

2

HTML5 y CSS3

Visual Studio 2015 Web Developer

Objetivos

Al finalizar el capítulo, el alumno logrará:

- Aplicar las características de HTML 5.
- Implementar hojas de estilo haciendo uso de SASS.



Agenda

- HTML5
- Estructuras y esquemas
- Nuevos inputs, selectores, elementos
- CSS3 con SASS
- Canvas y SVG
- Audio y video
- Geolocalización



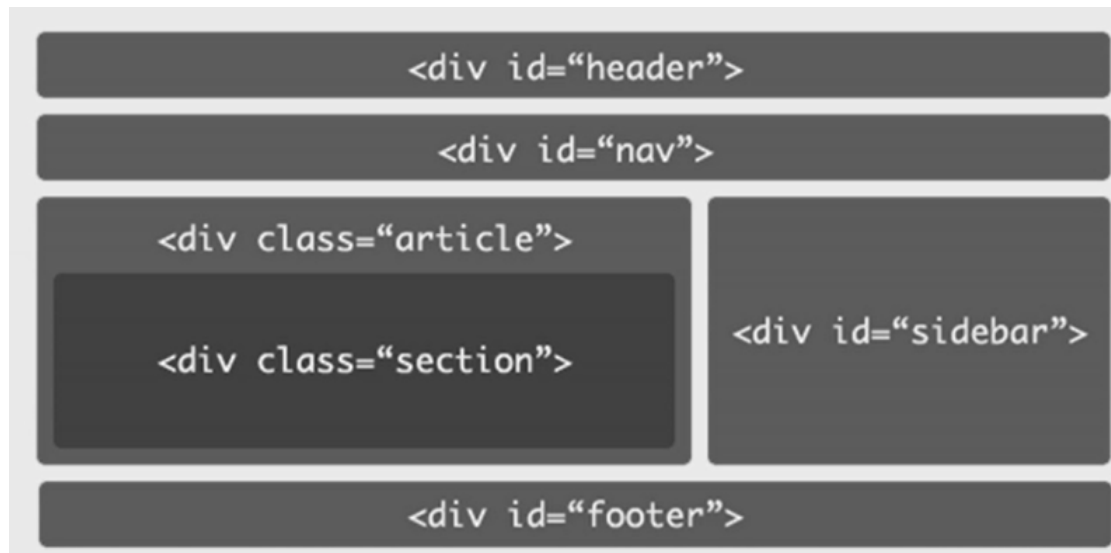
HTML5

- HTML5 es la nueva versión del Lenguaje de Marcado de Hipertexto, y cambia los paradigmas de desarrollo y diseño web al introducir herramientas notables como etiquetas que permiten la publicación de archivos de audio y video con soportes de distintos codecs, etiquetas para contenidos en 2D y 3D, cambios en los formularios y una web semántica mucho mejor aprovechada.
- Evolución de la web: <http://evolutionofweb.appspot.com>



