

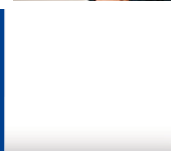
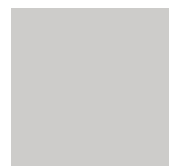
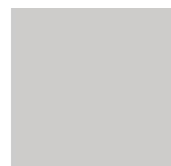
MÓDULO

1

Área: **NEGOCIOS**

Curso: **PROGRAMACIÓN DE APLICACIONES WEB**

Módulo: **Introducción al Lenguaje de Herramientas Web**



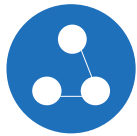
IPP

SUEÑA • APRENDE • CRECE

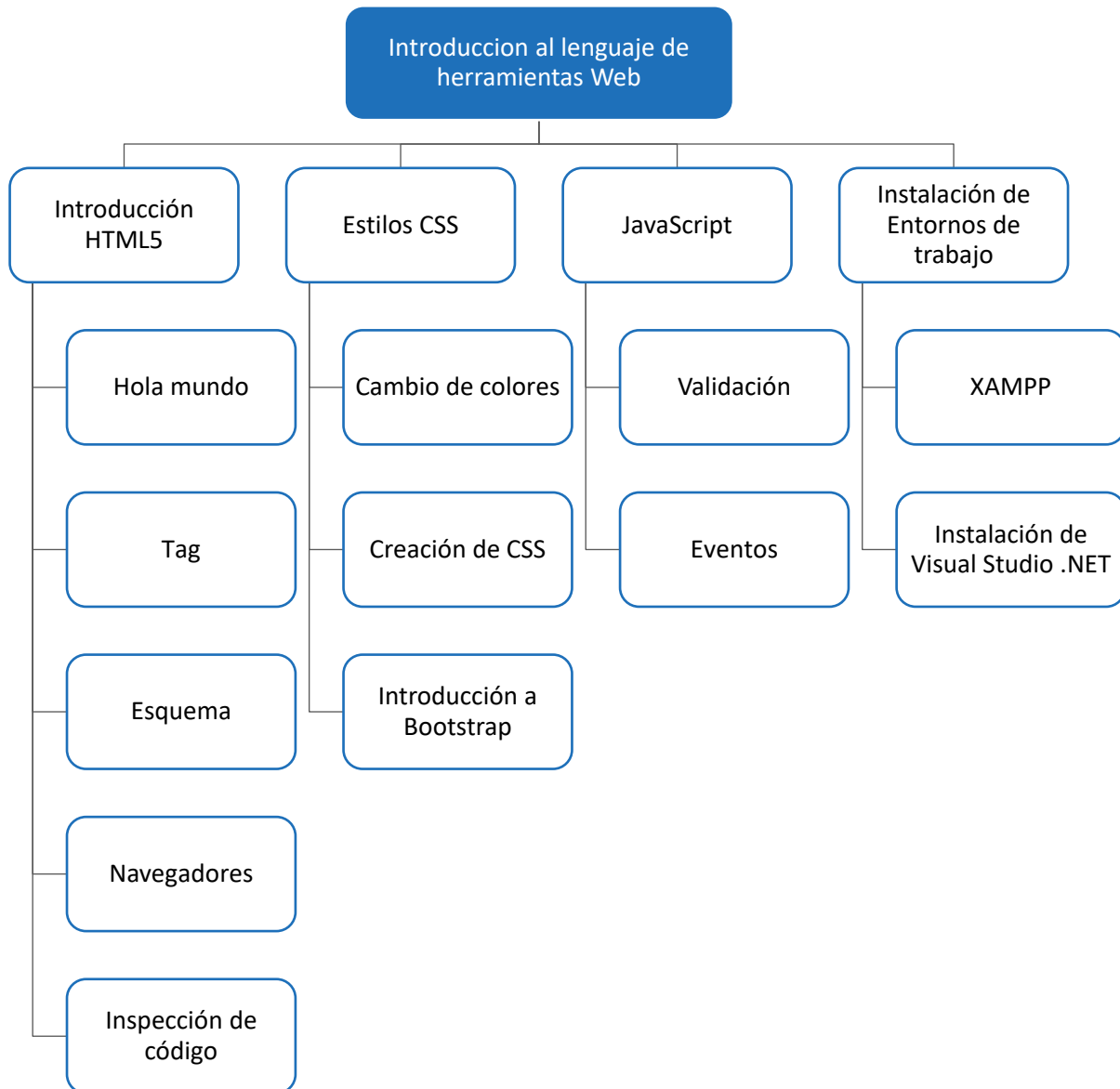


Índice

Introducción	3
Introducción HTML5	4
Estructura de un documento HTML	7
Navegadores	10
Inspección de Código	11
Estilos CSS	23
Sintaxis	23
Cambio de Colores	25
Introducción a Bootstrap	28
Javascript	32
Variables en Javascript	33
Validar un formulario	38
Eventos	39
Comentarios en los distintos lenguajes	42
Instalación de Entornos de Trabajo	43
XAMPP	43
Instalación de Visual Studio	46
Cierre	48



Mapa de Contenido



RESULTADO DE APRENDIZAJE DEL MÓDULO

Crea una página web simple, que cumpla con una estructura fija, sin códigos de programación, utilizando Bootstrap y eventos básicos de Javascript al iniciar la página Web.

Introducción

En el mundo cada día hay más gente con acceso a internet, y por este motivo la tecnología dentro de este ámbito ha ido creciendo más y más. Hoy en día tenemos un sinfín de herramientas en internet, sensores conectados, aplicaciones móviles y diversos sistemas Web.

En esto último se centrará el curso que desarrollaremos a continuación: los Sistemas Web.

Es importante contextualizar que internet hoy en día se utiliza como medio unificado de información, es decir que toda la gente en cualquier parte del mundo con acceso a nuestro servidor puede ver la misma información dentro de sus navegadores web. Claramente es una herramienta indispensable para el manejo de flujos de trabajo en las empresas, ver el avance de estos, tanto como contactar a un cliente con la empresa, mostrar información en tiempo real o también realizar compras para poder realizar distintas tareas. Se pueden generar mil ejemplos de cómo hoy se utiliza esta tecnología, pero en este módulo nos enfocaremos en algo más básico, ya que como siempre se dice “hay que gatear antes de caminar y caminar antes de correr”.

Este módulo estará enfocado en la primera impresión de nuestro sistema web, la presentación del mismo será lo primero que verá el usuario y, dependiendo de cómo esté hecha, se podrá interactuar con ella de manera adecuada, mostrando la información que se quiera representar, de la forma que ustedes estimen.

Todo lo mencionado, enfocado en la experiencia del usuario, es lo que se verá en este curso, recuerden que HTML es el cuerpo o base de una página web, CSS la estética de este y Javascript parte de la inteligencia que permite que sea utilizada de manera más intuitiva e interactiva.

