

Práctica Evaluable I: Diseño del sitio web de una red social

Lothar Soto Palma

19 de abril de 2017

Índice

1. Introducción	2
2. Descripción de los elementos	2
2.1. Común a todos los HTML (Layout)	2
2.2. portada.HTML	3
2.3. pepe.HTML	4
2.4. fotosPepe.HTML	4
2.5. informaciónPepe.HTML	5
2.6. pepe_20170305_1533.HTML	5
2.7. informaciónNico.HTML	5
3. Decisiones de diseño	5
3.1. Organización de elementos	5
3.2. Decisiones sobre el aside	6
3.3. Decisiones sobre el main	7
4. Aspectos innovadores	7
4.1. HTML	7
4.2. CSS	7

1. Introducción

En este documento se tratará de explicar el porque de las decisiones tomadas sobre la maquetación de la página web y otras cuestiones de implementación con las etiquetas de HTML 5 y opciones de CSS 3.

2. Descripción de los elementos

En esta sección se va a responder a la pregunta de ¿Cómo se hizo?, vamos a analizar cada uno de los diferentes diseños implementados para la web, las decisiones de diseño se tratarán en la siguiente sección. En esta práctica se ha supuesto que hay 3 usuarios, Pepe que sería el usuario con sesión activo, Nico y Andrea que son los usuarios activos en el sistema.

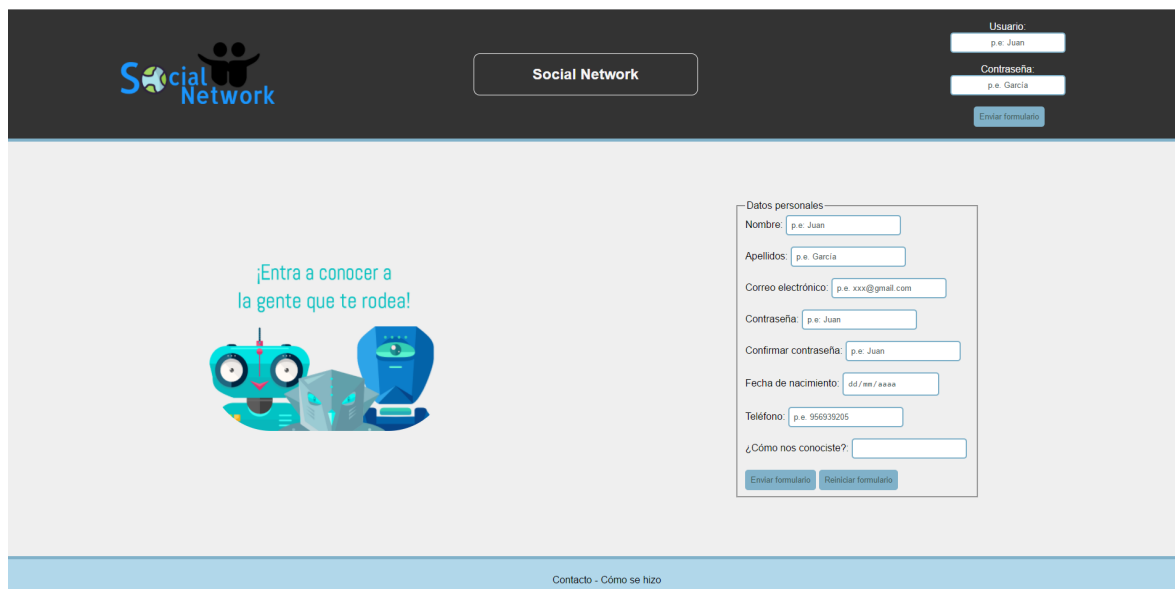
2.1. Común a todos los HTML (Layout)

Hay partes de la web que no parecen cambiar a medida que se lleva a cabo la navegación dentro de la web, por tanto se está repitiendo mucha cantidad de código pues todo es estático aún y nada se está generando con php. Estas partes serían la cabecera o header, la barra que muestra los usuarios del sistema y la barra lateral que muestra los usuarios activos en el sistema. El único archivo HTML que no posee alguna de estos tres elementos es el “*index.HTML*”.

