

## Software y estándares para la Web

---

### **PROPUESTA DE PROYECTO CONVOCATORIA EXTRAORDINARIA 2018/2019**

**Nombre:** Germán Iglesias Ramos

UO202549

## Contenido

Temática.....	2
HTML.....	2
CSS .....	3
XML .....	4
Computación ECMAScript en el cliente .....	5
Computación PHP en el servidor .....	6
Pruebas.....	8

## Temática

### Resumen de la temática general del sitio web a desarrollar en el proyecto

El sitio web que propongo para desarrollar, es la creación de una página web que pueda “simular” la venta online de botellas de sidra dentro de un abanico de diversas marcas y otros productos complementarios (escanciadores, vasos, tapones, etc.), como concepto general de la web. Asimismo, se describirán los establecimientos ficticios de los que disponemos, pensando en poner un nombre a nuestra “empresa”. Sería algún nombre familiarizado con el mundo sidrero, tradicional... existen numerosos ejemplos. Con este proyecto creo que puedo completar y abarcar todos los aspectos que se han impartido y desarrollado en las prácticas de la asignatura.

## HTML

Se deben especificar los documentos HTML5 que van a componer el proyecto, estableciendo el mapa de navegación entre ellos.

En general los sitios web que he pensado para guiarme a la hora de decidir los html5 que compondrán el proyecto serán:

- Index.html: nos servirá de punto de entrada del sitio web, con una breve bienvenida al sitio web y algún enlace directo (mínimo dos) a las sidras que ofertamos y nuestros establecimientos.
- Historia.html: en este html5, pondremos una historia inventada acerca de la empresa, de sus orígenes, sus miembros, etc. En general mostrare en este documento una mediana descripción de la empresa y sus inicios.
- Productos.html: en él tendremos tres etiquetas multimedia de enlace que nos derivarán a los productos que ofertamos, ya sean las sidras naturales, las sidras espumosas y otros productos como escanciadores, vasos, etc.
- Contacto.html: este documento html5, es muy común en muchos sitios web de venta online. En la nuestra se presentará un formulario a rellenar por el cliente para contactar con nuestra empresa introduciendo sus datos personales. Tras rellenar con sus datos podrá enviarnos sus sugerencias o cualquier otra cuestión.
- Establecimientos.html: aquí se mostrará un mapa con la localización de los diversos locales de los que dispone mi empresa de venta de sidras y la localización del cliente. Nos ayudaremos de APIs de geolocalización que nos exigirá la propuesta en otro apartado más adelante.
- tramitarPedido.html: este documento html5, nos presentará un formulario que deberá de rellenar el cliente con sus datos para formalizar el pedido que ha hecho.

Varios de estos documentos HTML5, estarán enlazados a través de un menú de navegación y siguiendo una estructura semántica similar que permita la coherencia del acceso a los contenidos.

Menú de navegación:

[Principal.html](#) | [Historia.html](#) | [Productos.html](#) | [Contacto.html](#) | [Establecimientos.html](#)

## CSS

Se debe establecer el aspecto general de los documentos del sitio web del proyecto, estableciendo la disposición general de los elementos (1 columna, 2 columnas, ...) o si se va a utilizar alguno de los módulos CSS3 como flexbox, o Grid Layout.

En un principio resulta complicado establecer que aspecto tendrá mi sitio web, ya que es un apartado que suele ir modificándose y evolucionando a medida que el proyecto se desarrolle, además tenemos que tener en cuenta todos los problemas que nos pueden surgir a la hora de pasar las diferentes validaciones de accesibilidad, para tener una estructura correcta que favorezca a cualquier usuario a acceder a nuestra web, siendo una apartado no difícil pero si que requiere de una serie de medidas que se deben de solucionar, y también a necesidad de tener una buena disposición de los elementos para las validaciones en cuanto a la usabilidad que se exigen en la propuesta. Dicho lo cual, una idea general para dotar de estilo completo a mi sitio web planteado y que las vistas tengan un estilo común sería el siguiente:

1. Cabecera (<head>): esta parte sabemos que no es visible al usuario que acceda a mi web, pero es la parte imprescindible donde nuestros documentos contendrán los metadatos <meta>, los links <link> para enlazar recursos, hojas de estilo, el título <title> de la página que es imprescindible, nuestros scripts <script>, etc.
2. Cuerpo (<body>): dentro del cuerpo ya podremos mostrar al usuario una clara disposición general de la estructura que percibirá en el global de nuestra página web.
  - Encabezamiento (<header>): en el encabezamiento pretenderé que sea lo más rígido y estático posible para que se mantenga una unidad visual en todas las vistas de mi página web. Sé que pretende ser como una ayuda introductoria para el usuario que nos visite por eso haré que aparezca un logotipo de mi empresa con el nombre de ella y que mantenga un enlace con la página principal.
  - Navegación principal (<nav>): esta sección contendrá enlaces a otras páginas o partes de nuestra web, ayudando de este modo al usuario a encontrar los componentes que él desee. En la parte que redacte correspondiente a HTML defino un mapa de navegación al final del texto que sería el que emplearía.
  - Contenido principal (<main>): este contenedor identificara el contenido general de mis documentos, mostrando únicamente el contenido exclusivo de cada documento (excluyendo por ejemplo el pie de página).
  - Sección genérica (<section>): aquí representare la temática del contenido, con los correspondientes encabezados (<h2>, <h3>, etc.), párrafos (<p>), etc.
  - Pie de página (<footer>): estará debajo de todas las secciones, y en el sitio web comparando y echando un vistazo a otros portales de venta online añadiré guiándome de ellos una navegación secundaria y partes adicionales como elementos de aviso de copyright, de autor y políticas de acceso diversas.

En cuanto a la disposición general de los elementos me declino por usar una columna en vez de dos o alguno de los módulos de CSS3. Sé que en los ejercicios de las prácticas había una actividad opcional que consistía en investigar en esos módulos y hacer algún ejemplo, pero básicamente decantarme en composiciones de una sola columna me resultara más apacible el desarrollo del proyecto, o eso creo. Por otro lado, he visto en documentaciones adicionales que el uso de esta disposición (una columna), no es absolutamente necesario usar contenedores <main>, pero de

igual forma creo que semánticamente no es incorrecto y además me ofrecerá más facilidades y flexibilidad en cuanto al estilo en relación a los elementos, reglas, imágenes, etc.

## XML

Se debe diseñar un archivo XML, de complejidad similar a las prácticas de la convocatoria ordinaria, con varios niveles de anidamiento. Se debe especificar la temática del archivo XML.

- Deberá contener validadores DTD y Schema
- Deberá contener transformaciones XSLT
- Deberá contener procesamiento de XML en ECMAScript utilizando jQuery

Para completar en el proyecto el módulo de XML creare tres archivos XML que me van ayudar a representar la información estructurada en la web, separando clara y radicalmente el contenido y el formato de presentación.

1. Sidras.xml: consistirá en un archivo XML que contendrá la relación de sidras naturales que ofertará mi aplicación web. Como he dicho, mostrara las diversas variedades (mínimo 15) de marcas de sidra que mi web vende, y en cuanto a las etiquetas que conformaran los distintos niveles se me ocurren los siguientes requisitos como: título, imagen de la sidra específica que ofertamos, el nombre específico de la sidra, una mediana descripción de su origen, sabor, color, ... su graduación de alcohol, y el volumen de la botella. En un principio son los requisitos mínimos que se me ocurren.
2. sidrasEspumosas.xml: es un archivo muy similar al anterior, por no decir igual, salvo que la temática únicamente varia, ya que en el archivo XML se ofertaran las sidras espumosas.
3. otrosProductos.xml: este archivo XML difiere un poco de los anteriores, en él mostrare también una oferta de productos varios como escanciadores de sidra, tapones escanciadores, vasos de sidra, barriles de sidra, etc., dónde ya los requisitos serán un tanto distintos, ya que constará de título, imagen, nombre, descripción y precio, pero etiquetas como grados de alcohol y volumen no aparecerán.

Se harán las correspondientes comprobaciones de que los documentos estarán bien formados en los distintos navegadores utilizados durante el curso.

Crear los correspondientes validadores DTDs que me permita validar los XMLs bien formados comprobando con alguna herramienta vista en clase su validez, para asegurarme de su uniformidad. Asimismo, también generar los XML Schemas correspondientes para comprobar que los documentos XML son válidos, sabiendo que dichos XML Schemas son más potentes que los DTD.

Aplicar a los documentos XML una hoja de estilos XSLT. Por lo pronto, creo que diseñare dos, ya que para los archivos sidras.xml y sidrasEspumosa.xml me servirá con hacer una hoja de estilos XSLT común a ambas, pero para otrosProductos.xml deberé de crear otra hoja de estilos. De esta forma con las transformaciones XSLT nos presentara el contenido de dichos XML más familiarmente (HTML).

