## 

**FACULTAD DE CIENCIAS E INGENIERÍA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**TEMA:**

HTML5

**CURSO:**

Algorítmica II

**PROFESOR:**

Coronel Castillo Eric Gustavo

**AUTORES:**

Clemente Mondragón Josías Código: 16101030

Holguín Cueva Pedro Código: 16101052

Cruz Acuña Jeferson Código: 16201017

Angie M. Reyes Calderón Código: 16201034

**FECHA DE ENTREGA:**

13-10-2017

**Los Olivos – Perú**

**2017-II**

**INDICE**

1. **RESUMEN**
2. **HISTORIA**
3. **FUNDAMENTOS DE LA INVESTIGACION** 
   1. **Identificación del problema**
   2. **Problema**
   3. **Planteamiento de solución**
   4. **Objetivos**
4. **DESARROLLO DEL TEMA**
   1. **Definición**
   2. **Etiquetas**
5. **CASOS**
6. **CONCLUCIONES**
7. **RECOMENDACIONES**
8. **BIBLIOGRAFIA**

**RESUMEN**

**HISTORIA**

El lenguaje HTML fue desarrollado en año 1989 por Tim Berners-Lee, y a partir de ese momento y hasta la actualidad se han ido incorporando nuevas especificaciones y etiquetas para adaptarlo a los requerimientos de los usuarios, así como a la propia evolución técnica de Internet y a la evolución conceptual de la Web (Terán, 2010).

Para marcar las directrices de estandarización del desarrollo del WWW se creó un consorcio internacional denominado W3C que está formado por diferentes instituciones internacionales de investigación y educación y grandes empresas vinculadas con la informática y las tecnologías de Internet. A cada desarrollo y ampliación importante se le asignó un número de versión. Las versiones existentes de HTML, y las características principales que incorporan, son las siguientes:

- HTML 0: Establece la estructura básica de un documento en Internet.

- HTML 1: Añade estilos para el texto.

- HTML 1.0: Reúne en una sola versión las dos anteriores.

- HTML 2.0: Añade formularios.

- HTML 3.0: Añade tablas.

- HTML 3.2: Incluye desarrollos de Netscape.

- HTML 4.0: Incluye hojas de estilo, JavaScript, marcos, objetos multimedia, mejoras en la accesibilidad, etc.

- HTML 5.0: Incluye gestión directa de vídeo además de otros avances.

Podría decirse que **HTML5** nació en 2004, cuando se fundó el grupo de trabajo**WHAT**con miembros de **Apple,** la **Fundación Mozilla** y **Opera Software**.

Dos años después consiguieron uno de sus principales apoyos, pues el **W3C** (World Wide Web Consortium) decidió abandonar **XHTML** y comenzó a trabajar con WHAT.

En el año 2008 fue un gran año para este lenguaje, pues **se finalizó su primera versión**.

Esto permitió que **Firefox 3**, el navegador de Mozilla fuese compatible con HTML5. Más adelante se sumarían **Internet Explorer**, **Google Chrome** y **Safari**.

Poco a poco se fueron sumando compañías y se fue abandonando el uso de tecnologías como Flash. De hecho, en septiembre de 2011, **34 de las 100 páginas con más tráfico según el ranking de Alexa utilizaban este lenguaje**.

**La evolución del HTML**

El lenguaje HTML es el resultado del trabajo de W3C, estos trabajos de investigación pasan por muchas etapas que han ido evolucionando con el tiempo. En la actualidad, los resultados de las investigaciones se publican en tres etapas según (AUBRY, 2017):

* Los borradores, working drafts, se publican para compartir los avances de las investigaciones con los demás.
* Las especificaciones se publican cuando el lenguaje está casi terminado y listo para usar.
* Las recomendaciones una vez que el lenguaje esta oficialmente aprobado y terminado.

**El Hyper Text Markup Language**

Fue en 1991 cuando Tim Berners-Lee redacto los primeros “bocetos” (draft) del HTML.