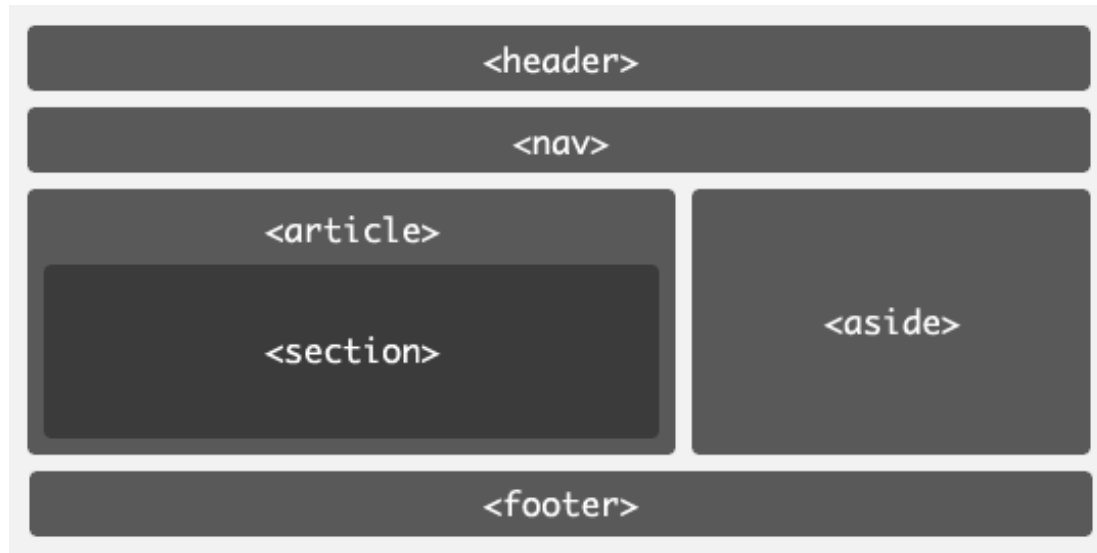
Estructuras y Esquemas

- Estructura típica de una página HTML:



Estructuras y Esquemas

- Con HTML5, se pueden aprovechar los tags semánticos:



Estructuras y Esquemas

- Tags semánticos de HTML5:

| Element | Level | Purpose |
|--------------|-------|-----------------------------------------------------------------|
| <article> | Block | Independent content such as a blog post or article |
| <aside> | Block | Content slightly related to primary content on page |
| <figure> | Block | Grouping stand-alone content, such as video or image |
| <figcaption> | Text | For use with <figure>, (optionally) used to provide caption |
| <footer> | Block | Providing author, copyright data, etc. |
| <header> | Block | Introductory headings, could include navigation |
| <hgroup> | Block | For grouping <h1> to <h6> |
| <nav> | Block | Navigation – typically site level |
| <mark> | Text | Text to be referenced or highlighted |
| <section> | Block | Grouping of content usually with a heading, similar to chapters |
| <time> | Text | For date and/or time representation |



Nuevos Inputs, Atributos, Elementos

- Nuevos Inputs HTML5:

Algunos muestran directamente algún contenido sobre el navegador y otros son para realizar validación. En caso que el navegador no soporte algún Input, simplemente se mostrará por defecto `<input type="text">`.

```
<input type="color" value="#b97a40">
<input type="date" value="2013-01-01">
<input type="datetime" value="2013-01-01T16:25:14.12">
<input type="datetime-local" value="2013-01-01T15:41">
<input type="email" value="ejemplo@cibertec.edu.pe">
<input type="month" value="2013-01">
<input type="number" value="6" min="0" max="50">
<input type="range" min="0" max="50">
<input type="search" value="[Buscar...]">
<input type="tel" value="[Numero...]">
<input type="time" value="15:28">
<input type="url" value="http://www.cibertec.edu.pe">
<input type="week" value="2013-W24">
```



Nuevos Inputs, Atributos, Elementos

- Nuevos Atributos HTML5:

Complementan los nuevos tipos de input para las diferentes acciones y validaciones que los desarrolladores puedan necesitar.

```
<input type="text" autocomplete="off">
<input type="text" autofocus>
<input type="submit" formaction="http://www.cibertec.edu.pe/registrar" value="Save">
<input type="submit" formenctype="application/x-www-form-urlencoded" value="Guardar con enctype">
<input type="submit" formmethod="POST" value="Enviar un POST">
<input type="submit" formnovalidate value="No validar">
<input type="submit" formtarget="_blank" value="POST hacia Tab/Window">
<input type="text" list="characters">
<input type="range" max="30">
<input type="range" min="2">
<input type="file" multiple>
<form action="http://www.cibertec.edu.pe" method="GET" novalidate>
<input type="text" pattern="[A-Z]*">
<input type="placeholder" placeholder="Ingrese su nombre...">
<input type="text" readonly>
<input type="text" required>
<input type="text" spellcheck="true">
<input type="number" step="4">
```



Nuevos Inputs, Atributos, Elementos

- Nuevos Elementos HTML5:
Brindan una serie de funcionalidades adicionales, como realizar binding, mostrar progreso, generar claves, etc.

```
<input type="text" name="Comidas" list="lista-datos">
<datalist id="lista-datos">
  <option value="Lomo Saltado">
  <option value="Arroz Chaufa">
  <option value="Seco de Pollo">
  <option value="Arroz con Pollo">
</datalist>
<keygen name="key"></keygen>
<meter min="0" max="15" value="8"></meter>

<input type="range" id="rango" name="rango">
<output for="rango" id="output"></output>

<script>
  (function () {
    var theForm = document.getElementById("form");
    if ("oninput" in theForm) {
      theForm.addEventListener("input", function () {
        output.value = range.value;
      }, false);
    }
  })();
</script>
```



