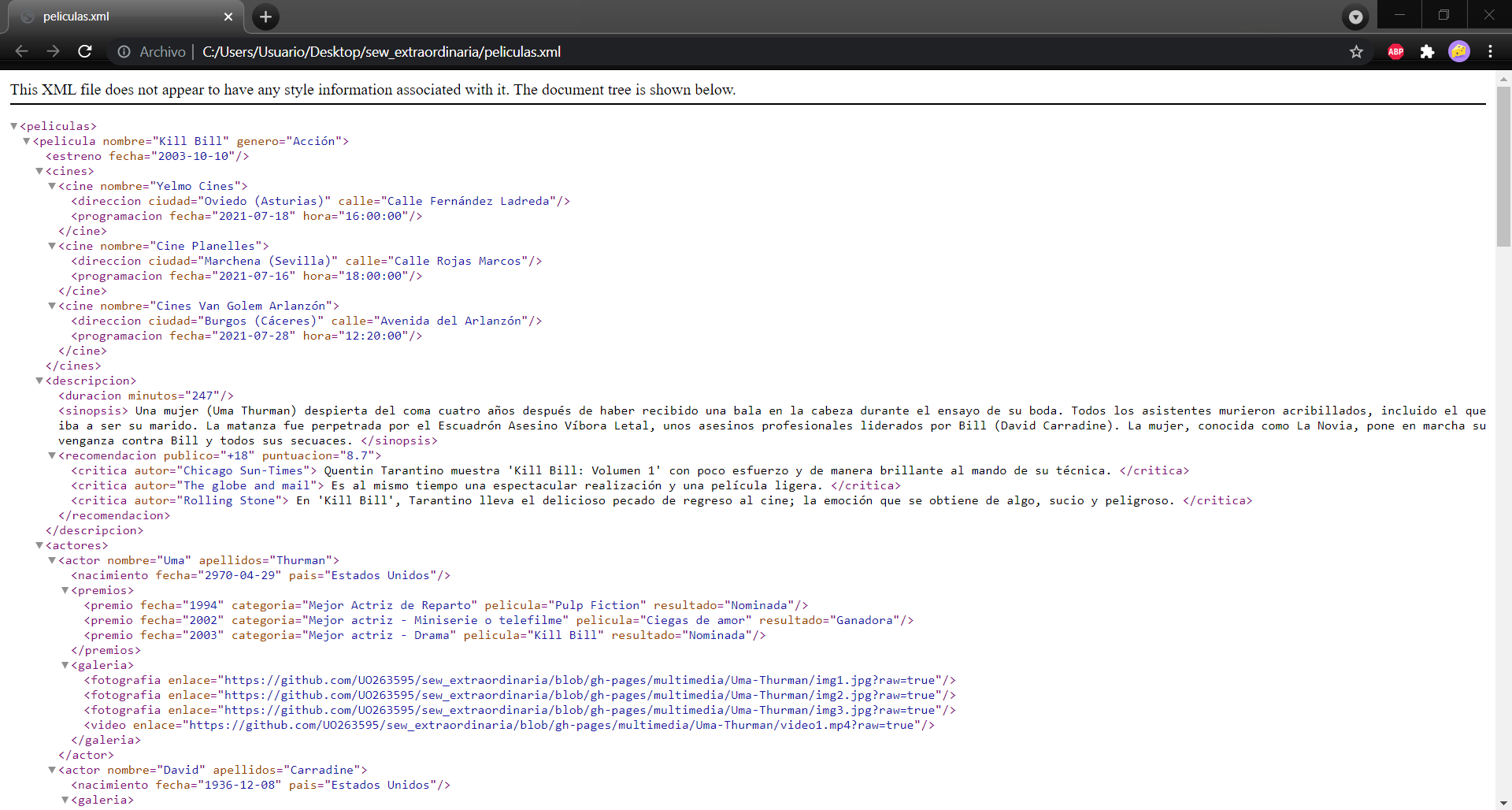
Con el fin de simular una cartelera de cine, este fichero contiene tres películas con información de cada una de ellas, tal como su nombre, género, fecha de estreno, sinopsis, etc. También se pueden ver las críticas que ha obtenido dicha película con los correspondientes autores de cada una de ellas y algo de información de los actores y actrices principales, como su nombre y apellidos, fecha de nacimiento y país, si tienen algún premio también aparecerá aquí, por último, podemos visualizar una galería con fotografías (mínimo una y máximo cinco) y opcionalmente vídeos (máximo tres) de cada uno de ellos. Finalmente está la bibliografía con las páginas web en las que se han consultado los datos.