En junio de 1993, aparece el primer documento técnico que describe el lenguaje HTML: “Hypertext Markup Language (HTML) – A Representation of Textual Information MetaInformation for Retrieval and Interchange”.

El 8 de noviembre de 1993, aparece el HTML+, la versión 2 del HTML ve la luz el 22 de 1995 bajo los auspicios del IETF (Internet Engineering Task Force).

En marzo de 1995 se publica el HTML 3, que se presenta como una extensión del HTML 2. Rápidamente es reemplazado por la recomendación del HTML3.2, el 14 de enero de 1997.

La recomendación del HTML 4.01 se publica el 24 de diciembre de 1999. Para W3C, esta versión es la última del HTML. El W3C piensa que el futuro de las páginas web no está en HTML sino en el XML. El HTML “ha muerto” para W3C (AUBRY, 2017).

**La evolución con el XHTML**

El lenguaje HTML conlleva limitaciones intrínsecas, (AUBRY, 2017):

* Los elementos que se pueden usar se limitan a los indicados en la recomendación.
* El lenguaje es muy permisivo: es posible indicar el nombre de los elementos en mayúsculas o minúsculas, algunas etiquetas de cierre son facultativas.
* Los elementos no son realmente semánticos: el elemento <p> puede contener cualquier tipo de texto.

El W3C publica el 10 de febrero la recomendación de un nuevo lenguaje, el XML. La quinta edición se remonta el 26 de noviembre de 2008. Este lenguaje permite superar todas las limitaciones del HTML. De este modo es posible crear todos los elementos que queramos de forma totalmente semántica (por ejemplo un elemento <cp> para contener un código postal o un elemento <precio> para incluir un precio), pero la sintaxis es muy estricta. Se trata de un lenguaje ideal, en especial, para cambiar datos estructurados en la web. Pero XML interviene además en muchos otros lenguajes estándares del W3C (MathML, SVG…) y en otras tecnologías web.

Debido a su sintaxis permisiva, el HTML en incompatible con el XML, Para resolver esta incompatibilidad, el W3C formulo la sintaxis del HTML de tal manera que fuera compatible con XML: Se trata del XTHML. La primera recomendación sale a la luz el 26 de enero de 2000.

El XHTML presenta una sintaxis más estricta, como podemos ver en estos dos ejemplos:

Si se abre un elemento, deberá incluirse la etiqueta de cierre:

* Sintaxis correcta en HTML: <p>Mi texto
* Sintaxis correcta en XHTML: <p> Mi texto </p>

El XHTML prevé el cierre de los elementos que no dispongan de una etiqueta de cierre:

* Sintaxis correcta en HTML: <br>
* Sintaxis correcta en XHTML: <br/>

Además, los elementos XHTML deben incluirse en minúsculas y los valores de los atributos deben estar entre comillas.

* Sintaxis correcta en HTML: <P ID= intro>
* Sintaxis correcta en XHTML: <P id=”intro”>

El W3C publica la recomendación del XHTML 1.1 el 31 de mayo de 2001. La última edición se remonta a noviembre del 2010, Esta versión abordaba el “auténtico XML” sin elementos incorrectos o que no se adaptaran a los estándares. Esto quiere decir que los documentos para la web ya no podían usar el tipo MIME text/html.

El XHTML 2 no aparece como Working Draft hasta agosto de 2002. Esta debería ser una versión de lenguaje “puro” sin ningún compromiso con el pasado, ¡hasta el punto que era incompatible con el contenido web existente! Se trató de una de una iniciativa demasiada radical. El W3C proponía un lenguaje totalmente desconectado de la realidad, hecho para los informáticos y no para los diseñadores web. Al optar por la incompatibilidad con lo existente, el W3C cometió un grave error estratégico.

La recomendación del XHTML 2 no llego a publicarse. El 17 de diciembre de 2010, el W3C disuelve oficialmente ese grupo de trabajo (AUBRY, 2017).