Canvas y SVG

- **Canvas**

Permite representar textos, imágenes, gráficos y formas geométricas de todo tipo, actualizando su contenido en tiempo real. Para ello, utiliza el API Canvas 2D y la aceleración de gráficos por hardware.

```
<canvas id="miCanvas" width="200" height="100">
  Tu navegador no soporta Canvas
</canvas>

<script>

  var c = document.getElementById("miCanvas");
  var ctx = c.getContext("2d");

  // Crear Gradiente
  var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);
  grd.addColorStop(0, "blue");
  grd.addColorStop(1, "white");

  // Rellenar con Gradiente
  ctx.fillStyle = grd;
  ctx.fillRect(10, 10, 150, 80);

</script>
```



Canvas y SVG

- **SVG (Scalable Vector Graphics)**

La renderización en el Canvas se hace como un mapa de bits. SVG es diferente, los comandos se almacenan en una manera que permite ser re-ejecutado. Si el tamaño de la superficie de dibujo cambia, los comandos se pueden escalar y volver a ejecutar para el nuevo dibujo y crear una nueva imagen. El resultado es que se ve una imagen nítida independientemente de la escala.

```
<svg width="100%" height="100%" xmlns="http://www.w3.org/2000/svg" viewBox="0 50 500 175">
  <path d="m267 76 l-21 -4 -144 0 -90 47 0 54 11 11 23 0 15 -30 15 -10 30 0
    15 10 15 30 220 0 15 -30 15 -10 30 0 15 10 15 30 125 0 7 -7
    -13 -38 -20 -10 -95 -15 z" fill="blue" id="carroceria" />
  <path d="m65 105 l40 -25 65 0 0 34 -112 0 z" fill="white" id="ventanaTrasera" />
  <path d="m300 105 l-40 -25 -78 0 0 34 122 0 z" fill="white" id="ventanaDelantera" />
  <circle r="35" cy="185" cx="90" fill="black" id="ruedaTrasera" />
  <circle r="35" cy="185" cx="400" fill="black" id="ruedaDelantera" />
</svg>
```



Audio y Video

- Audio

```
<audio src="audiodat.mp3" controls>
</audio>
```

```
<audio controls>
  <source src="audiodat.ogg" type="audio/ogg" />
  <source src="audiodat.mp3" type="audio/mpeg" />
  <object type="application/x-shockwave-flash" data="player.swf?soundFile=audiodat.mp3">
    <param name="movie" value="player.swf?soundFile=audiodat.mp3" />
  </object>
</audio>
```



Audio y Video

- Video

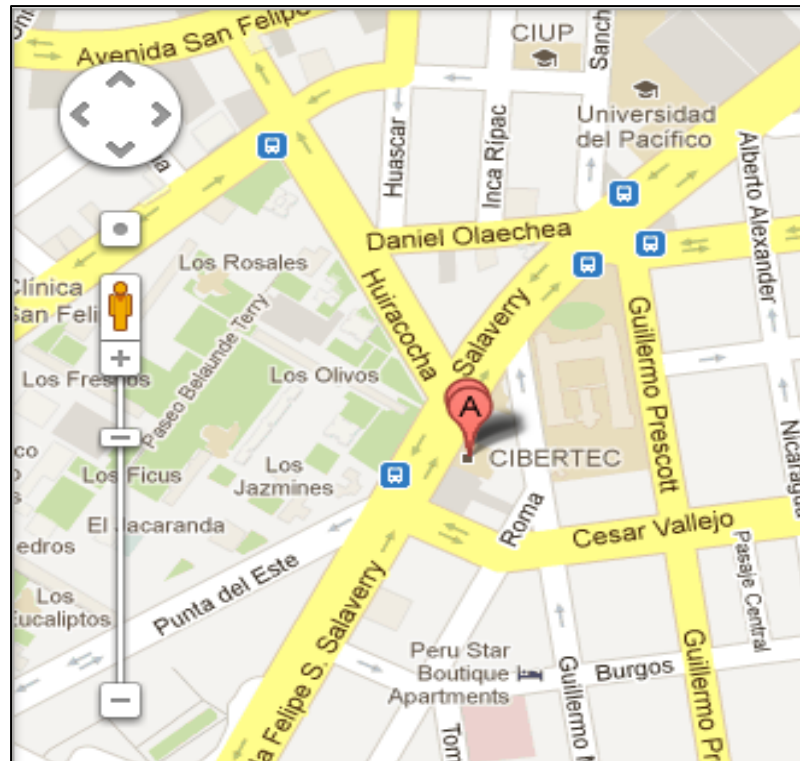
```
<video src="videodat.mp4" controls width="360" height="240" poster="portadadat.jpg">
</video>
```

```
<video controls width="360" height="240" poster="poster.jpg">
  <source src="videodat.ogv" type="video/ogg" />
  <source src="videodat.mp4" type="video/mp4" />
  <object type="application/x-shockwave-flash" width="360" height="240" data="player.swf?file=videodat.mp4">
    <param name="movie" value="player.swf?file=videodat.mp4" />
  </object>
</video>
```