**[]**

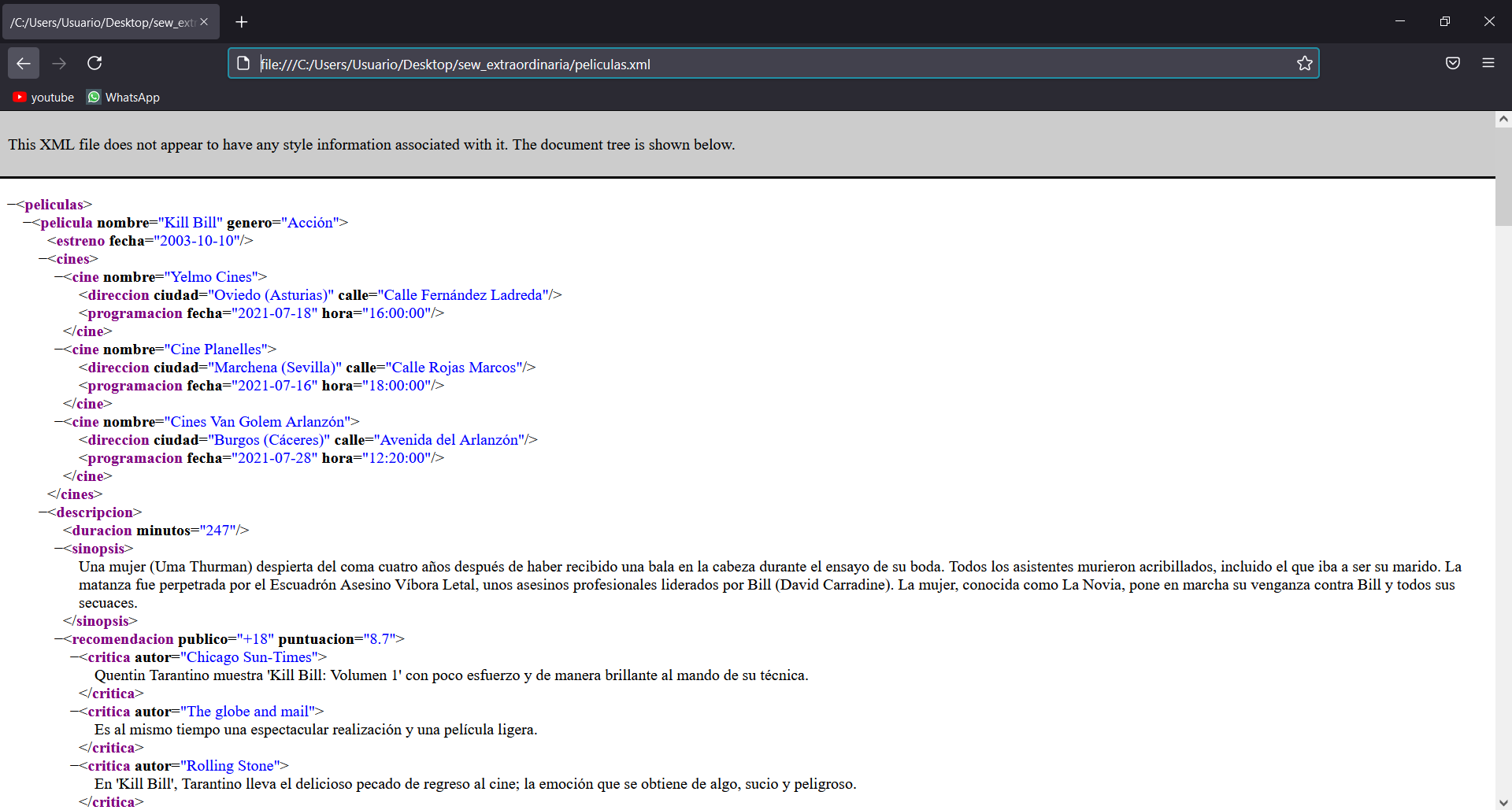
## Documento XML bien formado

Compruebo en los cuatro navegadores de referencia si el documento peliculas.xml que he creado es bien formado.