**FUNDAMENTOS DE LA INVESTIGACION**

**Identificación del problema**

**Problema**

**Planteamiento de solución**

**Objetivos**

* + - Mostrar una visión general de internet y los servicios que ofrece.
    - Conocer el servicio web para optimizar su funcionamiento.
    - Entender el funcionamiento de un lenguaje con marcas como HTML.
    - Conocer la evolución y tendencias futuras del lenguaje HTML.
    - Conocer lo posibilidad de crear documentos para la web utilizando solo código HTML.

**DESARROLLO DE TEMA**

**Componentes básicos**

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APls (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS, y JavaScript. Estas tecnologías son dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio.

Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero como esa estructura es construida. Crear fundamentos fuertes nos ayudara más adelante a aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías (Gauchat, 2012).

**HTML**

**Definición**

El lenguaje HTML funciona a través de marcas de sentido llamadas etiquetas. Las etiquetas son la herramienta fundamental para que los navegadores puedan interpretar el código y permitirnos ver imágenes, texto, párrafo, y estructuras. Los navegadores vendrían a ser como “traductores” de las etiquetas, usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas son palabras claves y atributos rodeados de los signos mayor y menor (por ejemplo, <html lang = “es”>). En este caso, html es la palabra clave y lang es el atributo con el valor es. La mayoría de las etiquetas HTML se utilizan en pares, una etiqueta de apertura y una de cierre, y el contenido se declara entre ellas. En nuestro ejemplo, <html lang = “es”> indica el comienzo del código HTML y </html> indica el final. Compare las etiquetas de apertura y cierre y vera que la de cierre se distingue por una barra invertida antes de la palabra clave (por ejemplo, </html>). El resto de nuestro código será insertado entre estas dos etiquetas: <html>. . . .</html> (Gauchat, 2012).

**HTML5,** es extremadamente flexible en cuanto a la estructura y a los elementos utilizados para construirla. El elemento <html> puede ser incluido sin ningún atributo o incluso ignorado completamente. Con el propósito de preservar la compatibilidad (Gauchat, 2012).

**Etiquetas**

Las etiquetas HTML son fragmentos de texto rodeados por corchetes angulares < >, que tienen funciones y usos específicos y se utilizan para escribir código HTML. Las etiquetas o tags son la forma de escribir código HTML.

En HTML existen etiquetas de apertura y etiquetas de cierre, tienen la forma: <etiqueta> </etiqueta>. Donde <etiqueta> es la etiqueta de apertura y </etiqueta> es la etiqueta de cierre indicada por la diagonal. HTML tiene definidas gran variedad de etiquetas para distintos usos.

**Concepto de etiquetas en HTML5**, en HTML5 se agregan nuevas etiquetas para utilizar que nos ahorran el uso de otros productos que se usaban para complementar y hacer cosas que con el simple HTML no se podían hacer. HTML5 fue creado para hacer que el proceso de escribir el código sea más simple y más lógico, por decirlo de una forma. La sintaxis de HTML5 se destaca, como dijimos, en el ámbito multimedia, pero son bastantes las etiquetas introducidas para generar una mejoría.

La idea detrás de HTML5 es que podamos visualizar el contenido multimedia variado que podemos encontrar en internet aun cuando nos encontramos en dispositivos de gama baja que no podrían soportarlo cuando tienen que instalar infinidad de plugin. No solamente contamos con etiquetas especiales como audio, video y canvas, sino también integración con contenidos de gráficos en vectores (que anteriormente se conocía como la etiqueta object. Con estas etiquetas, los usuarios pueden consumir videos y canciones, por ejemplo, sin necesidad de instalar nada de forma adicional.

A continuación haremos mención de la gran variedad de etiquetas que hay en HTML junto a una comparación con el HTML5.

**Estructura global**

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento esta diferenciada, declarada y determinada por etiquetas específicas. Se podrá apreciar cómo construir la estructura global de un documento HTML y los nuevos elementos semánticos incorporados en HTML5 (Gauchat, 2012).