Geolocalización

El API de Geolocalización es definido por la W3C para estandarizar una interfaz que recupere información geográfica de un dispositivo cliente. Es decir, ahora será el propio navegador el que detecta la latitud y longitud (a través de WIFI o GPS del dispositivo).



CSS3

- CSS = Cascading Style Sheets (hojas de estilo en cascada)
- Mientras que HTML nos permite definir la estructura de una página web, los CSS nos ofrece la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos capaces de mostrar contenidos web.
- En la versión CSS3, se realiza la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo de los elementos, sin tener que recurrir a trucos o hacks.



CSS3

| | |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bordes | <ul style="list-style-type: none">• border-radius:25px; |
| Fondos | <ul style="list-style-type: none">• background-size:100% 100%;• background-origin:content-box; |
| Efectos de Texto | <ul style="list-style-type: none">• text-shadow: 5px 5px 5px #FF0000; |
| Fuentes | <ul style="list-style-type: none">• @font-face{font-family: myFirstFont;src: url('Sansation_Light.ttf');} |
| Transformaciones 2D | <ul style="list-style-type: none">• transform: rotate(30deg); |
| Transformaciones 3D | <ul style="list-style-type: none">• transform: rotateX(120deg); |
| Transiciones | <ul style="list-style-type: none">• transition: width 2s; |
| Animaciones | <ul style="list-style-type: none">• @keyframes myfirst {background: red;} to {background: yellow;}} {from• animation: myfirst 5s; |
| Columnas Múltiples | <ul style="list-style-type: none">• column-count:3;• column-gap:40px;• column-rule:3px outset #ff00ff; |
| UI | <ul style="list-style-type: none">• resize:both;• box-sizing:border-box;• outline-offset:15px; |



SASS

- Syntactically Awesome Stylesheets (SASS)
 - Es un lenguaje de extensión
 - Brinda una forma ordenada y sencilla de administrar CSS
 - CSS nativo (lado del servidor) .
 - No ralentiza la ejecución del lado del cliente.
- La potencia de SASS viene con:
 - Variables
 - Herencia y Anidamiento.
 - Mixins



SASS Selector Nesting (Anidamiento)

```
body {  
  font: normal 16px arial;  
  color: #fff;  
  background-color: #011b63;  
  div {  
    font-size: 2.3em;  
    font-weight: bold;  
    ul li {  
      list-style-type: none;  
      a {  
        text-decoration: none;  
        color: white;  
      }  
    }  
  }  
}
```

SASS
Compile

```
body {  
  font: normal 16px arial;  
  color: #fff;  
  background-color:  
#011b63;  
}  
body div {  
  font-size: 2.3em;  
  font-weight: bold;  
}  
body div ul li {  
  list-style-type: none;  
}  
body div ul li a {  
  text-decoration: none;  
  color: white;  
}
```



SASS Selector Nesting (Anidamiento)

```
body {  
  font: normal 16px arial;  
  color: #fff;  
  background-color: #011b63;  
  a {  
    text-decoration:none;  
    color:white;  
    &:hover {  
      text-decoration:underline;  
      color:tomato;  
    }  
  }  
}
```

SASS
Compile

```
body {  
  font: normal 16px arial;  
  color: #fff;  
  background-color: #011b63;  
}  
body a {  
  text-decoration: none;  
  color: white;  
}  
body a:hover {  
  text-decoration: underline;  
  color: tomato;  
}
```