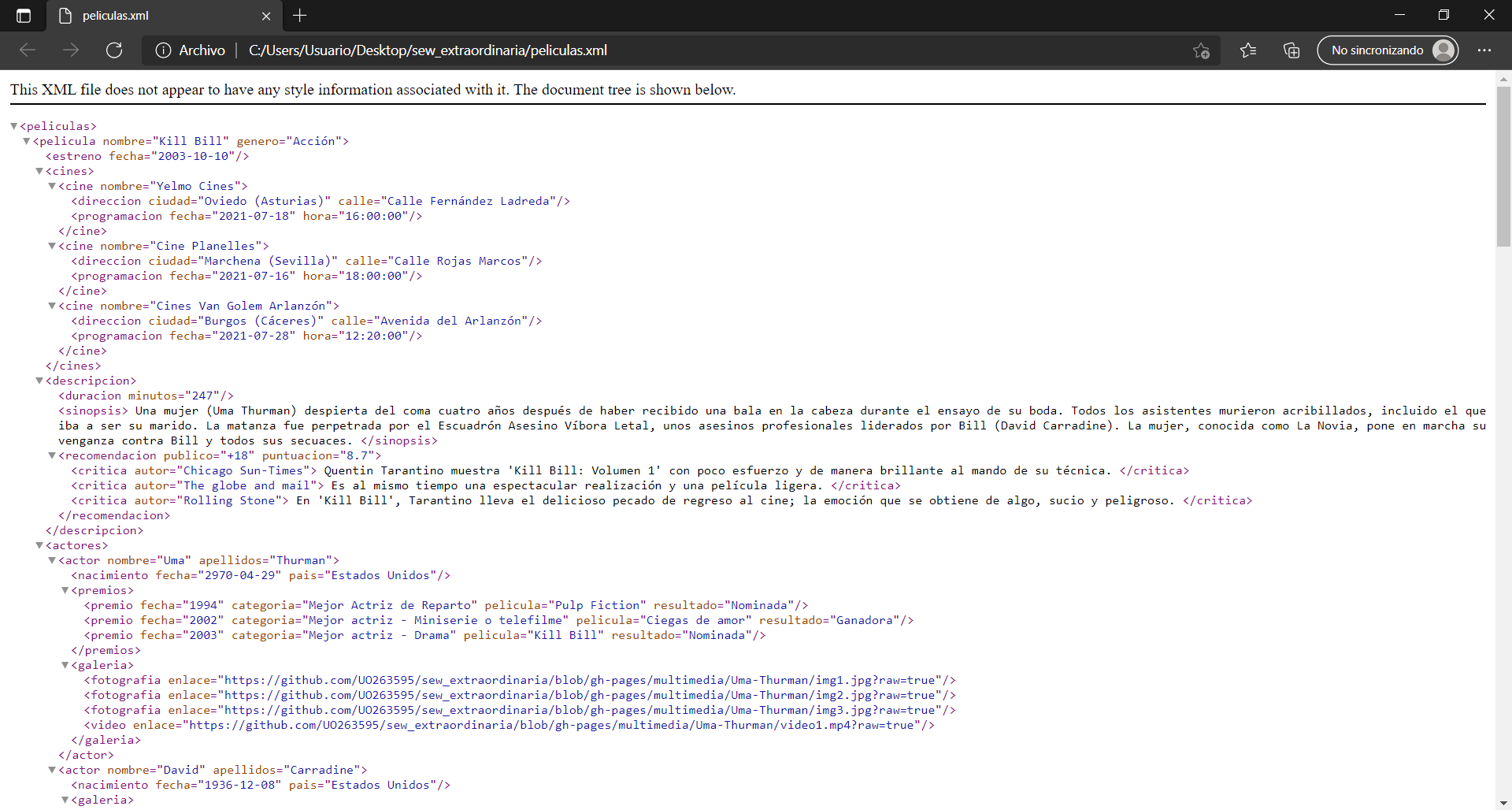
### Google Chrome



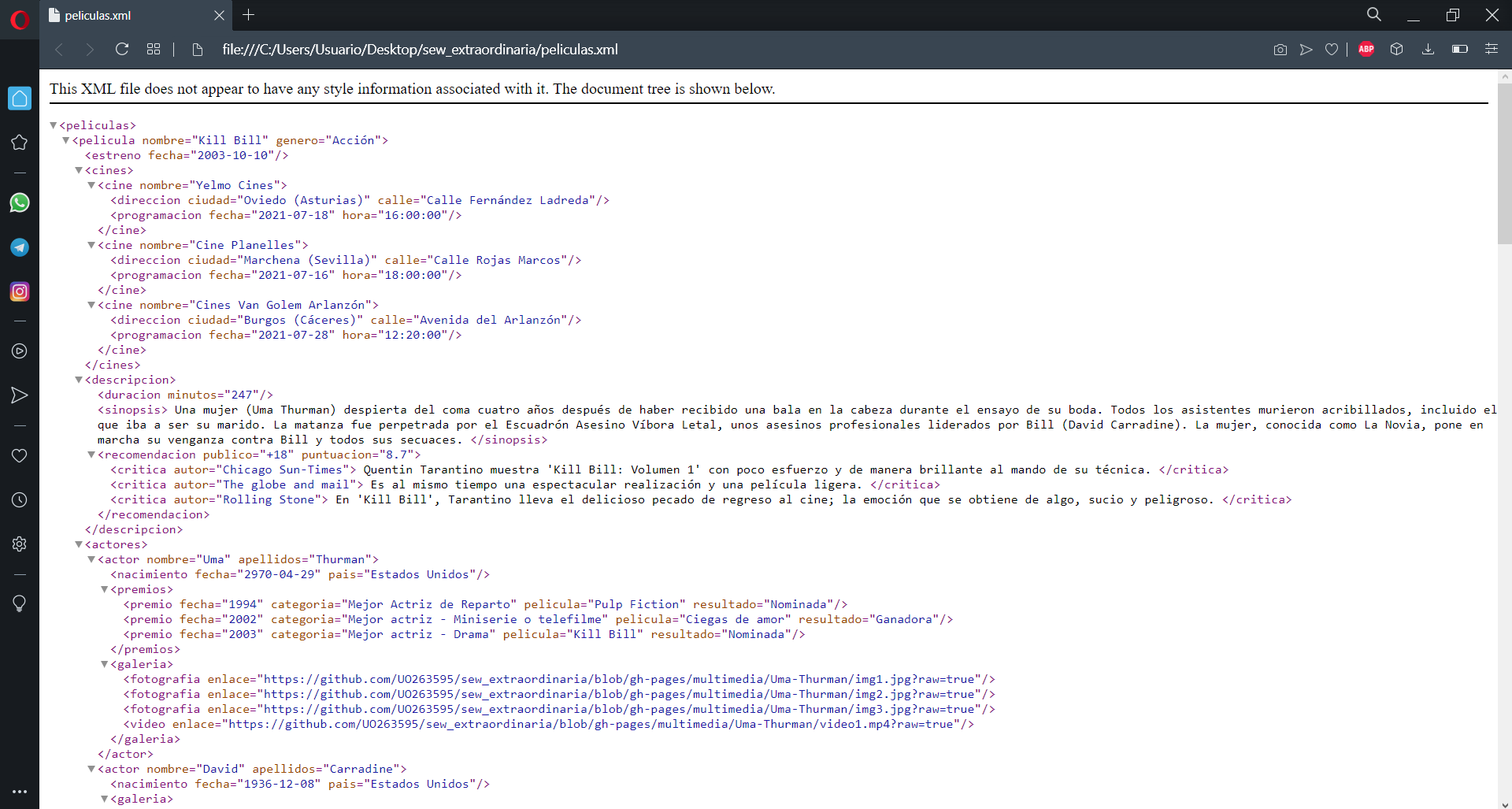
### Mozilla Firefox



### Microsoft Edge



### Opera

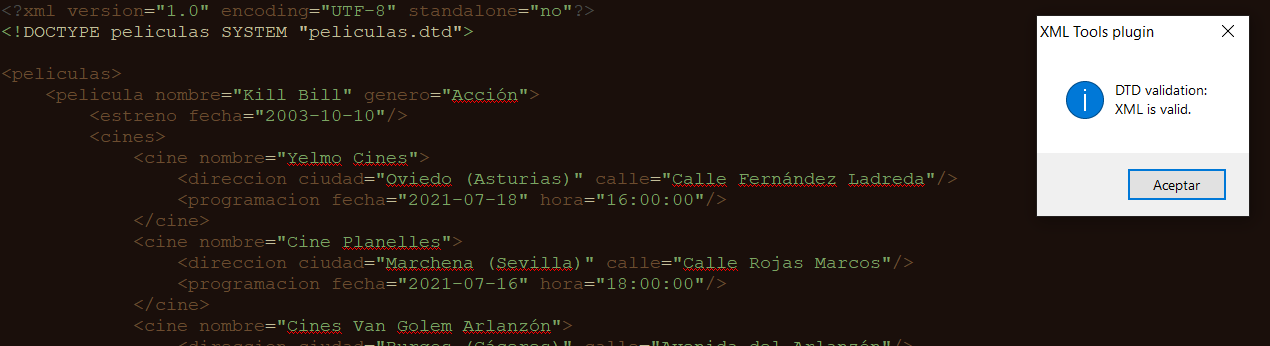


**[]**

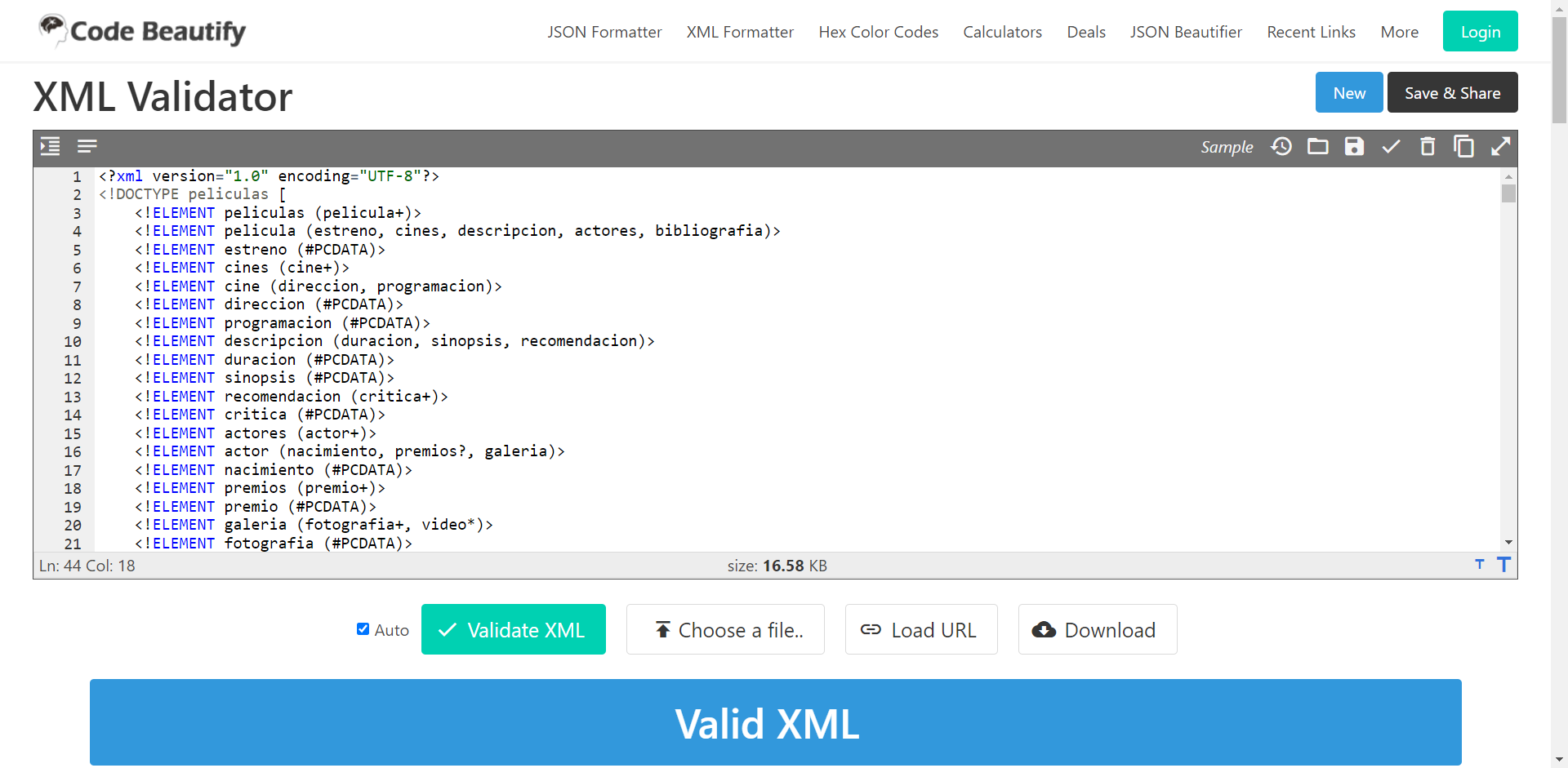
## Validación con DTD

Compruebo a través de distintos métodos que el fichero peliculas.xml es válido según el fichero peliculas.dtd que he desarrollado, en el que se indica cómo debe estar construido el XML.