- **Cabecera o header:** Se ha organizado la cabecera en dos secciones, la parte llamada “*header_section*” que incluye la imagen de la red social, el título de la red social y el logging de usuario o imagen del mismo si este está conectado, la segunda sección llamada “*navigation_bar*” que contiene la barra de navegación una lista con los elementos que nos va a permitir navegar por la web. En mi opinión lo mejor que se puede hacer para evitar usar la etiqueta “*div*” en el form de logging y el elemento de usuario conectado, para que el contenido quede organizado de la mejor forma posible es el uso de la etiqueta “*table*”, de forma que la imagen y el enlace “bye... ” que te devuelve al índice estén relacionados pues debería ser un único elemento. Otra decisión a destacar es el uso de cajas flexibles para organizar los elementos de la misma de forma automática.
- **Sección de usuarios del sistema:** La sección de usuarios del sistema no tiene misterio, la clase CSS se llama “*users*” e incluye a todos los usuarios registrados en la web. Esta sección puede deslizarse hacia abajo con la opción de CSS “*overflow-y: scroll*”, se ha optado por usar tablas para la organización de los nombres e imágenes y además para evitar el uso de “*div*”.
- **Aside de usuarios activos:** El aside es la barra lateral que se encuentra a la derecha en la web, se ha construido de la misma forma que el anterior, tiene cierta cuestión de diseño que podría arreglarse con JavaScript detallado en la siguiente sección.
- **Pie de Página o footer:** El pie de página tan solo contiene un menú con el contacto y un enlace a la documentación documentación, no ha habido problemas con este elemento ni casi decisiones más allá del típico ¿De qué color pongo el fondo?.

¿Cómo se hicieron cada una de las webs relevantes del sistema? ## index.HTML

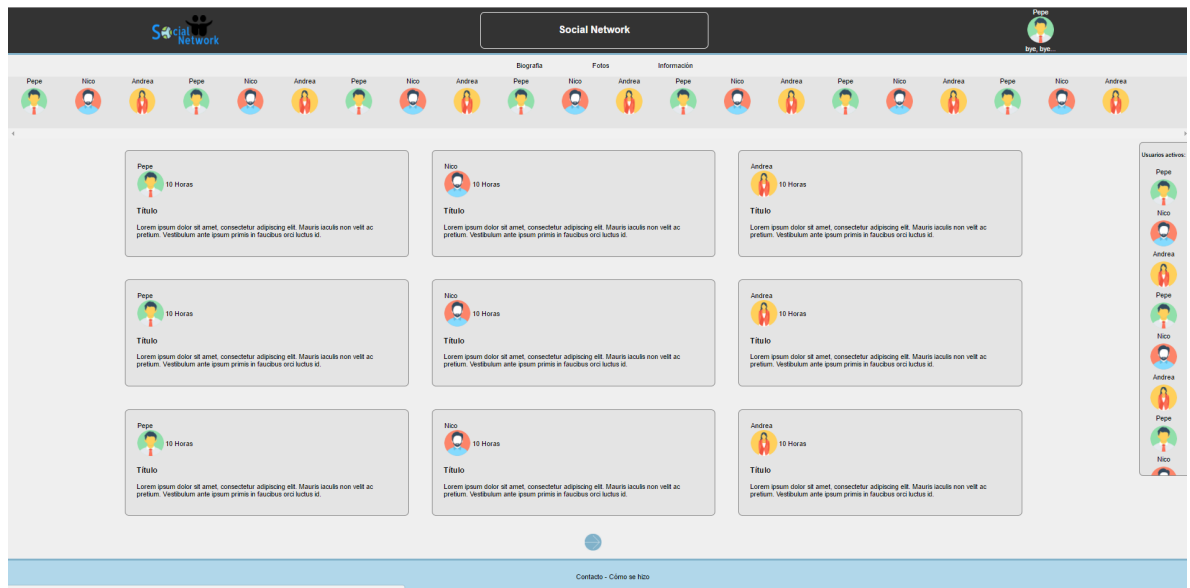
El index o índice fue lo primero que comenzó a construirse y diseñarse, prácticamente todo el diseño CSS posterior fue decidido en este momento, se comenzó por dar una estructura básica a la web HTML, comenzando por el header que se organizó de la forma anteriormente descrita y después se añadió una etiqueta “*main*” HTML para incluir el contenido relevante de la web en este caso el formulario de registro y una imagen. Una vez dada la estructura se mejoró el aspecto para ver como quedaría la primera versión del índice.



The screenshot displays the 'Social Network' index page. The header is dark grey with the 'Social Network' logo on the left, a 'Social Network' button in the center, and a login section on the right with fields for 'Usuario' (p.e. Juan) and 'Contraseña' (p.e. Garcia), and an 'Enviar formulario' button. The main content area is light grey and features a central illustration of two stylized robot heads with the text '¡Entra a conocer a la gente que te rodea!'. To the right of the illustration is a 'Datos personales' registration form with fields for Nombre, Apellidos, Correo electrónico, Contraseña, Confirmar contraseña, Fecha de nacimiento, and Teléfono, each with a placeholder example. There is also a '¿Cómo nos conociste?' field. At the bottom of the form are 'Enviar formulario' and 'Reiniciar formulario' buttons. A footer bar at the bottom contains the text 'Contacto - Cómo se hizo'.