Introducción HTML5

HTML [HyperText Markup Language] o, en español, Lenguaje de marcas de hipertexto, es el lenguaje de etiquetas más utilizado en la web, encontrándose actualmente en su 5ta versión, popularmente conocida como HTML5.

Este lenguaje de etiquetas fue diseñado como una estructura para la presentación sistemática de información basada en el clásico sistema de imprenta. Su funcionamiento es sencillo, ya que consiste en una estructura de etiquetas que es leída por un intérprete, conocido en la actualidad como navegador web (Chrome, Firefox, Opera y Edge son ejemplos de navegadores contemporáneos).

HTML fue creado con el único propósito de mostrar en forma ordenada y concisa la información requerida, por lo que en sus inicios no tenía muchas proyecciones sobre su forma de uso, aspecto visual o funcionalidad, lo que resultó en un alcance limitado de la tecnología comparada con los continuos avances de las necesidades en la web.

Varios desarrolladores crearon alternativas al desarrollo HTML puro para poder confeccionar una web más dinámica y entretenida. En su tiempo Java y Flash, fueron grandes exponentes de estas tendencias logrando mucho éxito, aunque con el tiempo fue más que aparente que presentaban grandes inconvenientes, como que no existía una integración directa de ellos con los navegadores, teniendo que instalar plugins que debían ser constantemente actualizados para poder visualizar correctamente la web, dado que se alojaban en el computador del huésped, sin contar la cantidad de datos que estos necesitaban para operar y recursos que consumían.



Pronto los usuarios comenzaron a integrar el uso de multimedia en las páginas web. Contenido como música, imágenes y hasta reproducción de video dejaron de ser raros para pasar a ser dominantes, lo que generó una gran variedad y volumen de contenido que debía ser presentado de manera adecuada. Este gran crecimiento en las necesidades de la web y la lenta evolución de tecnologías como Java en este ámbito, llevó al lenguaje a abandonar esta pelea del desarrollo web y, producto de esto, flash tuvo una gran crecida en el mercado, siendo conocido a nivel mundial, pero siempre manteniendo los mismos inconvenientes desde su inicio.

Los creadores de los principales navegadores, conociendo muy bien estos inconvenientes, comenzaron a desarrollar lentamente el interpretador de Javascript, el cual fue mejorando gradualmente sin mucho éxito al inicio, pero generando con el tiempo una explosión de funcionalidad. Javascript comenzó a entregar servicios que ninguno de los 2 lenguajes anteriormente mencionados pudo entregar, con la conveniencia de estar

integrado al navegador. Los impulsos de mejoras web con Javascript llegaron a un punto tan alto que se renombró la web y la comenzaron a llamar **WEB 2.0, priorizando** la interacción del usuario con el entorno web. En paralelo a toda esta historia los estilos de visualización CSS (Cascading Style Sheets), en español hojas de estilo en cascada, fueron evolucionando constantemente.

Basándose en la visualización y mejoras en la distribución del contenido dentro de las páginas web, se pasó de un aspecto visual más tosco a uno más pulido y junto con su potencial interacción con Javascript, se mejoraron los efectos visuales y estilos presentes en la web en gran medida.

Así la historia continúa¹, hoy en día con CSS en su 3ra iteración, y con Javascript convirtiéndose en una de las tecnologías punta en procesamiento de información e inteligencia dentro de la web mientras avanza a pasos agigantados, es difícil dilucidar los avances que pueden suceder hoy o mañana.

Con esta breve introducción de la evolución del mundo web, interiorizaremos el curso. Si bien el curso está enfocado en la programación web con PHP y .Net, es de vital importancia el saber manejar las herramientas web de bajo nivel, para poder ser un programador web integral.

HTML

Lo primero que debemos tener claro es que HTML **no es un lenguaje de programación**, es un lenguaje de etiquetas. Este lenguaje tiene una estructura básica y una sintaxis fija.

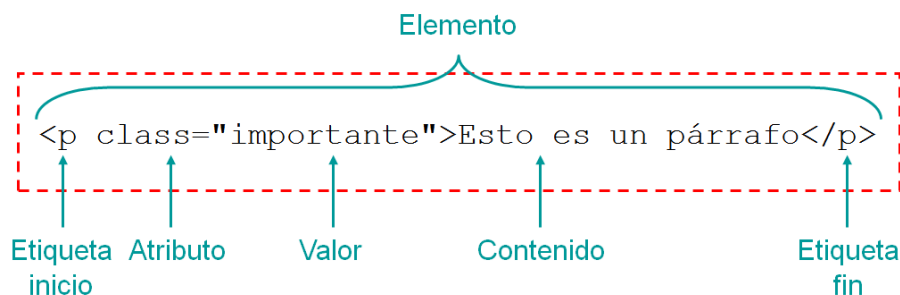
EJEMPLO

Las etiquetas siempre tienen un inicio y un final

`<>` = inicio

`</>` = final

y una estructura bien completa de una etiqueta sería:



¹ <http://www.evolutionoftheweb.com/?hl=es>

Con esto ahora podremos entender a qué se refiere este letrero.



Hoy en su 5^{ta} versión la estructura de distribución que tiene HTML² es:



Pero vamos por partes, esta es la distribución visual y primero tenemos que formalizar la estructura básica de contenido en HTML.

² <https://www.silocreativo.com/disenio-web-con-HTML5-composicion-y-estructura/>

Estructura de un documento HTML



1. **<html>**: Etiqueta que encierra todo el código HTML, indicando que estamos usando este lenguaje.
2. **<head>**: Cabecera, es lo primero de las dos partes de la estructura en un documento en HTML que carga el navegador y también lo primero que interpreta. Posee información sobre cómo se presentará o interpretará el documento a través del navegador.

Ejemplos de contenido dentro del head:

- El título de la página web mostrada por el navegador (en la barra de navegación).
 - Las direcciones de los archivos de estilo (CSS).
 - Las normas de lectura de los navegadores (por ejemplo, usando UTF-8, le informamos al navegador que existen textos con acento y la letra ñ).
 - La dirección del icono que representa a la página y que se muestra en la pestaña del navegador.
 - Metadatos (información utilizada para poner palabras claves de nuestro sitio web, antiguamente los buscadores se guiaban solo por esta información contenida en las páginas web)
 - Entre otras cosas, pero las más comunes son las anteriormente mencionadas.
3. **<body>**: El cuerpo del documento. A diferencia del head, es toda la información o contenido que mostrara nuestra página web. Puede poseer distintas divisiones conceptuales como:
 - **<article>**: Está diseñado para encapsular un conjunto de información autocontenido e independiente del resto de la página.
 - **<section>**: Similar a <div>, se utiliza para denotar distintas secciones o porciones de la página, las cuales uno quiere que tengan un comportamiento distinto.