### Notepad++



### Code Beauty

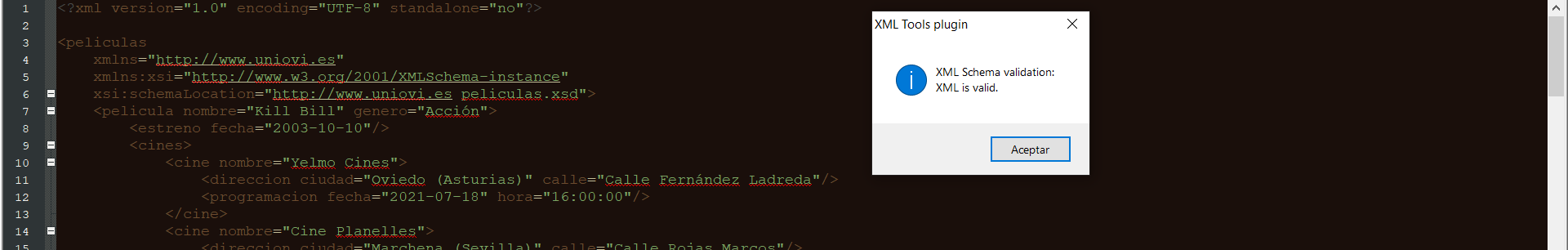


**[]**

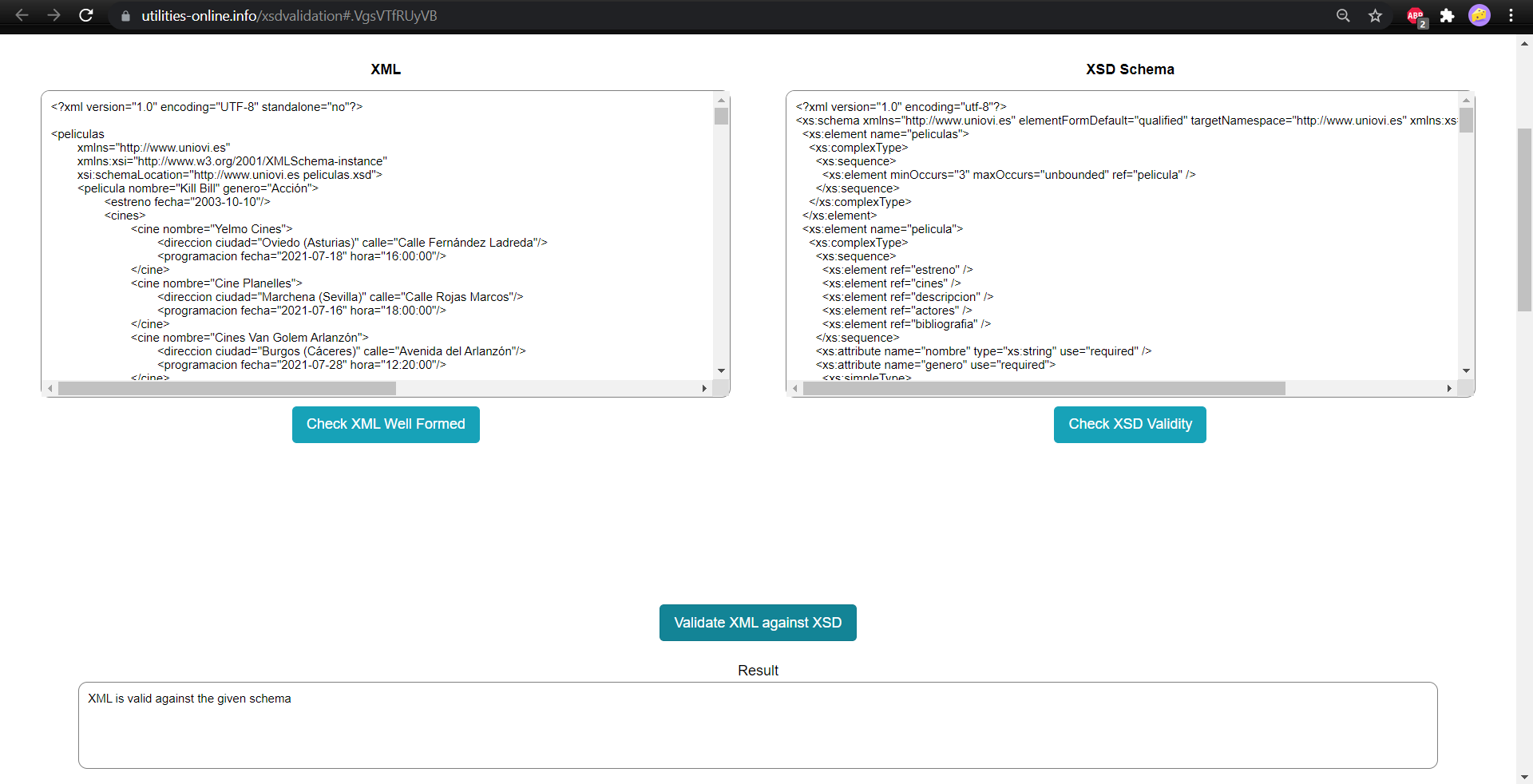
## Validación con XML Schema

Compruebo a través de distintos métodos que el fichero peliculas.xml es válido según el fichero peliculas.xsd que he desarrollado, en el que se indica cómo debe estar construido el XML.

