



PRÁCTICA 2: PROGRAMACIÓN DEL LADO DE CLIENTE HTML5/CSS3 - JAVASCRIPT

En esta práctica implementaremos la parte de cliente del sitio web. Para ello haremos uso de los lenguajes HTML5/CSS3 y JavaScript, del W3C (*World Wide Web Consortium*). Tomaremos como punto de partida los *mockups* de la Práctica 1, por lo que es conveniente que el equipo evalúe el alcance del trabajo antes de iniciar su desarrollo.

Dado que el desarrollo de la parte de servidor se llevará a cabo a partir de un sistema de plantillas, se deberá desarrollar cada *wireframe* de forma cuidadosa e incluyendo los comentarios necesarios para que, en la siguiente práctica, se tenga plenamente localizado y prototipado cada uno de los contenidos dinámicos que se van a incluir en la página. *Cada wireframe podrá dar lugar al desarrollo de una o más páginas* (denominadas **vistas**). Por ejemplo, el formulario de registro podrá implementar la vista de formulario de entrada de datos, la vista de registro-éxito y la vista de registro-fracaso. Se deben tener en cuenta todas las posibles vistas para cada una de las funcionalidades y acciones previstas para los *wireframes*. Así, a partir de la Arquitectura de la Información, se genera este modelo navegable como si del entorno real se tratara, de forma que el conjunto de vistas desarrollado podrá ser utilizado para demostración, evaluación de UX, promoción y presentación del sitio.

Obsérvese que el cliente deberá implementarse lo más grueso posible. No obstante, la funcionalidad del *mockup* es limitada y se basa fundamentalmente en implementar el aspecto que mostrarán los contenidos, su organización, etiquetado, etc. Su importancia es fundamental, ya que este diseño sirve a analistas y programadores web como base del posterior desarrollo y a los diseñadores para obtener el primer *feedback* del producto. Además, permite al usuario observar su navegabilidad tal y como lo haría una vez implementado. En la práctica, estas vistas deberán ser implementadas en HTML/CSS por cada equipo, sin utilizar ningún tipo de generador de código o asistente, y documentando el código. En el caso de HTML, se utilizarán siempre que sea posible el etiquetado semántico. Por otra parte, se debe tener implementadas en JavaScript todas las validaciones y funciones necesarias para aligerar las llamadas al servidor.

En esta fase se deben llevar a la práctica los puntos analizados en la AI, prestando especial interés a la navegabilidad, interacción, diseño web, estructura de los *layouts*, etc. Un fallo en cualquiera de estos puntos una vez el negocio está en línea puede causar un alto coste, tanto económico como en

falta de fidelización de usuarios. En esta fase, en cambio, ese mismo fallo puede ser fácilmente resuelto.

"Diseñar una interacción sin prototipar es como cocinar sin probar el guiso"
(Cr. Villamore)

En esta práctica se desarrollarán las vistas (parte de cliente) de nuestro sitio web mediante tecnologías HTML5 y CSS3. El primero de los lenguajes es utilizado para el desarrollo de páginas web, incluyendo contenido y estructura. El segundo lenguaje es utilizado para programar los aspectos de diseño gráfico separados del contenido y estructura. Ambos son estándares de la W3C, que pueden consultarse en la propia web de la organización de estandarización:

- <https://www.w3.org/TR/html52/>
- <https://www.w3.org/standards/webdesign/htmlcss>

Otra fuente habitual son los tutoriales de W3Schools:

- <http://www.w3schools.com/>
- https://www.w3schools.com/html/html5_intro.asp
- <https://www.w3schools.com/css/>

En el caso del lenguaje de programación JavaScript, hay fuentes de consulta igualmente contrastadas:

- <https://www.w3schools.com/js/>
- <https://developer.mozilla.org/es/docs/Web/JavaScript>

Para la práctica, se requiere que todas las validaciones posibles sean realizadas en la parte de cliente. Igualmente, cada equipo de prácticas debe implementar al menos 4 funciones de acceso y manipulación del DOM, de control de eventos del usuario (implementando la lógica de los mismos), e incorporar algunas lógicas en su página que requieran un desarrollo más avanzado en JavaScript (funciones, prototipos, etc.) que demuestre la comprensión y la capacidad de uso por parte del equipo de estos lenguajes. Se valorará la complejidad y originalidad de las soluciones aportadas con este lenguaje.

Cada equipo podrá hacer uso de llamadas a servicios de terceros (p.ej. Disqus, mailchimp, etc.), siempre y cuando estos servicios no interfieran en el objetivo principal de la práctica: aprender el desarrollo en HTML (semántico y estructural), CSS y Javascript.

Igualmente, recuerde las normas básicas de desarrollo (no limitadas a las siguientes):

- El documento HTML deberá estar estructurarse principalmente con etiquetas semánticas, siempre que sea posible.

- Tanto el/los documento/s CSS como el código JS será enlazado desde el documento HTML pero se situarán en archivos separados y correctamente estructurados y distribuidos en el sistema de ficheros del servidor web.
- En caso de compartir funciones JS desde más de un documento HTML, se deberán tener en cuenta aspectos de modularización y reutilización de código.
- Todo código debe estar correctamente documentado, incluyendo las cabeceras de los ficheros.
 - Los comentarios deben seguir preferente el formato de JSDoc: <https://jsdoc.app/about-getting-started.html>
 - En cualquier, los comentarios del código deberán incluir las cabeceras de los ficheros JS, en las que se deben indicar información sobre: desarrollador/es del fichero (no necesariamente todo el equipo), histórico de cambios en código, descripción del fichero, servicios de terceros que se incluyen y su motivación/uso.

Además de lo anterior, considérese que puede ser posible encontrar errores en la práctica 1 (entregada) o bien necesitar realizar variaciones en la misma debido a que no se midiera correctamente el alcance, no se detectaran correctamente los tipos de contenido (o su metainformación), etc. En estos casos, es posible realizar cambios en la práctica, siempre y cuando se tengan en cuenta los siguientes puntos:

- Cada cambio realizado deberá estar perfectamente documentado en la hoja de control de versiones que se debió haber incluido en el documento técnico.
- El PDF correspondiente a la práctica 1 debe subirse a Moodle junto con la práctica 2 (todo convenientemente recopilado en formato ZIP).

Tenga en cuenta los siguientes **aspectos de normativa** referentes a esta práctica:

- Junto con el código fuente, se presentará un informe de 4 páginas máximo en formato PDF, en el que el equipo explique los aspectos técnicos acerca de cómo han realizado el diseño de las páginas, los criterios prioritarios que han adoptado en el desarrollo, vinculen su desarrollo al temario de la asignatura, así como cualquier otro punto que deseen hacer constar para explicar el trabajo realizado por el equipo.

- El prototipo resultante será recopilado en una carpeta y comprimido en formato RAR o ZIP. Este documento, junto con el informe, se subirá a Moodle en la tarea dispuesta a tal fin.
- Se valorará fundamentalmente que el prototipo sea completo, así como la consistencia y coherencia respecto a la AI.
- Recuerde que el contenido de esta práctica podrá ser objeto de preguntas en el examen teórico.
- La fecha de entrega se dispondrá en la tarea Moodle correspondiente.
- El incumplimiento de este enunciado y su normativa puede implicar que la práctica no sea apta. Es responsabilidad única del estudiante su completa lectura y cumplimiento.

IMPORTANTE: No se admitirá ninguna entrega fuera de plazo o a través de un medio distinto a la tarea de Moodle (p.ej. envío por email).