

Ingenieria informatica de software

Software y Estándares para la Web



6 de junio de 2018

Martin pelaez diaz

# Introducción

El siguiente proyecto, nace como una propuesta para la evaluación continua de la asignatura Softwares y Estándares para la Web.

Para ello se propuso la creación de una página web que simulase la venta de pizzas, con el objetivo de poder mostrar los conocimientos adquiridos durante la asignatura.

Con este objetivo se realiza el sitio web aplicando HTML5 para la realización de las vistas de la página, de la misma forma se utilizan los conocimientos adquiridos de CSS para mostrar de forma correcta y tratar los estilos de dichas vistas.

A modo de demostrar lo aprendido de XML se decidió crear varios archivos XML los cuales contienen la relación de pizzas que se van a ofertar en la aplicación, juntos con estos archivos se crearon sus correspondientes validadores DTDs y XMLSchemas, así mismo para poder mostrar al usuario el contenido de los archivos XML de una forma amigable para el se utilizaron transformaciones XSLT.

En lo referente a la computación en la parte del cliente se decidió utilizar JQuery para la creación de un reloj para la web, se ha usado JSON para intercambiar información entre el cliente y el servidor y de igual forma se usa Ajax para enviar la información y recibir la información del servidor.

Como forma de probar los conocimientos en la utilización de APIs y servicios web, se decidió usar la API de geolocalización de HTML5 y consumir los servicios de Google Maps, para obtener y mostrar la localización del cliente y de los establecimientos a los que representa la página web respectivamente.

Por otro lado, la computación en el servidor se hace sobre PHP, se utiliza para validar la información del cliente y guardarla en una base de datos MYSQL y se envía desde PHP un mensaje a los usuarios de que se ha procesado su pedido correctamente.

Por último, el funcionamiento de todo lo anterior ha sido probado sobre la maquina local XAMPP, y se han utilizado los validadores de HTML5, CCS, XML, etc. con el fin de comprobar que están bien formados todos los ficheros que conforman el sitio web.

# Bitácora

## 09/05/2018 15:43

Se creo el repositorio en [GitHub](https://github.com/martinp95/SEW) para la realización del trabajo, de forma que facilitara la creación de la bitácora pensándolo desde el punto de vista de la hora y fecha de cada uno de los commit, así como su contenido.

En este primer commit al repositorio se añaden:

* Imágenes de las pizzas que se van a mostrar en la aplicación.
* Pizzas.xml y pizzasEspeciales.xml
* Pizzas.dtd y pizzas.xsd
* Pizzas.xsl
* Index.html, productos.html, locales.html, contacto.html y conocenos.html

Por un lado, los archivos XML contienen la relación de pizzas que se va a ofertar en el sitio web.

Así mismo, los ficheros DTD y XMLSchema se crean y son utilizados para validar los archivos XML.

Además de esto, en lo referente al tratamiento de los XML, se crea el archivo pizzas.xsl para poder transformas los XML para mostrarlos como una vista de la aplicación web.

Por ultimo se crean los principales archivos HTML que conformaran las principales vistas de la aplicación.

## 09/05/2018 16:45

Creación del fichero productos.html desde donde se podrá acceder a la oferta tanto de pizzas tradicionales como de pizzas especiales.

## 10/05/2018 16:53

Se crea el fichero reloj.js el cual crea un reloj en JQuery para la aplicación.

Dicho reloj se situará en la parte superior derecha de la pantalla mostrando la hora y la fecha actual y actualizándose cada segundo.

## 12/05/2018 13:59

Se modifican cada una de las vistas de la aplicación para pasar a mostrar el reloj.

Para ello se crea un div que lo va a contener, y se inserta el script correspondiente.

## 16/05/2018 00:26

Se crea el fichero mapa.js el cual es un archivo JavaScript en el que se crea un mapa de Google Maps.