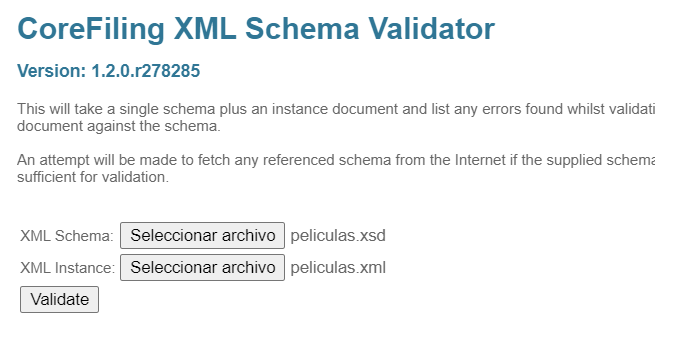
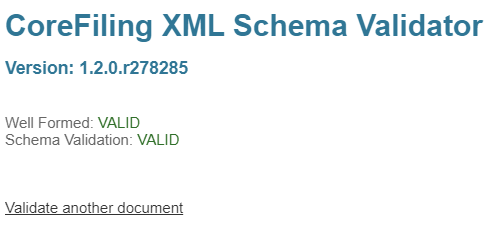
### Notepad++



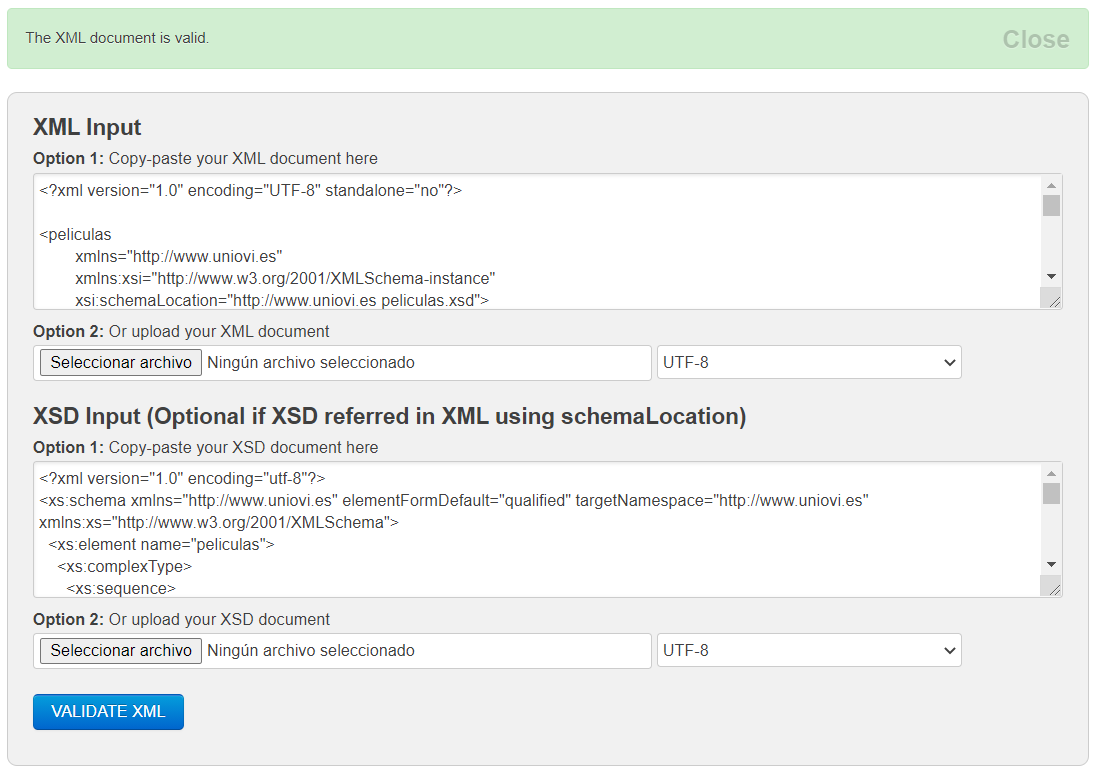
### Utilities-Online



### CoreFiling



### FreeFormatter



**[]**

## Procesamiento de XML

En mi caso, en la evaluación ordinaria realicé el procesamiento con Python, pero el cambio no me resultó demasiado complicado, además de que en parte de teoría “Procesamiento y generación de XML” se ejemplifica de una forma bastante concisa.

**[]**

## Problema con enlaces

En el procesamiento de XML no entendía por qué algunos enlaces no los formateaba como enlaces, si no como texto normal. Descubrí que se debe a que las direcciones de los enlaces no pueden terminar con una barra invertida “/”.

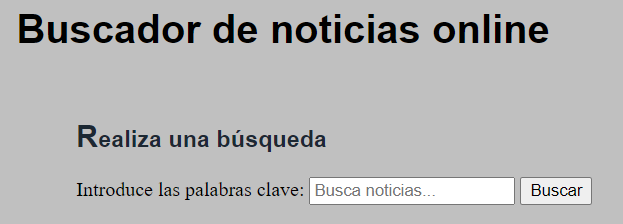
**[]**

## Descripción alternativa de imágenes

Añadí el atributo alt a las imágenes en el proceso de transformación de XML, porque es un atributo obligatorio y no lo tenía puesto, por lo que no pasaría las pruebas de validación, ni de accesibilidad.

**[]**

## API de Mediastack

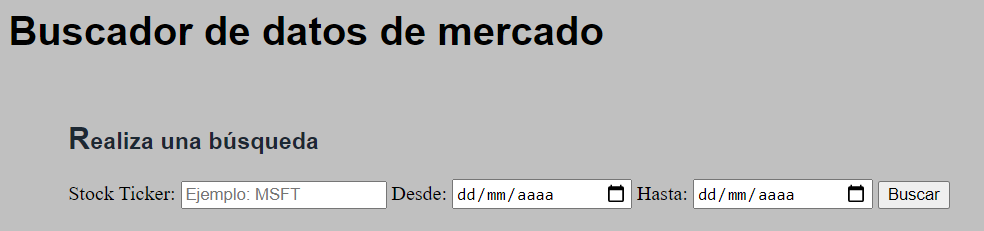
Esta API está destinada a la búsqueda de noticias a partir de palabras clave, por lo que la página en la que se implementa contiene un buscador (como se ve en la imagen), o si no se introduce ninguna palabra y se pulsa “Buscar”, aparecen las noticias más recientes.

buscador de noticias

Estas noticias se ordenan por orden de publicación, y también se indica que aparezcan noticias sólo en idioma español.

**[]**

## API de Marketstack

Esta API permite buscar datos de mercado, basta con introducir un ticker, junto con las fechas inicial y final desde las que se va a buscar la información.

En mi caso no sabía mucho sobre cómo consultar los datos del mercado, por lo que tuve que informarme un poco para tener algunas nociones básicas sobre el tema. Algo que me ayudó bastante fue la definición de ticker que adjunto a continuación.