- **<aside>**: su traducción literal es “Al lado” y, por lo general, se ocupa para definir contenido complementario a otro ya existente.
- **<nav>**: Está pensado para agrupar links de navegación. Muchas veces se utiliza para realizar menús, aunque se recomienda más que nada para cuando se quiere representar un grupo grande de links.
- **<footer>**: Pie de página, esta es la información que se muestra al final de nuestra página web, por lo general se ocupa para colocar las direcciones, agradecimientos o el nombre de los gestores.



Ya visto lo básico de la estructura de html vamos a comenzar a ponerlo en práctica. Lo primero que debemos hacer es abrir un block de notas y escribir lo siguiente:

Nuestro código HTML será:

- `<!DOCTYPE HTML>` agregaremos esto para indicar al navegador que es un documento de tipo HTML (mejora compatibilidad).
- Abrimos la etiqueta de HTML, indicando que el idioma es español `<HTML lang="es">`.
- Dentro de esta etiqueta, creamos el head y el body.

ASÍ QUEDARÍA NUESTRO DOCUMENTO

```
<!DOCTYPE HTML>
<html lang="es">
<head>

</head>
<body>

</body>
</html>
```

Pero no olvidemos nuestras buenas costumbres aprendidas en programación básica, a **INDENTAR**.

CÓDIGO INDENTADO

```
<!DOCTYPE HTML>
<html lang="es">
  <head>

  </head>
  <body>

  </body>
</html>
```

Posteriormente dentro del Head colocaremos el título del sitio con las etiquetas **title**,

```
<title> Mi primera página Web</title>
```

y dentro del body colocaremos "Hola Mundo".

```
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title> Mi primera pagina Web</title>
  </head>
  <body>
    <p>HOLA MUNDO IPP!!!</p>
  </body>
</html>
```

Finalmente guardamos nuestro código con el nombre del archivo **hola.html** (es muy importante que la extensión sea **.html**). Posterior a guardar el archivo lo abrimos, con doble clic o arrastrándolo hacia un navegador.



Navegadores

Tenemos nuestra primera página web con estructura básica y antes de comenzar con el diseño de ella es importante conocer el ambiente en el cual se ejecutarán nuestras páginas web. Es decir, los navegadores.

Cada navegador tiene sus ventajas y desventajas, por este motivo es que muchas veces leemos, que ésta o aquella página funciona “solo con Internet Explorer”.

A continuación, no se hablará de cuál es el mejor navegador, el más rápido o el que consume mayor RAM, sino el que se asocia mejor a la programación web en este caso **HTML**, **CSS** y **Javascript**.

Diferencias y utilidades que nos ayudaran para desarrollar

Internet Explorer, está muy bien integrado a todo lo que es Javascript, aunque se considera antiguo y obsoleto. El nuevo navegador de Microsoft que reemplaza a Internet Explorer llamado **EDGE** está basado 100% en HTML5 generando muchas incompatibilidades en las antiguas páginas web (de la era de Internet Explorer).

A nivel de manejo de gráficos y muestra de contenido (render), el mejor de todos es Chrome. Seguido por Firefox y Opera.

Respecto a compatibilidad se considera que Chrome y Firefox son el tope de línea.

Hasta la fecha, Edge de Windows es una buena solución muy compatible con paginas 100% HTML, pero es común que las empresas soliciten realizar páginas web para otros estándares y que idealmente se vean consistentemente bien en todos los navegadores actuales, entonces no sería la mejor opción programar y testear sobre este navegador.

Las mejores opciones para testing de sus páginas web actualmente, tomando en cuenta lo analizado, serían:

- Chrome (Debido que es el navegador más usado en el mundo)
- Firefox (Es el navegador más usado entre las personas con conocimiento informático)
- Opera (Es un navegador sencillo y equilibrado, muy rápido de operar y con la misma barra de inspección de código que Chrome, perfecto para tester de páginas 100% HTML5)

SABER MÁS

Si deseas saber cuál es el mejor navegador respecto al desempeño que tiene en tu equipo personal, puede probarlo en el siguiente enlace: <http://peacekeeper.futuremark.com/>

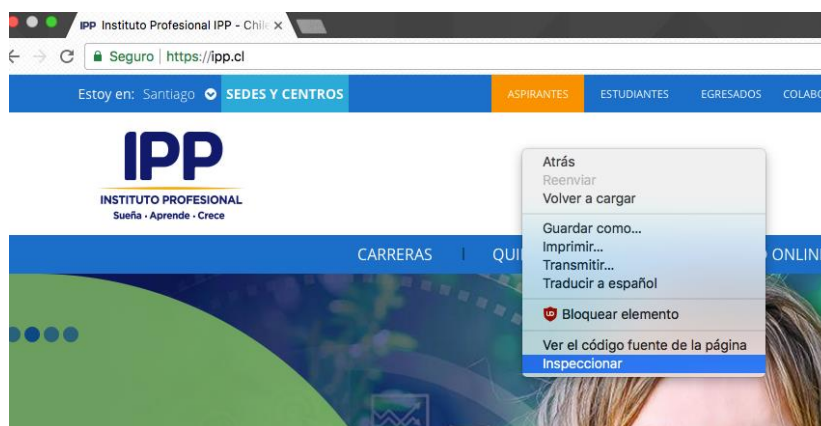
Inspección de Código

En cada navegador podemos ver el código fuente de todo tipo de páginas web, sin restricciones. Todo lo que se encuentra en el HTML, CSS y Javascript es visible por cualquier usuario que ingrese a una página, es decir, son códigos públicos (no quiere decir que cualquiera los pueda ocupar).