2.2. portada.HTML

La portada fue quizás aquel documento con mayor trabajo ya que es un poco más complejo que el índice y provocó muchos de los cambios de diseño mencionados en la siguiente sección. Se compone por los tres elementos comunes antes mencionados pero ahora el contenido del “*main*” ha cambiado, ahora este elemento contiene dos secciones una primera llamada “*articles*” que va a contener todos los posts que los usuarios vayan realizando (uno por usuario aunque ahora mismo solo se añadieron tres por eso se repiten), y otra sección que contiene los elementos de paginación que puede ser que sean finalmente cambiados a una paginación numérica típica en función de como se desarrolle la siguiente práctica.



En la previsualización de la web la web está reducida. como se especificaba que este documento debía ser responsive, las decisiones que se han tomado con respecto a ello fueron poner cada artículo de uno en uno listados en columna, hacer el “*aside*” y la sección “*users*” desplegable, además de cambios en el menú para que se apilen también en columna. Este cambio del CSS se produce cuando la web se reduce hasta los 650px de ancho.

Un elemento a destacar del del diseño responsive es que se ha eliminado el aside con los usuarios, pero se ha incluido un elemento que cuando se pasa el raton por encima del mismo hace aparecer el aside, y este se mantiene hasta que dejas de tener el ratón encima del mismo. Esto se ha hecho con opciones CSS que permiten superponer elementos como “*z-index*”.

2.3. pepe.HTML

Una vez realizada la portada el resto de elementos de las webs restantes fue copiar y pegar elementos que ya tenía y añadir algunos nuevos que incorporarán funcionalidades posteriormente, en este caso tan solo con cambiar unos enlaces e imágenes podíamos obtener la vista que Pepe nuestro usuario activo vería si estuviera conectado en el sistema. El cambio más notable fue añadir una nueva sección que contiene un formulario con un campo de texto para subir imágenes y escribir los posts.

2.4. fotosPepe.HTML

En esta web parte también de la portada se eliminan los artículos y se disponen las imágenes de todos los post de Pepe vinculados a cada uno de los mismos. Tan solo fue necesario cambiar el contenido del “*main*” y darle una clase propia para ajustar las imágenes llamada “*photos*”, aunque se permiten imágenes de diferente tamaño.

2.5. informaciónPepe.HTML

La web de información tampoco resultó muy problemática, en este caso como se trata del usuario activo se añadió un registro que te permite cambiar tu imagen, tu información personal, contraseña. También parte de la portada, pero tiene diferencias si se trata de un usuario del sistema y no el activo, los cambios se comentan en la sección “*informaciónNico.HTML*”.

2.6. pepe_20170305_1533.HTML

Este archivo trajo algún problema extraño con el CSS y la organización de los elementos, fue posible resolverlo simplificando un poco las opciones del archivo CSS, pero en general lo que se hizo en esta vista fue tonar la portada, eliminar el contenido del “*main*” y añadir tres nuevas secciones que son respectivamente, el contenido el post del usuario, los comentarios del mismo y una sección que permitirá añadir tus propios comentarios. De forma obvia los comentarios son considerados como artículos dentro de la sección.

2.7. informaciónNico.HTML

El procedimiento fue el mismo usado en “*informaciónPepe.HTML*” salvo que ahora no se permite cambiar la indormación del usuario del cual miras los datos.

3. Decisiones de diseño

3.1. Organización de elementos

En un primer momento surge la necesidad de dar una organización de los elementos web ordenada y que se puede modificar de forma sencilla, en un primer lugar se pensó en crear haciendo uso de CSS una serie de clases que sirvieran como la típica combinación de contenedor/fila/columna típica de los frameworks actuales como por ejemplo bootstrap, sin embargo esa forma hacia que a nivel de HTML se tuvieran que introducir niveles de etiquetas que no tendrían sentido lógico con las etiquetas de HTML 5, en otras palabras tendríamos que usar la etiqueta div para evitar sections dentro de sections.

Finalmente se decidió hacer uso de una opción de CSS 3 llamada grid, que permite organizar la pantalla creando una malla a través de la que se colocarán los elementos en función de como se vayan configurando, como ejemplo se incluirá un ejemplo del código:

```
.container {  
  display: grid;  
  grid-template-columns: auto 150px;  
  grid-template-rows: auto 150px auto auto;  
}
```

header,

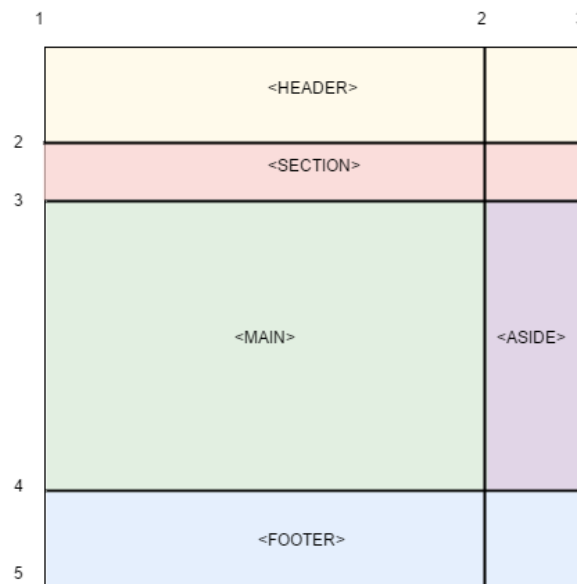
```

footer {
  grid-column-start: 1;
  grid-column-end: 3;
}