- Ticker o símbolo bursátil: es un código alfanumérico que sirve para identificar de forma abreviada las acciones de una determinada empresa que cotiza en un determinado mercado bursátil.

Me parece útil también poner el enlace a una página con los [tickers disponibles](https://marketstack.com/search) que proporciona la API.

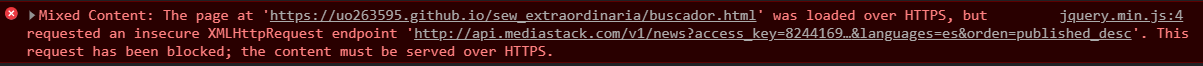
**[]**

## Problema con las APIs

Algunas de las APIs que he elegido, concretamente Mediastack y Marketstack, en su versión gratuita sólo ofrecen consultas con el protocolo HTTP, por lo que existe un conflicto con el protocolo HTTPS que estaba utilizando.



error API Mediastack



error contenido mixto

**[]**

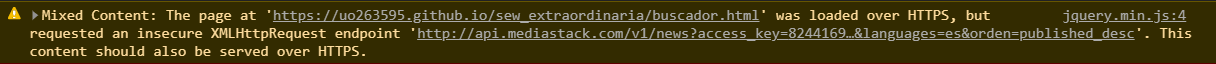
## Intento de solución

En el proceso de desarrollo, como se indica anteriormente, el acceso a mi sitio web se realiza a través de GitHub, que obligatoriamente usa HTTPS, por lo que se me ocurrió que tal vez se podía cambiar esto en la configuración del mismo, y no me equivocaba, pero es necesario un dominio (de pago) del que yo no dispongo, así que me fue imposible solucionar el problema por esta vía.

**[]**

## Solución final

Como última opción que encontré para solucionar el problema del contenido mixto, vi que en los ajustes del propio navegador se puede modificar en la “configuración de sitios”, una opción para permitir el contenido no seguro, de esta forma las APIs que están ocasionando problemas pueden proporcionarnos los datos y no es necesario cambiar el protocolo general de la página.



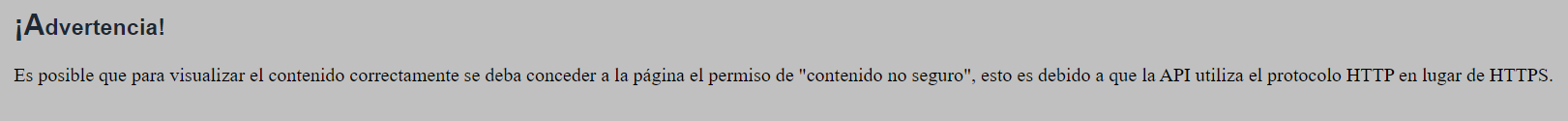
solución contenido mixto

**[]**

## Advertencia contenido mixto

Con el fin de advertir al usuario de la página web de la posible necesidad de modificar la configuración de forma que se admita el contenido mixto, como se indica anteriormente, añadí una advertencia en los apartados en los que sea necesaria, en este caso en los sitios en donde de usan las APIs de Mediastack y Marketstack.

Estas advertencias desaparecen una vez se realiza con éxito una primera búsqueda exitosa, porque se entiende que si el usuario a podido llegar a realizar dicha búsqueda va a poder realizar las posteriores.

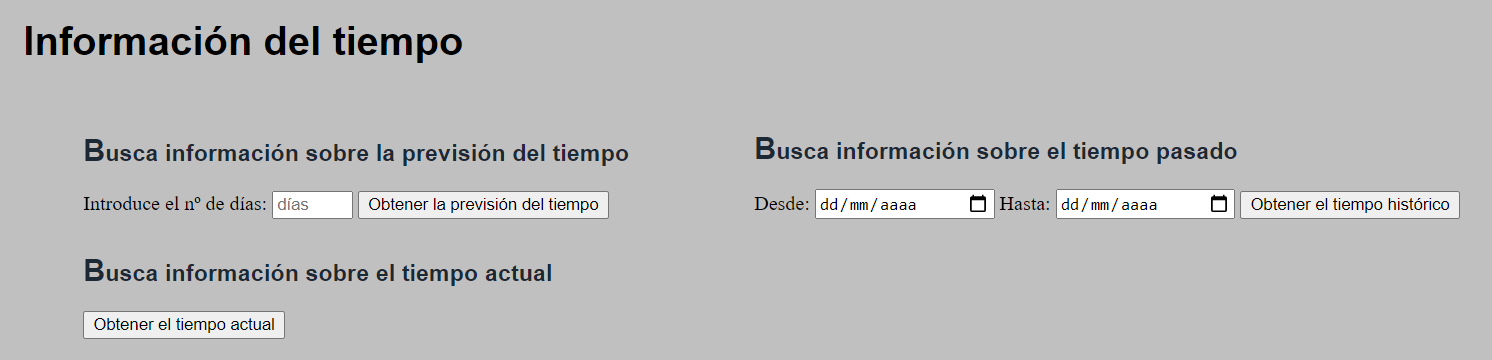


advertencia contenido mixto

**[]**

## API de Weatherbit

Esta API permite visualizar datos acerca del tiempo, en mi caso utilizado las funciones tanto de visualizar el tiempo presente, como el pasado y la previsión de hasta 16 días.



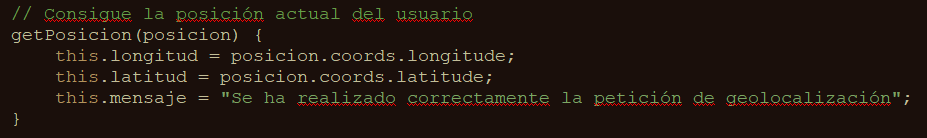
consulta del tiempo

Además, esta funcionalidad utiliza a su vez la API de Geolocalización de HTML5, ya que el tiempo mostrado se corresponde con la localización actual del usuario.



geolocalización

Una vez consultada la localización del usuario, se asignan la longitud y la latitud a las variables que posteriormente se usarán en la aplicación para conseguir los datos del tiempo de la ubicación.



asignación de la latitud y la longitud

**[]**

## Descargar cartelera



En la página dedicada a visualizar las carteleras de cine también puede descargarse la cartelera actual, gracias a un enlace.

enlace para descargar



código del enlace

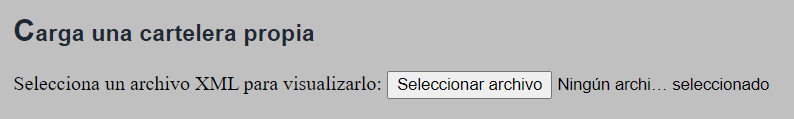
Una vez descargada la cartelera esta se puede guardar y volver a cargar en la página para su posterior visualización en un formato más fácil de comprender.