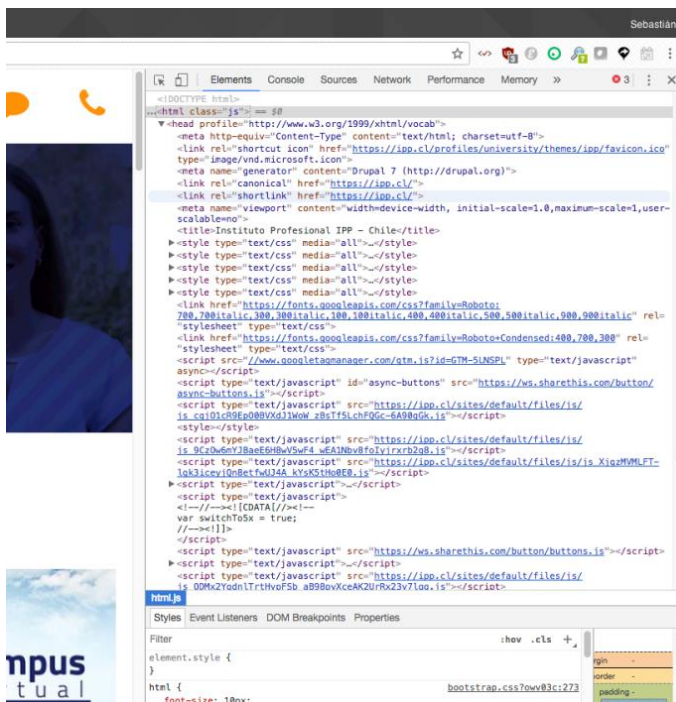
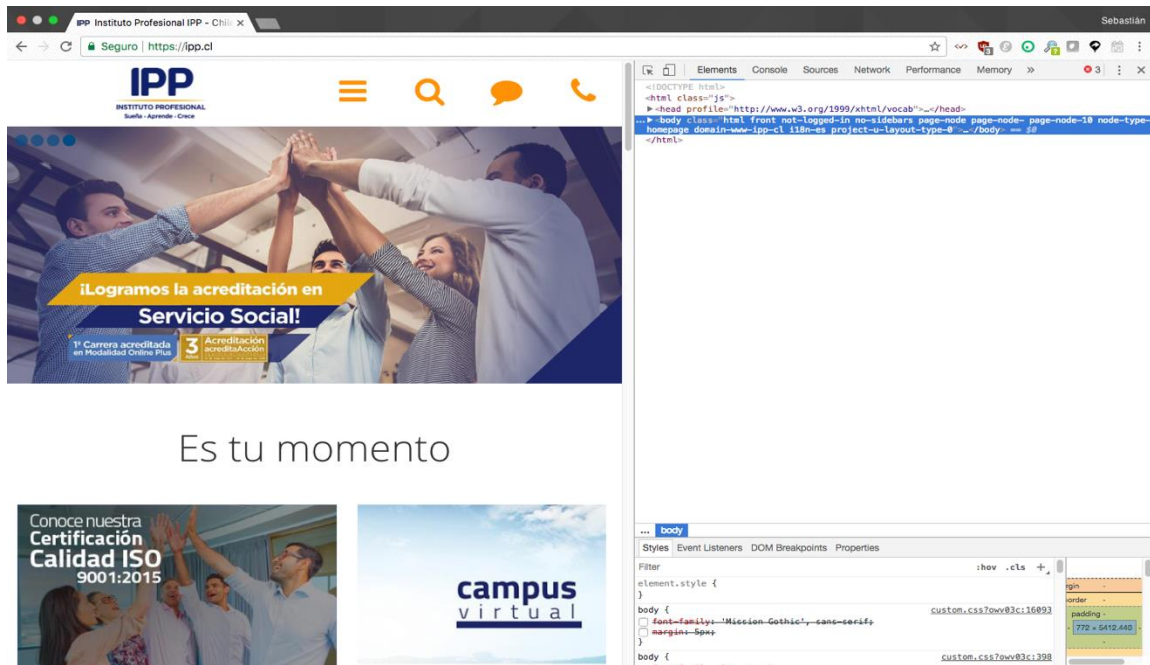
En Chrome, por ejemplo:

Entramos a una página, en este caso www.ipp.cl, y realizamos clic derecho en cualquier lugar de la pantalla.

Esto nos dará la opción de inspeccionar (también se puede realizar por teclado mediante **control + shift + i**). Para realizar lo mismo en Mac el comando cambia (**cmd + alt + i**).



Y veremos lo siguiente:



Si nos fijamos veremos que, a la derecha, dentro del código de la página, se pueden observar secciones como:

1. Tipo de documento.
2. Etiqueta HTML
3. Head
4. Body
5. Etiqueta de cierre.

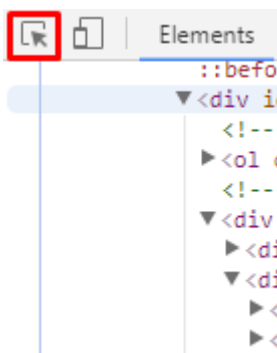
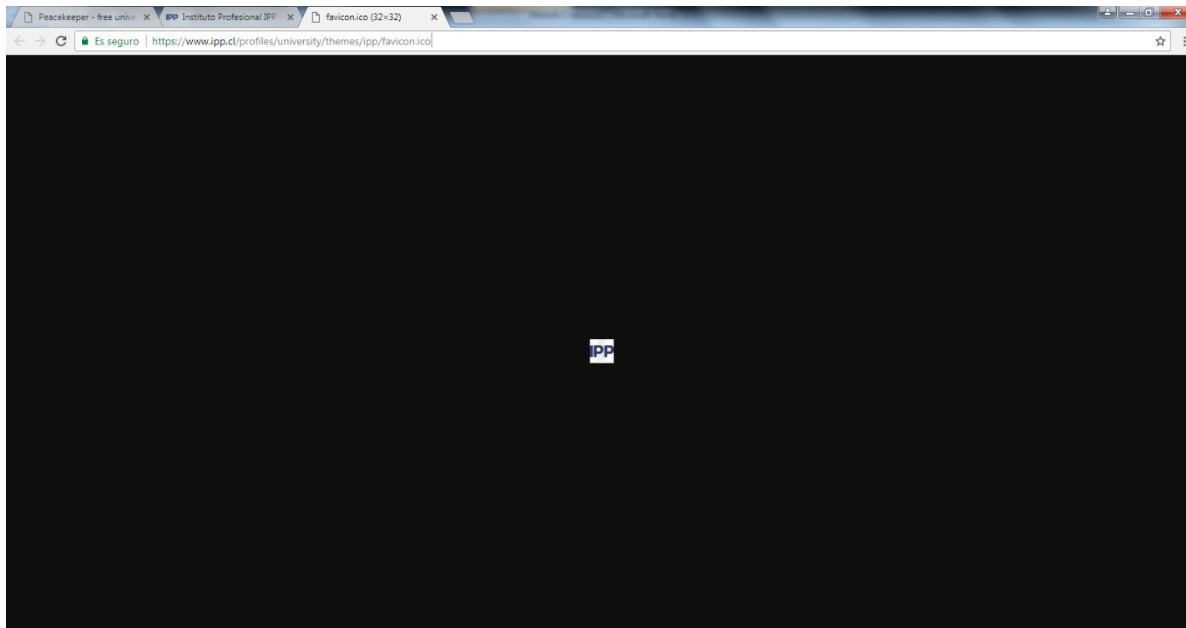
El punto 3 y 4 se pueden desplegar, para ver de mejor manera los códigos.