**<! DOCTYPE>**

En primer lugar necesitamos indicar el tipo de documento que estamos creando. Esto es HTML5 es extremadamente sencillo:

<! DOCTYPE html>.

**<html>**

Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento <html>. Este elemento envolverá al resto del código:

<! DOCTYPE html>

<html lang=”es”>

</html>

El atributo lang en la etiqueta de apertura <html> es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso es por español.

**<head>**

Continuemos construyendo nuestra plantilla. El código HTML insertado entre las etiquetas <html> tiene que ser dividido entre dos secciones principales. A l igual que es versiones previas de HTML, la primera sección es la cabecera y la segunda es el cuerpo. El siguiente paso, por lo tanto, será crear estas dos secciones en el código usando los elementos <head> y <body> ya conocidos.

El elemento <head> va primero, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre:

<! DOCTYPE html>

<html lang=”es”>

<head>

</head>

</html>

La etiqueta no cambio desde versiones anteriores y su propósito sigue siendo exactamente el mismo. Dentro de las etiquetas <head> definiremos el título de nuestra página web, declararemos el set de caracteres correspondiente, proveeremos información general acerca del documento e incorporaremos los archivos externos con estilos, códigos JavaScript incluso imágenes necesarios para generar la página en la pantalla. Excepto por el título y algunos iconos, el resto de la información incorporada en el documento entre estas etiquetas es invisible para el usuario.

**<body>**

La siguiente gran sección que es parte principal de la organización de un documento HTML es el cuerpo. El cuerpo representa la parte visible de todo documento y es especificado entre etiquetas <body>. Estas etiquetas tampoco han cambiado en relación con versiones previas del HTML:

<! DOCTYPE html>

<html lang=”es”>

<head>

<body>

</body>

</head>

</html>

**<link>**

Otro importante elemento que va dentro de la cabecera del documento es <link>. Este elemento es usado para incorporar estilos, códigos JavaScript, imágenes o iconos desde archivos externos. Uno de los usos más comunes para <link> es la incorporación de archivos con estilos CSS:

<! DOCTYPE html>

<html lang=”es”>

<head>

<meta charset = “iso-8859-1”>

<meta name= “description” content=”Ejemplo de HTML5”>

<meta name= “keywords” content=”HTML5, CSS3, JavaScript”>

<title>Este texto es el título del documento</title>

<link rel=”stylesheet” href =”misestilos.css”>

</head>

<body>

</body>

</html>

En HTML5 ya no se necesita especificar qué tipo de estilos estamos insertando, por lo que el atributo type fue eliminado. Solo necesitamos dos atributos para incorporar nuestro archivo de estilos: rel y href. El atributo rel significa “relación” y es acerca de la relación entre el documento y el archivo que estamos incorporando por medio de href. En el caso, el atributo rel tiene el valor stylesheet que le dice al navegador que el archivo misestilos.css es un archivo CSS con estilos requeridos para presentar la página en pantalla. Con esta última inserción podemos considerar finalizado nuestro trabajo en la cabecera. Ahora es tiempo de trabajar en el cuerpo, donde ocurre la magia.