Se inserta en mapa, así como el script dentro del fichero locales.html donde se mostrará el mapa con las ubicaciones de los locales de venta de pizzas, al igual que la del usuario de la aplicación en caso de que este de consentimiento a que el sitio web conozca dicha ubicación.

En este punto, cabe destacar una modificación sobre la propuesta realizada para la realización del proyecto, en la propuesta se dice que se consumirán los servicios de Openlayers, pero dado que ya poseía una API Key de Google Maps y por la facilidad a la hora de encontrar información se decidió pasar a consumir los servicios de esta última.

## 16/05/2018 20:38

Creación del código JavaScript dedicado a la validación del formulario de la vista contacto.html así como modificaciones sobre el formulario de dicha vista para poner los campos como obligatorios, el tipo del campo de email como email etc.

También se introducen cambios en la organización del contenido de la vista conocenos.html.

## 17/05/2018 14:32

Se añaden al proyecto los ficheros PHP que permiten la conexión con la base de datos y guardado de comentarios de los usuarios en la base de datos a través del formulario de contacto.

## 28/05/2018 13:08

Se eliminan comentarios de error del PHP encargado de hacer la conexión con la base de datos, y se hace que el PHP encargado de insertar los comentarios en caso de éxito muestre una salida HTML en la que se le comunique al usuario que todo se ha realizado con éxito.

## 28/05/2018 19:24

Se crea el archivo carrito.js encargado de ser el carrito de la compra de compra de la aplicación. Este se crea de forma incompleto, y se añade el script en la vista generada por el archivo pizzas.xsl.

## 29/05/2018 13:14

Se crea un aside en la vista que genera el archivo pizzas.xsl en el que se mostrara el carrito de la compra con los productos en forma de tabla según el usuario los vaya añadiendo.

También se completa el archivo JavaScript encargado de gestionar dicho carrito de la compra.

## 29/05/2018 22:48

Se modifica el fichero carrito.js ya que se vio la necesidad de compartir información entre vistas de la aplicación, para lo que se aprovechó la API Web Store de HTML5.

## 30/05/2018 00:12

Se introduce en el archivo carrito.js la redirección a la vista para la formalización del pedido y se crea dicha vista que consta de un formulario para que el usuario introduzca sus datos de cara a guardarlos en la base de datos y poder identificar el pedido.

## 30/05/2018 00:34

Creación del script dedicado a tramitar el pedido, su función es validar los datos del formulario en el cliente, sacar los datos del pedido de la cache de sesión y hacer la llamada al servicio web que se creara en PHP a través de Ajax para que se guarde el pedido en base de datos.

## 30/05/2018 02:08

Introducción de Ajax en el script para tramitar el pedido.

Creación del servicio en PHP para guardar el pedido del usuario.

## 30/05/2018 13:01

Se modifica el script encargado de la tramitación del pedido para obtener los datos introducidos por el usuario y poder enviárselos con Ajax al servicio web.

## 31/05/2018 16:59

Creado servicio web en PHP para la inserción del pedido en la base de datos tratando archivos de tipo JSON y enviando mensaje de éxito o de error en función de lo que haya pasado.

En el script del cliente se mostrará un alert con lo que haya sucedido y en caso de éxito una vez que se cierre el alert se redirigirá al usuario a la vista principal de la aplicación.

## 04/06/2018 12:16

Creación de la hoja de estilos para el reloj de la aplicación y añadida la misma a todas las vistas de la aplicación.

## 04/06/2018 12:46

Creación de la hoja de estilos para la barra de navegación de la aplicación y añadido de la misma a todas las vistas de la aplicación.

## 05/06/2018 15:58

Terminada hoja de estilos principal de la aplicación donde se le aplican los estilos al body para aparecer centrado y al footer.

Se realizan cambios menores de renombrado de clases y sangrado del código en algunas de las vistas de la aplicación.

## 05/06/2018 16:24

Creación de la hoja de estilos para los formularios de la aplicación.

## 05/06/2018 18:23

Creación del CSS dedicado a mostrar correctamente las vistas donde se encuentra la relación de pizzas.