Si abrimos la etiqueta **head** y revisamos con detalle, observaremos que parte del código dice lo siguiente:

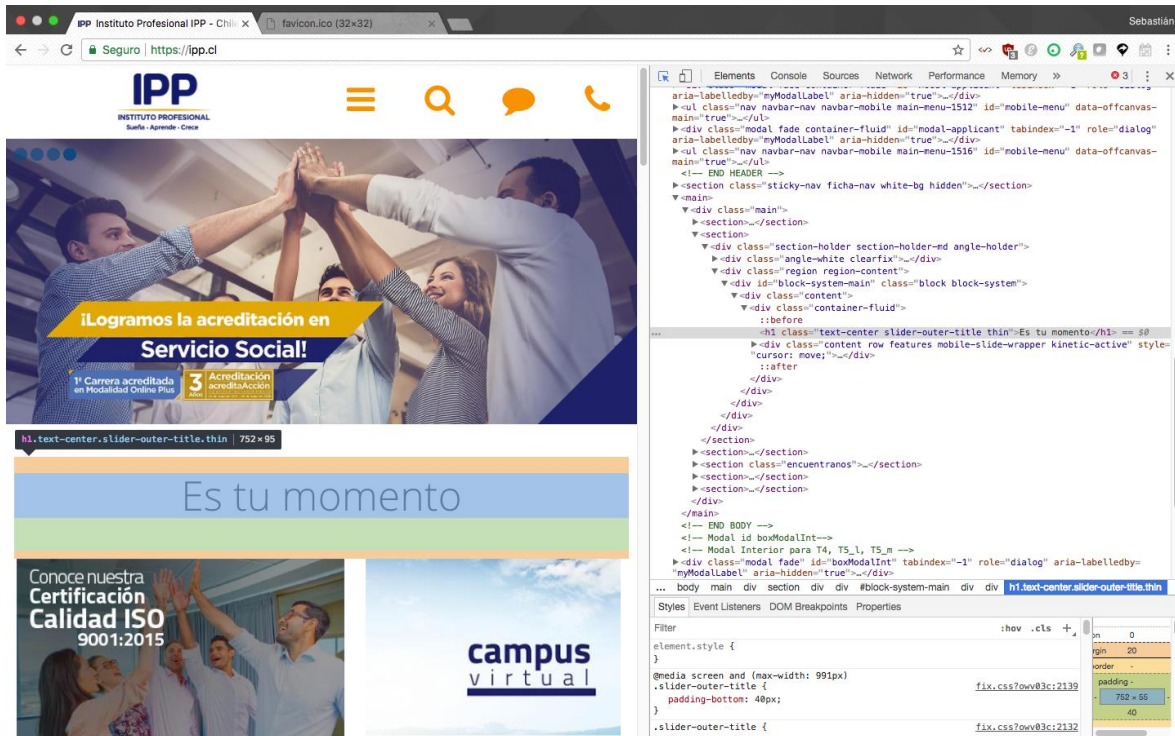
```
<!DOCTYPE html>
<html class="js">
  <head profile="http://www.w3.org/1999/xhtml/vocab">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <link rel="shortcut icon" href="https://www.ipp.cl/profiles/university/
themes/ipp/favicon.ico" type="image/vnd.microsoft.icon">
    <meta name="generator" content="Drupal 7 (http://drupal.org)">
    <link rel="canonical" href="https://www.ipp.cl/">
    <link rel="shortlink" href="https://www.ipp.cl/">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1, user-scalable=no">
    <title>Instituto Profesional IPP - Chile</title>
```

```
<link rel="shortcut icon" href="https://ipp.cl/profiles/university/themes/ipp/favicon.ico"
type="image/vnd.microsoft.icon">
```

Este código de link representa al ícono que nosotros vemos en nuestro navegador en la barra superior de IPP, incluso si copiamos y pegamos el código de la referencia (href) veremos lo siguiente:

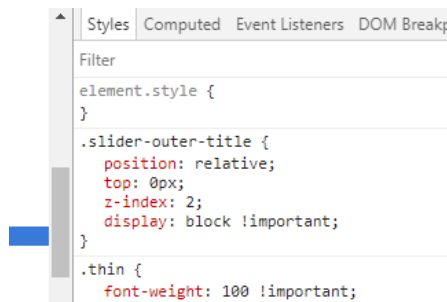


Además de este simple análisis se pueden realizar otros más complejos. Si no quieres leer la página web y prefieres encontrar algo directamente apuntando la página visualmente con el mouse, puedes utilizar el cursor en el siguiente icono que está en la parte superior izquierda del inspector de código.

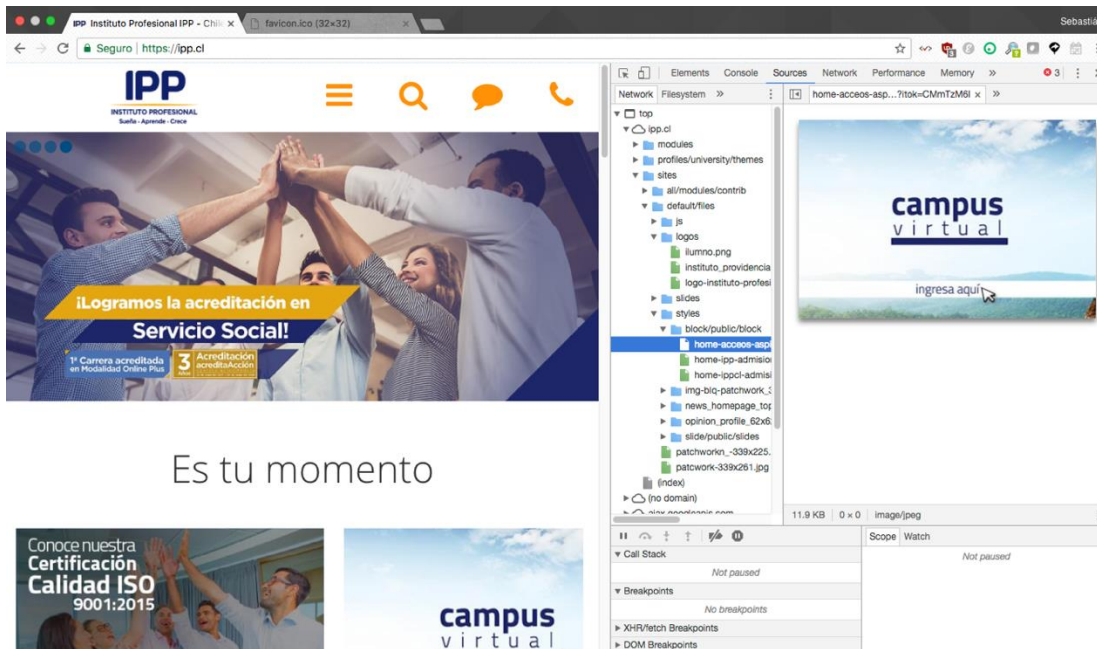


Se realiza clic en el texto **“Es tu momento”**, en ese instante el inspector mostrará que el texto está dentro de una etiqueta de **H1**, la cual tiene diversas clases CSS: **class="text-center slider-outer-title thin"**.

En el cuadro inferior de la derecha, se encuentran las propiedades CSS, las cuales se pueden modificar en vivo, de manera de ir probando cómo afectan los cambios al sitio.



Al hacer clic en la pestaña **Sources**, se pueden ver las carpetas y archivos que componen el sitio web.



Es tu momento

La **Inspección** de código, que ahora podemos utilizar para practicar, será nuestra aliada al momento de ir testeando o realizando cambios en nuestra página web, dado que todas las modificaciones que realicemos dentro del inspector al momento de refrescar o reiniciar la página no se realizarán, entonces es muy bueno para ir probando cosas.