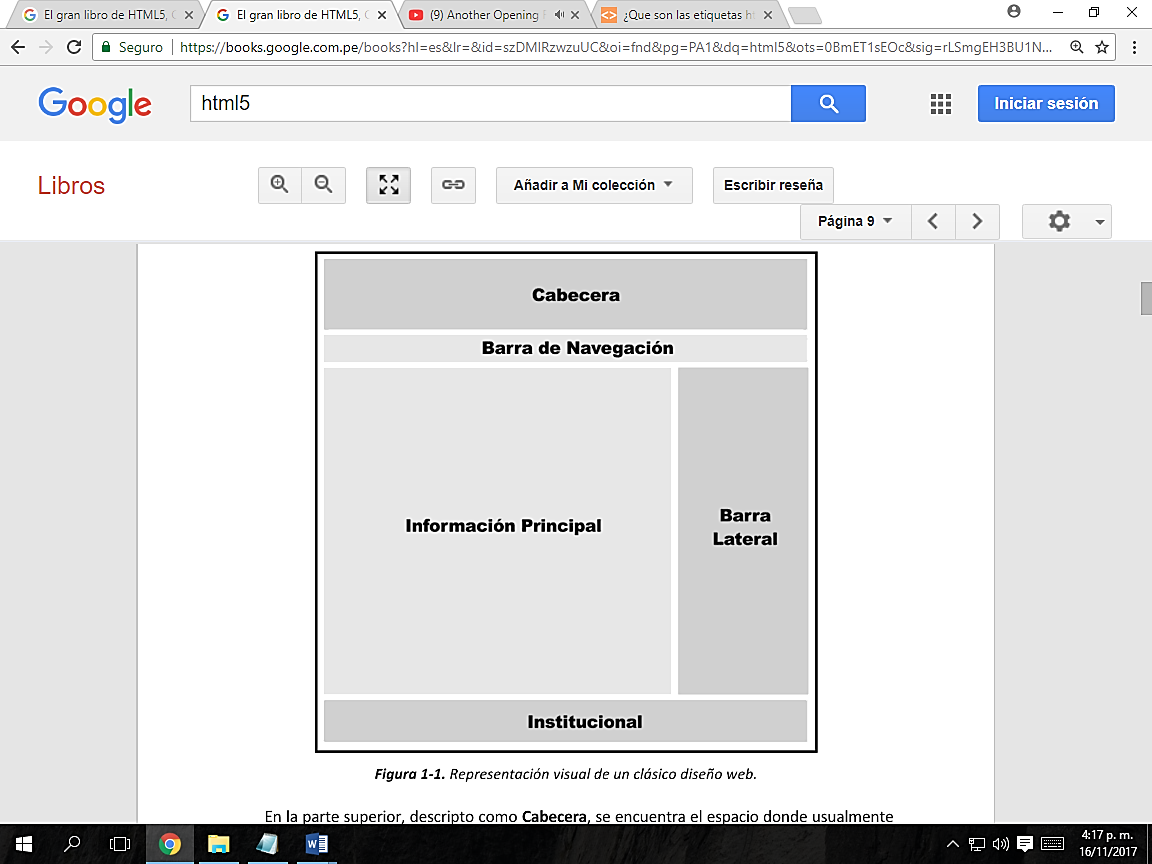
**Estructura global**

La estructura del cuerpo (el código entre las etiquetas <body>) generara la parte visible del documento. Este es el código que producirá nuestra página web.

**Organización**

La siguiente figura representa un diseño común encontrado en la mayoría de los sitios webs estos días. A pesar del hecho de que cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en cada sitio web estudiado:

1. Cabecera
2. Barra de Navegación
3. Sección de Información
4. Barra Lateral
5. El pie o la barra Institucional



**Figura 1- Representación visual de un clásico diseño web**

**<header>**

El elemento de HTML Header (<header>) representa un grupo de ayudas introductorias o de navegación. Puede contener algunos elementos de encabezado, pero también otros elementos como un logo, una sección que aglutine secciones de encabezados, una formulario de búsqueda o cosas parecidas (teoli, 2014).

**Sintaxis:**

<header>

<!-- Código HTML -->

</header>

**<nav>**

Esta etiqueta permite definir un elemento navegacional de la página web, como puede ser el menú principal o el menú secundario. Se debe utilizar para marcar los elementos principales de navegación, no se deben marcar todos los elementos. En conclusión crea un menú de navegación (Mora, 2011).

**Sintaxis:**

<nav>

<ul>

<li><a href="DireccionPagina"> Item de Navegación 1 </li>

<li><a href="DireccionPagina"> Item de Navegación 2 </li>

<li><a href="DireccionPagina"> Item de Navegación Etc </li>

</ul>

</nav>

**<section>**

El elemento de HTML section (<section>) representa una sección genérica de un documento. Sirve para determinar qué contenido corresponde a qué parte de un esquema. Piensa en el esquema como en el índice de contenido de un libro; un tema común y subsecciones relacionadas.  Es, por lo tanto, una etiqueta semántica. Su funcionalidad principal es estructurar semánticamente un documento a la hora de ser representado por parte de un agente usuario. Por ejemplo, un agente de usuario que represente el documento en voz, podría exponer al usuario el índice de contenido por niveles para navegar rápidamente por las distintas partes.