**[]**

## API File de HTML5

Para que el usuario pueda cargar una cartelera que haya descargado con anterioridad se hace uso de la API File de HTML5.

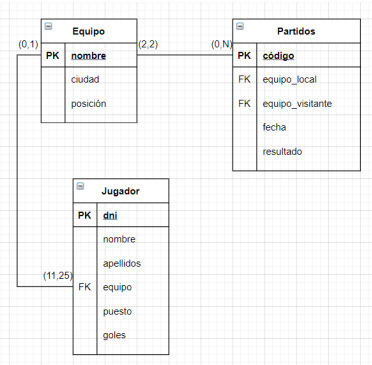
Una vez seleccionado el fichero deseado se pasa a realizar el procesamiento del fichero, este proceso es equivalente a como se hace con el proceso de cargar la cartelera actual que se proporciona por defecto.



cargar cartelera

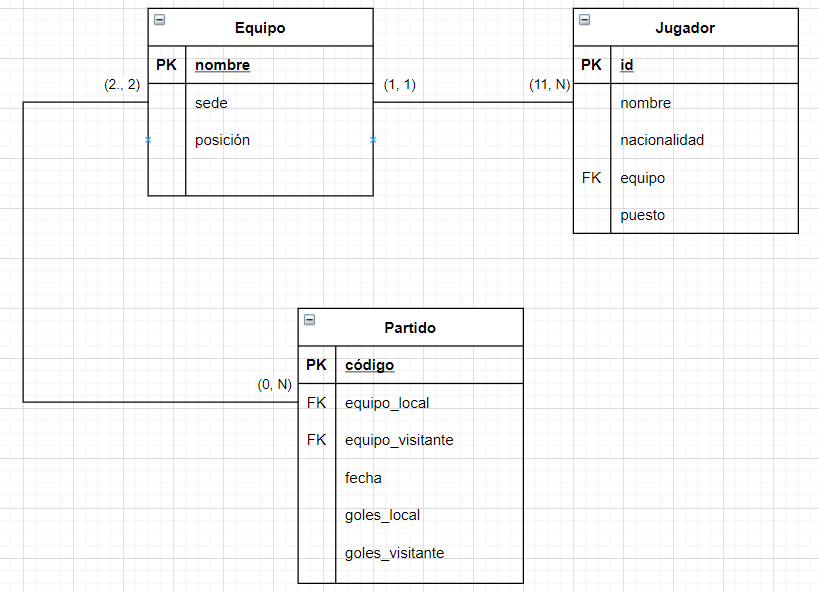
**[]**

## Modificaciones base de datos

Realicé algunas modificaciones en la base de datos que había propuesto. Estos se basan en algunos cambios de atributos en las tablas:

- Tabla Equipo: cambio del atributo ciudad por el atributo sede, este cambio se debe únicamente a que el nuevo término me pareció más adecuado, pero no tiene ninguna repercusión en el diseño final.

- Tabla Partidos: esta tabla no ha sufrido cambios.

- Tabla Jugador: el atributo dni pasa a llamarse id, pero a términos prácticos implica lo mismo, los atributos nombre y apellidos se juntan únicamente en nombre que los recoge a ambos, y se añade un nuevo atributo nacionalidad, a su vez se elimina el atributo goles debido a la dificultad a la hora de recuperar datos para hacer una base de datos lo más realista posible.

Base de Datos antigua

Estas modificaciones se justifican debido a que los datos son recogidos de varios datasets correspondientes a datos de LaLiga del año 2018, por lo que ha sido necesario hacer estas pequeñas adaptaciones para asegurar la coherencia final.

Base de Datos nueva

**[]**

## Carga de datos

Para poder visualizar el funcionamiento de la aplicación, he recopilado información a partir de un dataset y algunas páginas que muestro a continuación:

[Clasificación LaLiga 2018-19](https://www.google.com/search?rlz=1C1GCEA_enES807ES807&q=la+liga+2018-19+clasificacion&sa=X&ved=2ahUKEwiKoPrswpXxAhVEXRoKHeCABq8Q1QIwEnoECA8QAQ&biw=1536&bih=722#sie=lg;/g/11f57gslw8;2;/m/09gqx;st;fp;1;;)

[Dataset LaLiga 2018-19](https://datahub.io/sports-data/spanish-la-liga)

[Información LaLiga 2018-19](https://www.livefutbol.com/jugador/esp-primera-division-2018-2019/)

Con estos datos desarrollé tres ficheros .cvs a partir de los cuales realizo la carga en la base de datos.

En caso de que se navegue a la página de información deportiva y la base de datos se encuentre vacía se pasa a ejecutar una función en la que se recupera la información de estos tres ficheros anteriormente mencionados y se añaden a la base de datos, pero si ya hay datos estos no se modifican.

**[]**

## Diseño del apartado de información deportiva



información deportiva

A través de esta interfaz se puede buscar información relacionada con LaLiga (temporada 2018-19), desde sus jugadores, los equipos y los partidos que sucedieron en esta temporada, además de los equipos con sede en una ciudad determinada y las rivalidades entre equipos, que en este último caso se deberá realizar la búsqueda con los nombres completos y correctamente escritos.

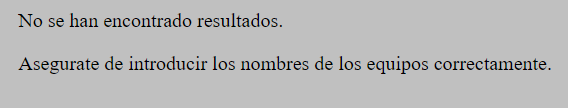
**[]**

## Búsqueda sin resultados

En la página que muestra información de LaLiga puede que se introduzca alguna entidad de la que no se tienen datos, por lo que aparecerá el siguiente “búsqueda sin resultados”.

En el caso de buscar una rivalidad también puede haber sucedido que se haya introducido mal el nombre de alguno de los equipos, pero en este caso el mensaje que aparece es un poco distinto, tal como se muestra a continuación:

búsqueda sin resultados

Aunque en cualquiera de los dos casos el mensaje indica una inserción incorrecta en los parámetros que definen la búsqueda.

rivalidad sin resultados

**[]**

## Búsqueda parcial

En los buscadores de LaLiga se permite realizar una búsqueda parcial, de modo que si se introduce por ejemplo “Real” en el input correspondiente a la búsqueda de equipos aparecerán los datos correspondientes a los equipos “Real Madrid” y “Real Sociedad”.

Este tipo de búsqueda no funcionaría únicamente en el caso de rivalidades.