Por último, en cuanto al procesamiento de XML me surgen un par de cuestiones que tendré que consultarlas con los profesores de la asignatura. Por un lado, puedo usar la herramienta (programa xml2svg) proporcionada por los profesores para la generación de árboles XML en

SVG, de manera que aparezcan todos los elementos y sus atributos. Y por el otro lado, durante las practicas correspondientes a esta parte un trabajo que se pedía, consistía en la generación y el diseño de un programa que tuviera un archivo XML de entrada y la salida de la aplicación fuese otro archivo XML. Para el cual, creé una aplicación en java que permitía, tras pasarle un archivo XML de entrada con la relación de varios jugadores de futbol de distintas posiciones, aplicando un filtro para determinar qué posición prefería el cliente, la generación de un XML con los jugadores de la posición seleccionada. Podría trasladar esa aplicación a mi web de sidras haciendo un “screening” según el precio por ejemplo o según los grados de alcohol. Por lo que un principio incluiré las dos propuestas en el proyecto de mi sitio web.

## Computación ECMAScript en el cliente

Se debe describir la aplicación de computación en el cliente que se desarrollará en ECMAScript. Deberá usar obligatoriamente el paradigma de orientación a objetos. No se admitirá el paradigma procedimental.

- Deberá usar jQuery
- Deberá usar JSON
- Deberá usar Ajax
- Deberá usar el API de HTML5 de Geolocalización
- Deberá usar otro API de HTML5
- Deberá consumir servicios Web. Es necesario especificar el sitio (o los sitios) web de donde se van a consumir los servicios web

Para esta parte propondré para hacer computación en el cliente usando ECMAScript una serie de “programas” y ficheros que garanticen el desarrollo usando orientación a objetos y no paradigma estructural, para ello utilizaremos JQuery para la creación de la fecha (fecha.js) y un reloj analógico (analógico.js) en constante funcionamiento que se situaran en la parte superior de las pantallas. El reloj analógico, es un proyecto que desarrolle a causa de una de las tareas optativas de las prácticas y que creo que lucirá en mi sitio web. Además, estos dos ficheros (.js) son bastante similares a los ejemplos descritos y vistos en teoría como era el cronómetro, por ejemplo. Así pues, creare un archivo (formulario.js), como el que se enseñó en clase de teoría y guiándonos por el ejemplo de JavaScript que está en el campus virtual modificándolo y adaptándolo a nuestra web y necesidades, para que el cliente se pueda poner en contacto con mi empresa, como habíamos anticipado en el apartado inicial de HTML. Estaría enlazado en el documento contacto.html y el cliente tan sólo tendría que rellenar sus datos personales tales como: nombre, apellidos, email y su mensaje, por ejemplo.

En lo que concierne al uso de Ajax y JSON, trataré de usar JSON para intercambiar información entre el cliente y el servidor al igual que Ajax para el envío y recepción de información del servidor. Como estoy proponiendo el portal de venta online de sidras es lógico pensar que debo de crear un archivo (carrito.js), el cuál maneja el uso del estándar JSON y la tecnología Ajax para que en los archivos descritos en el módulo XML (sidras.xml y otrosProductos.xml) manejen este JQuery. A su vez, crearemos un fichero (pedido.js) que use las anteriores tecnologías y estándares descritas que nos permita una vez añadidas las sidras u otros productos en el carrito permita rellenar los datos del cliente para formalizar de una manera y valga la redundancia el pedido.

Para reflejar lo aprendido en la utilización de las APIs y servicios web, uno de los ejemplos descritos en clase nos enseñaba la API de geolocalización de HTML5, que nos permite consumir los servicios de Google Maps, por lo que creare un archivo llamado establecimiento.js, que consistirá en un archivo JavaScript que crea un mapa de Google Maps. En la vista establecimientos.html introduciremos el script donde mostrará las localizaciones de mis establecimientos y la ubicación del usuario. Otra API que usaré será la API Web Storage de HTML5 para que el fichero carrito.js comparta información entre vistas de la aplicación, hay un ejemplo claro de su uso en los ejemplos JavaScript proporcionados en el campus virtual. Esta API de almacenamiento web me proporcionara herramientas mediante las cuales el navegador almacenara información de una forma más intuitiva que si utilizase las cookies.

## Computación PHP en el servidor

Se debe describir la aplicación de computación en el servidor que se desarrollará en PHP. Se debe describir la Base de Datos MySQL y cómo van a ser tratados estos para obtener informes, cálculos, etc. La Base de Datos debe tener un mínimo de 2 tablas relacionadas y normalizada.

El usuario y password de la Base de Datos MySQL en XAMPP debe ser el mismo que en la convocatoria ordinaria.