El inspector de Opera es exactamente igual, y el de Firefox tiene algunas funcionalidades y apariencia distintas, pero la lógica es muy similar.

Manos a la obra.

Ahora empecemos a crear una página web más estructurada.

LO PRIMERO ES CREAR LA ESTRUCTURA

```
<!DOCTYPE HTML>
<html lang= 'es'>
  <head>
  </head>
  <body>
    <header>
      <nav>
      </nav>
    </header>
    <section id="wrapper">
    </section>
    <footer>
    </footer>
  </body>
</html>
```

IMPORTANTE

¿por qué **id= wrapper**? Se puede asignar un identificador único a una etiqueta para luego manipular sus propiedades con CSS. El nombre wrapper se eligió porque es un concepto que se refiere a contener otras estructuras y se espera que dentro de esta sección existan nuevas secciones.

1. Poblar la información del head.

El título de nuestra página y los metadatos de nomenclatura que ocupará nuestra página sería 'UTF-8', para nosotros que hablamos español. Es importante recordar que esto efectivamente va dentro de la etiqueta **head**.

ENTONCES QUEDARÍA DE LA SIGUIENTE MANERA

```
<meta charset="UTF-8">
<title>Mi primera página Web IPP</title>
```

2. Poblar información del Header

En esta sección es importante que vaya todo lo que se encuentre visualmente en la parte superior de nuestra página web, El logo, el título de la página y la barra de navegación.

- Una imagen se integra con la siguiente etiqueta ``
- El título de la página es lo más grande que se verá y también uno de los factores que los navegadores buscan, para esto es necesario saber las etiquetas de tamaño que son **H1**, **H2**, **H3** y así consecutivamente hasta **H6**, **el título principal es H1 y el secundario H2** por lo general. Es decir `<h1> titulo 1 </h1> <h2> subtítulo </h2>`
- La barra de navegación, como se mencionó al principio del curso, se puede crear con la etiqueta `<nav>` y por lo general ésta presenta enlaces de dirección a otras páginas. Se tiende a ocupar la etiqueta `` (lista desordenada) para agrupar los enlaces a las distintas partes del sitio, después se utiliza la etiqueta `` dentro de la lista para escribir dentro de ella cada uno de los enlaces que se requieran realizar (cada `li` es un elemento) con la etiqueta ` nombre_enlace `

Entonces el contenido del **body** en el ejemplo quedaría así:

ENTONCES EL EJEMPLO QUEDARÍA ASÍ

```
<header>
  <div>
    <img src='logo.png' width='100' alt='logotipo'>
  </div>
  <nav>
    <ul>
      <li>
        <a href="#">inicio </a>
      </li>
      <li>
        <a href="#">menu 1 </a>
      </li>
      <li>
        <a href="#">menu 2 </a>
      </li>
    </ul>
  </nav>
</header>
```

3. Después agregamos información a la página.

- Primero comenzaremos creando una sección principal (con la etiqueta **section**), designándole el identificador **wrapper** mediante el atributo **id** de las etiquetas HTML (**id** se refiere a un identificador único, por regla no puede haber 2 etiquetas dentro de un mismo documento con la misma **id**). Luego, dentro de esta sección principal crearemos otra sección llamada **main**, en donde estará todo nuestro contenido.
- Dentro del **main** estarán los artículos con información (para el caso crearemos 2, con la etiqueta **article**), el título del artículo se ingresará usando la etiqueta **H2** y para escribir texto dentro de cada artículo utilizaremos la etiqueta **p** que significa párrafo.
- utilizaremos la etiqueta **aside** para crear una **barra lateral** (este no es el uso que se le da normalmente a esta etiqueta, pero luego mediante el uso de CSS se puede ajustar la etiqueta al uso que queremos darle), con un título **H3** e información dentro de un párrafo.
- Para este ejemplo la información de los párrafos se creará por el auto completado que tienen los editores de texto (en este caso, nos referiremos al editor de texto atom), ustedes escriben 'lorem' y les mostrará la información de autocompletar y escribirá un texto por defecto, esto es muy útil cuando quieres realizar plantillas y se requiere rellenar el espacio donde estaría el texto finalmente para verificar como quedaría visualmente la distribución de espacio.

ENTONCES EN EL EJEMPLO LA SECCIÓN WRAPPER QUEDARÍA ASÍ

```
<section id="wrapper">
  <section id="main">
    <article>
      <h2> titulo </h2>
      <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
      </p>
    </article>
    <article>
      <h2> titulo 2</h2>
      <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
      </p>
    </article>
  </section>
  <aside>
    <h3> barra lateral </h3>
    <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>
  </aside>
</section>
```

Nos queda solo el footer, el cual solo tendrá información dentro de un párrafo.

FOOTER

```
<footer>
    <p> Pie de pagina - plantilla ipp </p>
</footer>
```

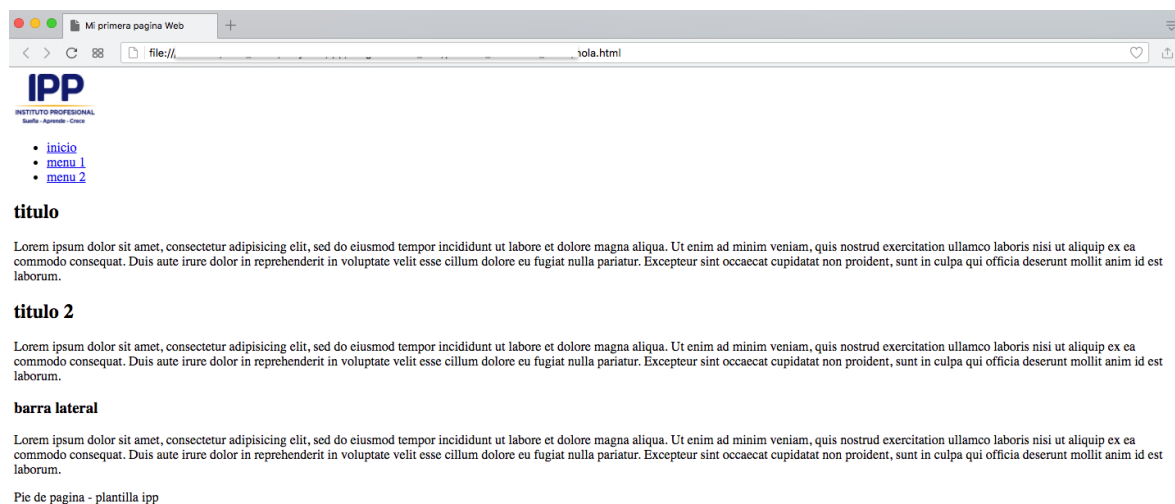
CÓDIGO FINAL

```
<!DOCTYPE HTML>
<html lang= 'es'>
  <head>
    <meta charset="UTF-8">
    <title>Mi primera pagina Web</title>
  </head>
  <body>
    <header>
      <div>
        <img src='logo.png' width='100' alt='logotipo'>
      </div>
      <nav>
        <ul>
          <li><a href="#">inicio </a></li>
          <li><a href="#">menu 1 </a></li>
          <li><a href="#">menu 2 </a></li>
        </ul>
      </nav>
    </header>
    <section class="wrapper">
      <section class="main">
        <article>
          <h2> titulo </h2>
          <p> Lorem ipsum dolor sit amet, consectetur
            adipisicing elit, sed do eiusmod tempor incididunt ut
            labore et dolore magna aliqua. Ut enim ad minim
            veniam, quis nostrud exercitation ullamco laboris
            nisi ut aliquip ex ea commodo consequat.
          </p>
        </article>
```

```
<article>
  <h2> titulo 2</h2>
  <p> Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
  </p>
</article>
</section>
<aside>
  <h3> barra lateral </h3>
  <p>
Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo consequat.
  </p>
</aside>
</section>
<footer>
  <p> Pie de pagina - plantilla ipp </p>
</footer>
</body>
</html>
```