```

La clase de CSS *container* hace que el lo que tenga dentro del elemento HTML se organice de la forma descrita inmediatamente después, con “*display: grid;*” estamos indicando el tipo de caja que se usará el elemento HTML, luego “*grid-template-columns: auto 150px;*” sirve para configurar como serán las columnas de la malla, en este caso hay dos columnas la primera tiene un tamaño automático y la segunda solo tiene 150 pixeles de ancho, por último “*grid-template-rows: auto 150px auto auto;*” sirve para configurar las filas de la malla, en este caso hay 4 filas todas con un tamaño automático exceptuando la segunda que tiene 150 pixeles de largo.

Ahora bien si queremos que un elemento HTML ocupe el 100 % de la pantalla a lo ancho es decir (width: 100 %) tenemos que indicar en la hoja de estilo correspondiente las columnas y filas que estas ocuparan, por ejemplo en este caso se usa que el header y el footer van a ocupar toda la pantalla a lo ancho por lo tanto como tenemos dos columnas y tras índices en la malla, el elemento comenzará desde el primer índice hasta el último, si solo va a ocupar una única fila o columna puede usarse la orden “*grid-column: n*” donde n es el número de la columna. Para indicar las filas se hace de forma análoga.



Por otro lado parece que esta opción no está soportada por algunos navegadores como safari, aunque si para google chrome.

3.2. Decisiones sobre el aside

En el aside de la web se ha incluido la parte que informa al usuario de que otros usuarios están activos en el sistema, esta parte es un poco problemática puesto que no se ha encontrado la forma de hacer

solo con CSS y HTML que la altura del elemento HTML sea la misma en todo momento que la del contenido principal o main (para que cuando se añadan elementos al HTML el aside se vaya ajustando), esto es debido a que usa la opción “*overflow.y: scroll;*” para incluir una barra de desplazamiento cuando los elementos dentro del aside superen la altura máxima. Se han probado múltiples cosas para intentar arreglarlo pero en todas hacían que el elemento tomara el tamaño máximo dado por los elementos dentro del aside. Por último la decisión final tomada sobre este elemento fue que tuviera una altura fija proporcional a la pantalla pero que cuando se incrementa esta altura no cambie, posteriormente se puede añadir la opción de que el elemento siga la vista del usuario al desplazarse por la web.

3.3. Decisiones sobre el main

En un primer lugar se trataba de alinear los elementos usando “*float: left;*” de forma que cada uno se fuese apilando por la derecha, pero se optó por usar la opción “*display: flex;*” y “*flex-wrap: wrap;*” que nos va a permitir que los elementos se centren y estén separados de igual forma automáticamente, además apila los elementos de una forma más vistosa cuando falta alguno de los artículos de su interior.

4. Aspectos innovadores

4.1. HTML

- Uso de la etiqueta “**main**” para incluir el contenido relevante o que va a cambiar entre las webs.

4.2. CSS

Opciones innovadoras:

- **border-radius:** Esta opción sirve para que el borde que tenga el elemento sea redondeado, se puede usar con % para obtener circunferencias que a veces vienen bien.
- **display: flex:** Esta opción es un tipo de caja HTML que permite que el objeto al que se aplique sea “flexible” además tiene buen tratamiento con los elementos de su interior y permite modificar su posición de forma sencilla.
- **align-items: center:** Alinea los elementos hijos de un elemento padre en el centro pero con respecto al eje y.
- **justify-content: flex-start:** Especifica el como debe estar alineado los elementos de una caja flexible.
- **transition: all 0.5s ease 0s:** Es una opción que sirve más bien para el diseño, da una transición entre el estilo del elemento en cuestión y el estilo cuando es seleccionado o se encuentra debajo del ratón.
- **flex-wrap: wrap:** Sirve para crear una malla sencilla dentro de una caja flexible, nos va a permitir organizar con mayor facilidad los artículos.

- **filter: saturate(2):** Sirve para saturar los colores de un elemento, en este caso se ha usado para las flechas de la paginación.
- **display: grid:** Se trata de la opción de tipo de caja que se usa de forma exhaustiva en la estructura de la web.
- **z-index: 1** Esta opción sirve para superponer elementos HTML.