Para esta última parte, la computación en el servidor la hare sobre PHP, para poder validar la información del cliente y guardar esa información en una base de datos MySQL. Creare dos archivos PHP:

1. Pedido.php: consistirá en un script para poder tramitar el pedido de sidras u otros productos que realice el cliente. El objetivo que perseguirá este archivo es la validación de los datos que rellene el cliente (pedido.js y tramitarPedido.html) en el formulario, guardar esos datos del pedido y llamar al servicio web que se creara mediante Ajax para almacenar ese pedido en nuestra base de datos. Usaremos el estándar JSON para tratar archivos y mandar un mensaje de éxito o fracaso de la operación de inserción (del grabado del pedido).
2. Comentario.php: este fichero PHP permitirá el guardado de los comentarios de los usuarios hacia nuestra empresa almacenados en nuestra base de datos a través de otro formulario bastante similar al citado anteriormente (contacto.html y formulario.js). En caso de que sea exitosa la inserción del comentario mostraremos una salida en formato HTML con el resultado.

Para este módulo de la propuesta usare el servidor Apache y el sistema de gestión de base de datos relacional MySQL, por tanto, debemos funcionar con XAMPP. Las bases de datos deberán de estar normalizadas en, al menos, tercera forma normal. Su posible forma será:

- (\*)Tabla pedidos = {nombre, apellidos, email, nombreProducto (ya sea sidra, sidraEspumosa u otrosProducto), precio}
- (\*)Tabla comentario = {nombre, apellidos, email, mensaje}

(\*) -> hay que aclarar que como se muestran en los ejemplos del campus virtual antes citados, tendré que añadir un identificador a ambas tablas como clave primaria.

En cuanto a las credenciales citadas en el enunciado, serán:

USUARIO: DBUSER2018

Nombre del HOST: Local (localhost)

PASSWORD: DBPSWD2018

### ¡Corrección!:

Para completar este apartado y añadir el punto que he de corregir, conservare el archivo php pedido.php, pero en cuanto al archivo comentario.php lo suprimiré, mejor dicho, lo transformare en dos archivos php: usuarioContacto.php y comentarioContacto.php de esta forma no tendré que hacer cambios en otros puntos de la propuesta que tengo aprobados y me permitirá crear las dos tablas relacionadas que se exigen.

Crearé una base de datos llamada sidra, por ejemplo, que contenga dos tablas que estén relacionadas como serán las tablas Usuarios y Comentarios, también esta base de datos hará que sea también la correspondiente con la tabla Pedidos citada al principio del apartado de computación a través de PHP con el servidor, aunque no tenga relación ni temáticamente ni con ninguna clave con las tablas que describiré a continuación:

Usuarios:

UsuarioContacto_id	NombreCompleto	Email
...	...	...

La tabla Usuarios tendrá como llave primaria el identificador **UsuarioContacto\_id**, que nunca será null e ira incrementando automáticamente a medida que se le añadan más registros. El **NombreCompleto**, será una de las pequeñas transformaciones del archivo suprimido citado antes (comentario.php), en el que el cliente introducirá su nombre de pila y apellidos (VARCHAR), y por último el email del usuario **Email** (VARCHAR).

Comentarios:

Comentario_id	Asunto	Comentario	UsuarioContacto_id
...	...	...	...

La tabla Comentarios tendrá como llave primaria el identificador del comentario **Comentario\_id**, que como en el caso de la tabla Usuarios nunca será null e incrementará automáticamente, según se añadan más. El **Asunto** (VARCHAR), será un título descriptivo del motivo por el que el cliente quiere contactar vía-comentario con mi empresa. En cuanto al comentario, el cliente dejará sus reflexiones y cuestiones, consistirá en un VARCHAR también. Por último, y aquí es donde “señalaremos” la relación entre mis dos tablas, será **UsuarioContacto**, una clave foránea que hace referencia a la tabla Usuarios.

Las dos tablas pretenden guardar esa relación común de registro de usuario o cliente y su consiguiente comentario, serán creadas, se conectarán a la base de datos y procederán a insertar los datos del cliente a través de dos archivos php, anteriormente citados (usuarioContacto.php y comentarioContacto.php). Serán manejados en la vista de contacto.html, dónde el cliente a través de un formulario introducirá los datos requeridos para manejar esa computación con el servidor.



## Pruebas

Se debe comprobar que se realizan las pruebas siguientes:

- Funcionamiento en máquina local mediante XAMPP
- Funcionamiento en la nube con una cuenta de estudiante. Se debe documentar en la Bitácora
- Validación del cumplimiento de los estándares HTML5 y CSS3, sin errores ni advertencias, mediante los validadores del W3C
- Pruebas de usabilidad del proyecto con al menos 12 personas indicando su nivel de destreza en la web, en 3 tandas de 4 personas. Se debe explicar en la Bitácora las tareas solicitadas a los usuarios y los tiempos que tardan en realizarlas. Entre las tandas se debe especificar en la Bitácora los cambios realizados para mejorar la usabilidad.
- Pruebas de adaptabilidad usando al menos 3 herramientas. Se debe documentar en la Bitácora
- Pruebas de accesibilidad utilizando al menos 3 herramientas. Se debe documentar en la Bitácora