**Sintaxis:**

<section>

<h1>Encabezado</h1>

<p>Un montón de contenido impresionante.</p>

</section>

**<aside>**

El elemento HTML Aside (<aside>) representa una sección de una página que consiste en contenido que está tangencialmente relacionado con el contenido que le rodea, que podría ser considerado independiente de ese contenido. Estas secciones son a menudo representadas como barras laterales o como inserciones y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente, como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados.

**Sintaxis:**

<aside>

<div class="block">

<h3>Titulo</h3>

<ul>

<li>Opciones</li>

</ul>

</div>

</aside>

**<footer>**

El Elemento HTML Footer (<footer>) representa un pie de página para el contenido de sección más cercano o el elemento raíz de sección (p.e, su ancestro más cercano <article>, <aside>, <nav>, <section>, <blockquote>, <body>, <details>, <fieldset>, <figure>, <td>). Un pie de página típicamente contiene información acerca del autor de la sección, datos de derechos de autor o enlaces a documentos relacionados.

**Sintaxis:**

<footer>

<address>

Jhonny Doe<br>

jhonnydoe5468@mypc.com

</address>

<ul>

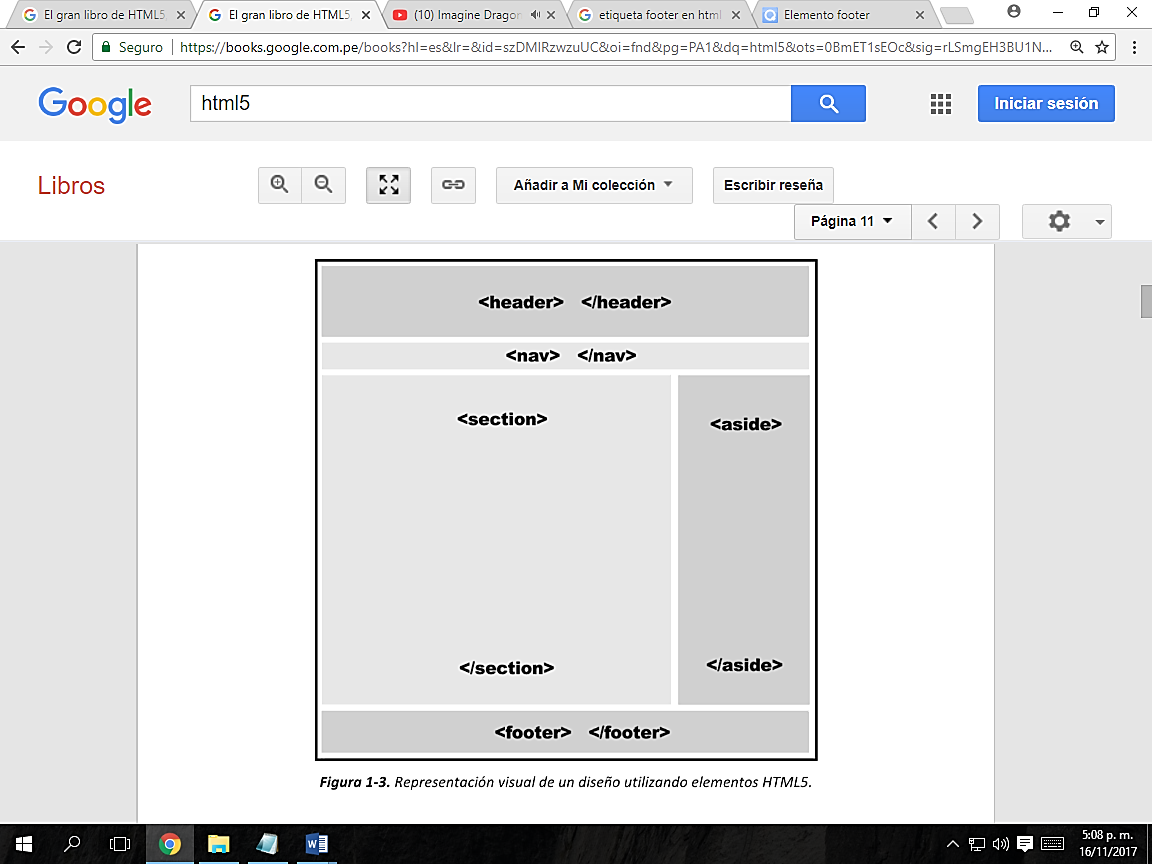
<li><a href="/">Home</a></li>

<li><a href="/food">Food</a></li>

<li><a href="/drinks">Drinks</a></li>

</ul>

</footer>



**Figura 2- Representación visual de un diseño web utilizando elementos HTML5**

**CSS y HTML**

Como aclaráramos anteriormente, la nueva especiación de HTML (HTML5) no describe solo los nuevos elementos HTML o el lenguaje mismo. La web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. En este nuevo paradigma, HTML se presenta junto con CSS y JavaScript como un único instrumento integrado.

La función de cada uno como ya se ha explicado, así como las etiquetas de HTML responsables de la estructura del documento. Ahora es momento de analizar CSS, su relevancia dentro de esta unión estratégica y su influencia sobre la presentación de documentos HTML.

Oficialmente CSS nada tiene que ver con HTML5, no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML provenían estilos esenciales para cada elemento, pero a medida que el lenguaje evoluciono, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo satisfacer la demanda de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado el proceso de evolución de HTML.

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías totalmente separadas (AUBRY, 2017).