SASS Variables

- SASS soporta **variables**
 - Debemos hacer uso del símbolo de dolar \$
 - Ejem. **\$text-color**, **\$font-size**
 - Variables tienen nombre y un valor
 - Se pueden definir en otra ubicación y siempre estarán disponibles.
 - Se puede definir colores, fuentes, etc...
- Crea código reusable



SASS Ejemplo de Variables

```
$color-non-visited: black;  
$color-visited: tomato;  
body {  
  a {  
    color: $color-non-visited;  
    &:visited {  
      color: $color-visited;  
    }  
  }  
}
```

SASS
Compile

```
body a {  
  color: black;  
}  
body a:visited {  
  color: tomato;  
}
```



Mixins: Bloques resusables

- Mixins: Son un tipo de funciones definidas por el desarrollador
 - Permite la reutilización de código las veces necesarias.
- Existen 2 tipos de mixins
 - Sin Parametros
 - Renderriza el mismo código muchas veces.
 - Con parametros
 - Renderiza estilos basado en los parametros de entrada.
 - Ejem. gradients, borders, etc...



Defining and Using Mixins

- Definición de un Mixin (sin parametros):

```
@mixin default-element-border {  
    border: 1px solid black;  
    background: tomato;  
    border-radius: 10px;  
}
```

- Haciendo uso de la clausula including en un mixin:

```
ul li {  
    @include default-element-border;  
}
```



Mixins with Parameters

- Mixins can also be defined with parameters
 - Defined like in C# / Java / JS and can have default value

```
@mixin opacity-maker($value) {  
  opacity: $value;  
  filter: alpha(opacity=($value*100));  
  zoom: 1;  
}  
  
@mixin box-  
initialize($border:none,$background:rgba(0,0,0,0.7),$si  
ze:200px) {  
  width: $size;  
  height: $size;  
  border: $border;  
  background: $background;  
  padding: 15px;  
}
```



SASS Selector Inheritance (Herencia)

- Para hacer uso de herencia en SASS se usa la clausula `@extend`

```
.default-border {  
  border: 3px solid black;  
  background: tomato;  
  border-radius: 8px;  
}  
.dotted-border {  
  @extend .default-border;  
  border-style: dotted;  
}
```

SASS
Compile

```
.default-border,  
.dotted-border {  
  border: 3px solid black;  
  background: tomato;  
  border-radius: 8px;  
}  
.dotted-border {  
  border-style: dotted;  
}
```



Ejercicio N° 2.1: Estructurar el layout de una aplicación web haciendo uso de los nuevos tags semánticos de HTML5

Crear un nuevo proyecto y aplicar el uso de los nuevos tags semánticos de HTML5. Utilizar los tags header, section, article, footer.

Al finalizar el laboratorio, el alumno logrará:

- Identificar cuando una página web está utilizando HTML5.
- Conocer y aplicar los nuevos tags semánticos.



Ejercicio N° 2.2: Implementar navegación con CSS3 y SASS

Implementar la navegabilidad de nuestro sitio web con estilos CSS3 y SASS

Al finalizar el laboratorio, el alumno logrará:

- Implementar un Framework CSS3 con SASS.



Ejercicio N° 2.3: Crear un formulario y hacer uso de los nuevos inputs, atributos y elementos de HTML5.

Crear una página que use las nuevas etiquetas de input Html5

Al finalizar el laboratorio, el alumno logrará:

- Conocer, usar e identificar nuevos tipos de input HTML5.



Ejercicio N° 2.4: Hacer uso de Audio y Video con HTML5

- Crear una página HTML5 que contenga Audio y Video.

Al finalizar el laboratorio, el alumno logrará:

- Conocer cómo utilizar las etiquetas Audio y Video.



Lecturas adicionales

- Para obtener información adicional, puede consultar los siguientes enlaces:
 - https://www.youtube.com/watch?v=_Swi5_rmztg&list=PLPI81lqbj-4KaMYQ9nrh-NjTlcgmbL1V2
 - <https://www.youtube.com/watch?v=8YFoilZMRvU&index=5&list=PLPI81lqbj-4KaMYQ9nrh-NjTlcgmbL1V2>



Resumen

- HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.
- Especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml).
- Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