**[]**

## Todos los resultados

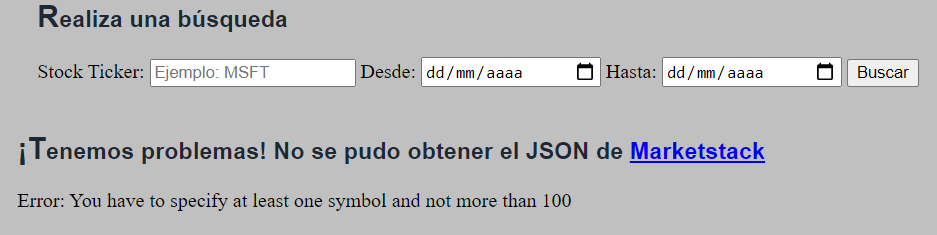
En algunos de los campos de búsqueda (exceptuando los numéricos y los que implican fechas) se permite realizar una búsqueda con el campo vacío, esto hará que se muestren todos los resultados posibles. En el caso de las noticias se mostrarán las más recientes, y en el caso de información deportiva aparecerán todos los datos almacenados en la base de datos que tengan que ver con el campo en el que se haya realizado la búsqueda.

**[]**

## Mensaje de error en las APIs

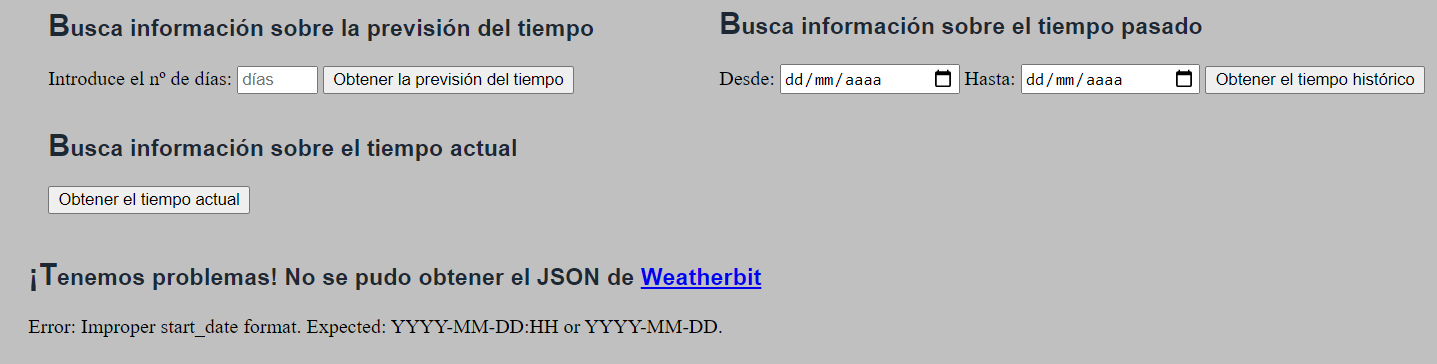
Existen varios errores que pueden devolver las APIs, normalmente por la falta de algún parámetro en la búsqueda. Cuando alguno de estos errores sucede, la página lo muestra en un mensaje.

En el caso de la API de Marketstack podría ser que no se introduzca un ticker válido o que directamente no se introduzca ninguno, por lo que aparecerá el siguiente mensaje:



error Marketstack

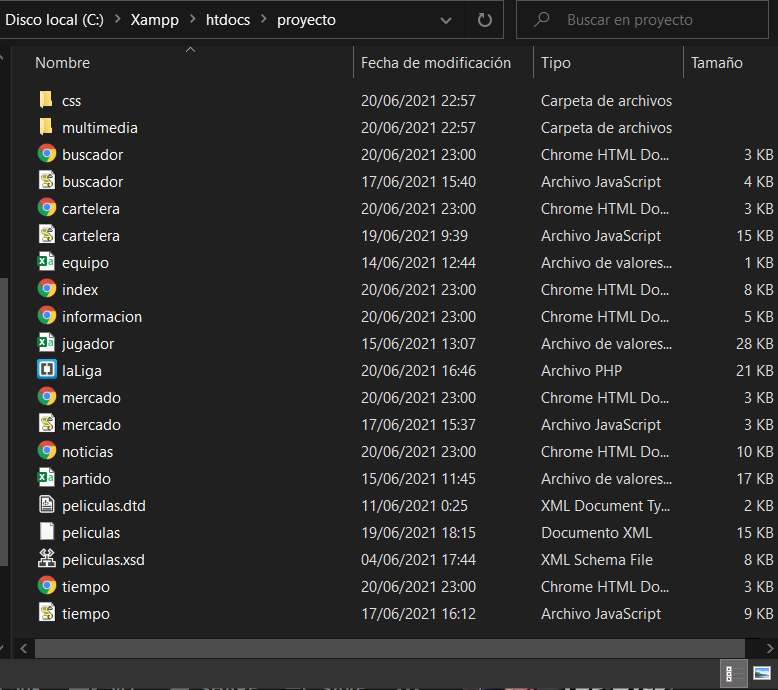
También en Weatherbit puede que no se haya introducido una fecha válida en la búsqueda del tiempo histórico, así que de igual manera se muestra en un mensaje:



error Weatherbit

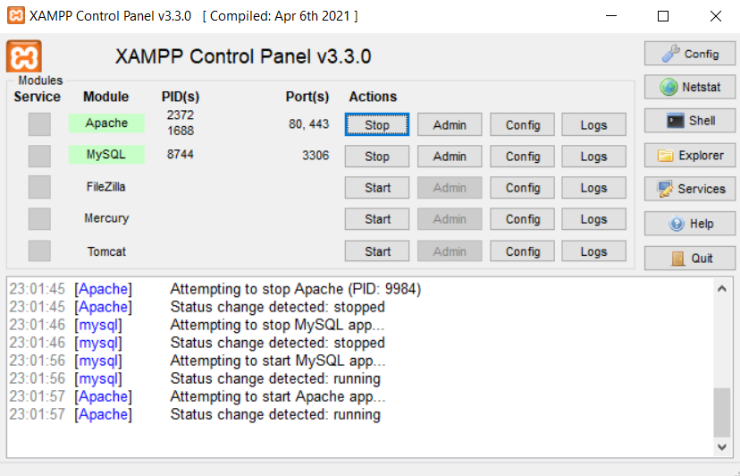
**[]**

## Funcionamiento en máquina local

Para desplegar el proyecto en local utilizando Xampp, se copia todo el contenido y los recursos locales que sean necesarios en la carpeta htdocs, que se crea automáticamente tras la instalación de Xampp.

En mi caso creé una carpeta llamada “proyecto” dentro de esta primera y metí ahí el contenido necesario.

carpeta htdocs

Una vez organizado el contenido, basta con activar Apache y MySQL en Xampp.

Finalmente, en el navegador escogido, se accede a localhost/proyecto (o /<la carpeta que se haya creado>). En este punto si se tiene un fichero index, se mostrará ese directamente, si no será necesario especificar el fichero concreto que se quiere visualizar.

Xampp



visualización en navegador

Como observación, destacar que en este caso el sitio web no tendría el problema con los protocolos y el contenido mixto que sucede al desplegarlo con GitHub (este problemas se explica más en profundidad en entradas anteriores).

**[]**

## Funcionamiento en la nube

### Paso 1: Lanzamiento de la instancia

**[]**

Validación HTML5

Contenido

**[]**

Título

Contenido