## 05/06/2018 19:20

Modificación de la hoja de estilos encargada de la barra de navegación para que aparezca en otro color con los bordes redondeado y mostrar el borde.

## 05/06/2018 21:51

Modificaciones sobre los HTML de la aplicación para que estén correctos y poder generar los svg con ellos.

Añadir los svg generados al proyecto.

## 05/06/2018 23:30

Modificaciones de las vistas para que pasen las pruebas de accesibilidad y estén ortográficamente correctas.

# Validación y cumplimiento de los estándares HTML5 y CSS

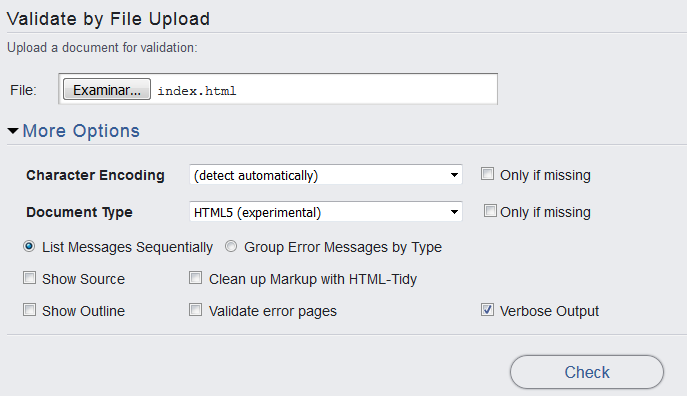
La validación de los diferentes archivos HTML que conforman las vistas de la aplicación, así como los CSS que permiten añadirles estilos a dichas vistan han ido siendo validados a medida que se realizaba la aplicación.

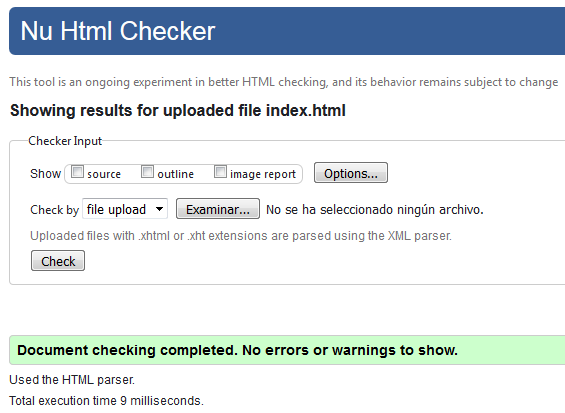
Para la validación se han utilizado los validadores del W3C:

* <https://validator.w3.org/>
* <https://jigsaw.w3.org/css-validator/>

A continuación, mostrare un par de ejemplos del resultado de la validación de los HTML y CSS que conforman el sitio web.

## Validación HTML

Para validar los archivos dado que no acepta una URL en localhost e optado por validarlos subiendo el archivo uno por uno a la web.

Dado que el archivo es un HTML valido nos muestra la siguiente salida.

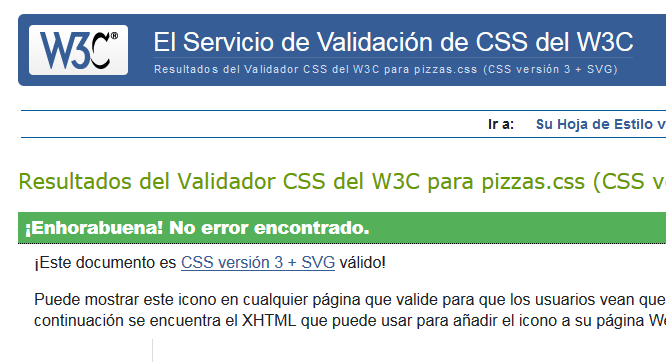
Por otro lado, los archivos también se validad generando el svg con la herramienta proporcionado por los profesores de la asignatura, que dado que es mas restrictiva que estos validadores nos asegura haber generado HTML valido.