**La Relevancia de JavaScript**

HTML5 puede ser imaginado como un edificio soportado por tres grandes columnas: HTML, CSS y JavaScript. Ya hemos visto los elementos de HTML y la importancia de CSS que la hace una herramienta ideal para diseñadores. Ahora es momento de develar lo que puede ser considerado como uno de los pilares más fuertes de esta especiación: JavaScript.

JavaScript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudo a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandiendo en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APls) fueron incorporadas por defecto en cada navegador para asistir el lenguaje en funciones elementales. Estas nuevas APls (como web Storage, Canvas y otras) son interfaces a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web (AUBRY, 2017).

**Bootstrap**

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

Las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso (Solis, 2014).

**Características principales de Bootstrap**

Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que nos permiten integrar el framework de forma sencilla y potente en nuestros proyectos webs (Rodriguez, 2012).

* Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
* Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
* Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
* Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
* Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.
* Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

**CASOS**

**CONCLUSIONES**

**RECOMENDACIONES**

**BIBLIOGRAFIA**

AUBRY, C. (2017). *HTML5 y CSS3 revolucione el diseño de sus sitios web* (3 edicion ed.). Barcelona, España: Ediciones ENI.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3, y Javascript.* Barcelona: marcombo.

Mora, S. L. (3 de marzo de 2011). *Blogger*. Obtenido de Blogger: http://accesibilidadenlaweb.blogspot.pe/2011/03/la-etiqueta-de-html5.html

teoli. (2 de Julio de 2014). *MDN web docs*. Obtenido de MDN web docs: https://developer.mozilla.org/es/docs/Web/HTML/Elemento/header

Terán, A. J. (2010). MANUAL INTRODUCCIÓN AL LENGUAJE HTML. *editorialCEP*, 66.

Rodriguez, T. (16 de Junio de 2012). *GENBETA:dev*. Obtenido de GENBETA:dev: https://www.genbetadev.com/frameworks/bootstrap

Solis, J. (26 de 09 de 2014). *ARWEB*. Obtenido de ARWEB: https://www.arweb.com/chucherias/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/