## Validación CSS

Con la validación de los archivos CSS nos pasa lo mismo que con los HTML por lo que se validan subiendo directamente el archivo en vez de usar una URL.



Si los archivos están bien formados, como es el caso, el mensaje que nos devuelve el validador es como el siguiente.



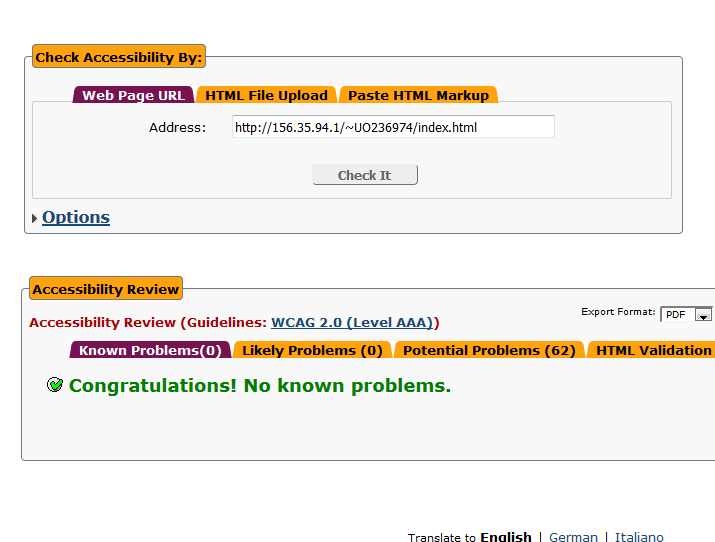
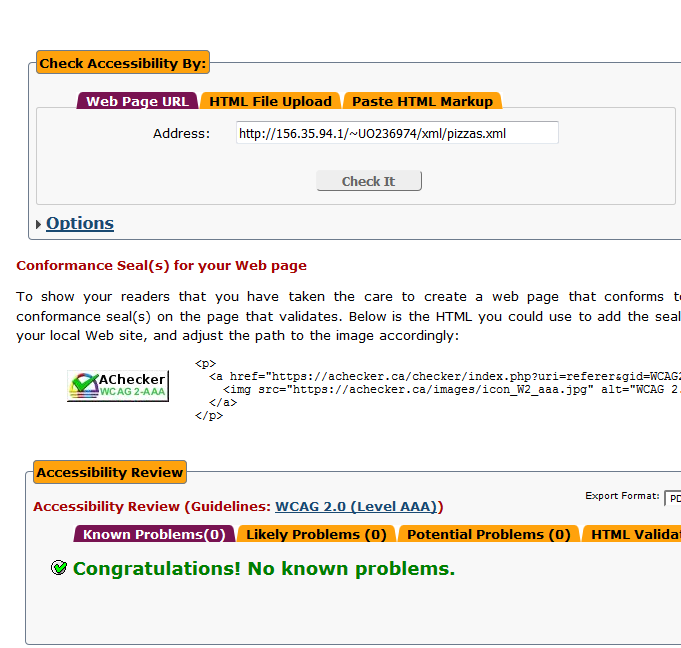
# Pruebas Accesibilidad

Para realizar las pruebas de accesibilidad sobre la web, y propiciado porque no he encontrado muchas validadores de accesibilidad que permitan subir el código, sino que lo prueban a partir de la URL, me he decidido por utilizar los dos principales que vimos en clase de prácticas de la asignatura.

Ya que uno de ello solo valida vía URL se han subido los archivos a Ritchie para por lo menos el contenido estático de la aplicación poder validarlo.

Validador:

* <https://achecker.ca/checker/index.php>

A continuación, se muestra un ejemplo de las pruebas de accesibilidad.

# Pruebas Maquina Local XAMPP

Comentar la configuración como en clase de practicas

Poner alguna captura de ficheros dentro de tal y alguna de algún php ejecutado o algo asi para documentarlo