Se debe poner especial atención en que las etiquetas **azules**, demarcan la **estructura básica de HTML**, en **verde** están las **secciones visuales de la página** (header, sección de wrapper y footer) y en **rojo** las **subsecciones** de estas. Siempre hay que recordar que HTML es un lenguaje jerárquico, cuyas etiquetas se contienen unas a otras y que mantener un correcto orden permite una buena representación de la información.

Si abrimos este código en un navegador web, se ve de la siguiente manera.



Estilos CSS

Ya sabemos realizar una página web, ahora la modificaremos de manera que sea más amigable, lo que es posible utilizando CSS.

Es fundamental entender que tanto HTML como CSS **no son lenguajes de programación sino lenguajes de marcado (es decir que representan conceptos y estructuras, pero no aportan inteligencia por si mismos)**. Otro punto importante es que cuando uno busca en internet alguna referencia sobre CSS (ejemplos, definiciones, etc) como resultado pueden aparecer respuestas sobre **CSS, CSS2 o CSS3**. Estos resultados son **técnicamente lo mismo** ya que en cada versión se agregan funcionalidades, manteniendo la misma estructura y funcionalidad pasada, por lo que los códigos realizados con cualquier versión funcionan en los navegadores actuales.

Para aplicar estilos CSS a una página HTML uno puede escribir código tanto en el mismo archivo HTML, como en un archivo aparte con **extensión .css**. Se recomienda siempre ocupar un archivo aparte dado que facilita la escritura de estas hojas de estilos (y permite un mejor orden de código). Recuerden que al igual que en programación básica, se puede ocupar cualquier editor de texto para trabajar, como el block de notas, pero se recomienda utilizar editores de texto más adecuados como en programación básica. Algunos que podrían querer ocupar pueden ser Sublime Text, Notepad++ o Atom, debido que nos ayudaran a distinguir las variables por colores y pueden dar algunos otros tipos de ayuda durante el desarrollo (aunque es una decisión personal, ya que no afecta la funcionalidad del código,).

Sintaxis

El código CSS permite modificar el HTML, mediante una serie de selectores, propiedades y valores.

Su composición es la siguiente:



1. **Selector:** Es la sección de un código CSS que indica a qué elemento se le va a aplicar el estilo CSS. Hay 3 tipos de selectores básicos que es vital reconocer y saber aplicar:
 - a. Selector de elemento: Consiste en escribir directamente el nombre de una etiqueta. Al realizar esto, todo el estilo que se escriba a continuación se aplicará a todas las etiquetas con ese nombre que haya en el documento.

Por ejemplo: `h1 {font-size:12px}`

- b. Selector de id: Consiste en escribir el símbolo gato (#) inmediatamente seguido del nombre asignado al atributo **id** de una etiqueta html. Ya que los identificadores deben ser únicos, esto sirve para aplicar estilo a un solo elemento.

Por ejemplo: #wrapper {font-size:12px}

- c. Selector de clase: Consiste en escribir un punto (.) inmediatamente seguido del nombre asignado al atributo **class** de una etiqueta html. Ya que se puede asignar un mismo atributo class a diversas etiquetas dentro de un documento, este selector sirve para asignar estilos a grupos de elementos.

Por ejemplo: .main {font-size:12px}

2. Una vez escrito el selector, se abre llave y dentro de estas llaves estarán todas las modificaciones que realizaremos sobre el selector.
3. Se realizarán **declaraciones** que indican qué estilos o propiedades se modificarán y cómo. **Cada declaración se separa de otra mediante punto y coma (;)**, siendo posible modificar una propiedad cada vez. La estructura de una declaración sería:
 - Nombre de la propiedad a modificar (fondo, tamaño de letra, tipo de letra, etc.)
 - Dos puntos,
 - El valor que se le dará a propiedad. Dependiendo de qué propiedad utilicemos, varía lo que se debe escribir en esta parte, pudiendo ser uno o múltiples valores los necesarios para modificar el estilo con éxito.
4. Cierre de llave (recuerde que, para cada selector, solo se tomará en cuenta lo que esté entre llaves).

Entendido esto, ahora mostraremos algunos ejemplos en la página web:

Cambio de Colores

EJEMPLO

```
body{
    background: green;
}
```

Primero que todo, al utilizar el selector `body`, se indica que se aplicará el estilo sobre todo lo que contiene esta etiqueta, es decir toda la página visible. Luego vemos que la propiedad `background` es la que se modificará, permitiendo cambiar el color de fondo, para lo cual se escribe el valor **green** que permite dejar el mismo de color verde.

Este código puede estar en nuestro archivo .css junto con cualquier cantidad más de sentencias con la misma estructura base. Por lo general el acceso a los archivos CSS en HTML, tiene que ir en la parte superior de nuestra página web dentro de la etiqueta **head** con la siguiente sintaxis:

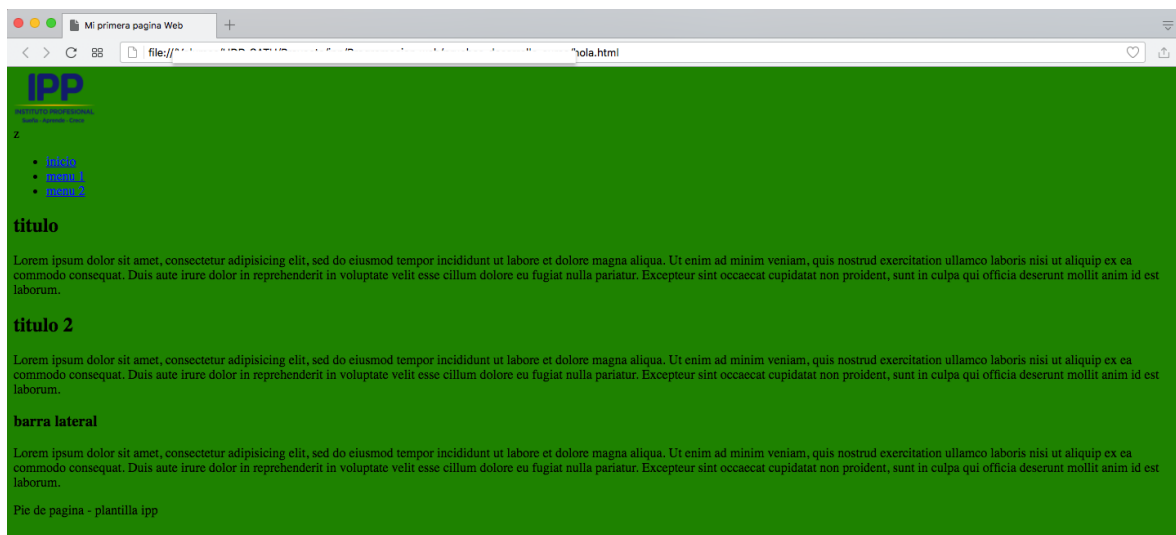
```
<link rel="stylesheet" href='archivo.css'>
```

El atributo **rel** indica que se está refiriendo a una hoja de estilo y **href** al lugar físico dónde se encuentra el archivo en el computador.

Si creáramos un archivo llamado `estilo.css` y lo agregáramos al `head` del ejemplo anterior, tendríamos que escribir la siguiente línea:

```
<link rel="stylesheet" href='estilo.css'>
```

Y la página se vería de la siguiente forma:



Existen una multitud de propiedades que se pueden modificar con CSS, por ejemplo, podemos cambiar el color, tipo de letra de un texto, el tamaño de una imagen es posible modificar los bordes de cualquier elemento haciéndolos visibles, cambiando su grosor, color y propiedades, eliminar sus márgenes, etc.

EJEMPLO

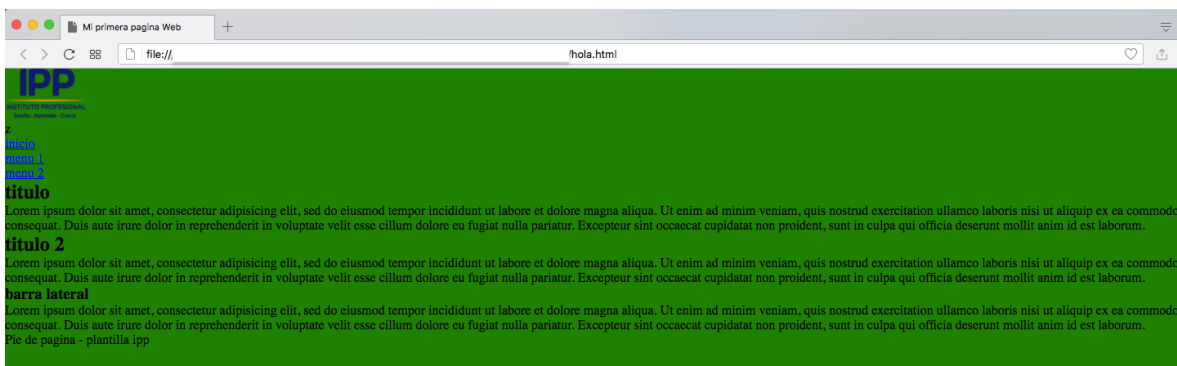
Podríamos querer reducir los márgenes exteriores e interiores de todas las etiquetas del documento. Para este caso utilizaríamos un selector especial que sería un asterisco (*) (lo que significaría que seleccionaría todo).

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Lo podemos agregar al archivo css que utiliza nuestro ejemplo...

```
1  Body {  
2  background:green;  
3  }  
4  * {  
5  margin: 0;  
6  padding:0;  
7  }  
8
```

Y al refrescar la página se vería de la siguiente manera:



EJEMPLO

Si en vez del ejemplo anterior, modificáramos:

- Los títulos h1 y h2, para que sean de color azul.
- Las secciones, para que su color de fondo sea celeste.
- El footer, para que su color de fondo sea amarillo.

Tendríamos un código como este:

```
1  h1,h2 {  
2  color: blue;  
3  }  
4  section {  
5      background: #00A0E3; /*color celeste en html */  
6  }  
7  footer {  
8      background: yellow;  
9  }  
10
```

Quedando la página como en la imagen:



- [inicio](#)
- [menu 1](#)
- [menu 2](#)

título

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute inure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

título 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute inure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

barra lateral

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute inure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Pie de pagina - plantilla ipp

De esta forma se puede modificar la apariencia del sitio, utilizando las posibilidades que el CSS provee.

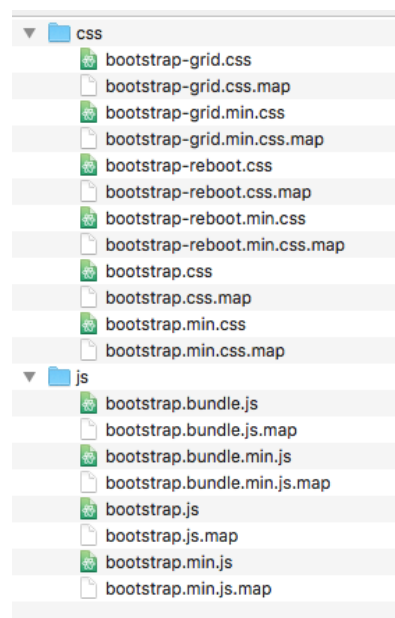
Introducción a Bootstrap

Bootstrap es un Framework creado por **Twitter, Inc.**, el cual nos entrega muchas herramientas para crear páginas web de manera sencilla y amigable.

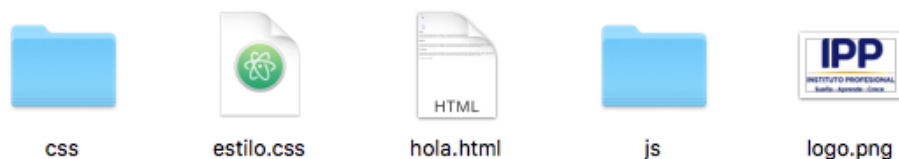
IMPORTANTE

Para descargar Bootstrap tenemos que entrar a <https://getbootstrap.com/> y seleccionar la versión compilada (<https://github.com/twbs/bootstrap/archive/v4.0.0.zip>)

Primero descargamos Bootstrap, lo que nos entregará los siguientes archivos separados en dos carpetas y compresos en un archivo de tipo .zip:



Estos archivos son las plantillas de Bootstrap que operaremos en nuestra página web, por lo que las dejaremos en la carpeta de nuestro proyecto. Si has realizado los ejemplos anteriores por tu cuenta, tu carpeta de proyecto debería verse algo similar a esto:



Y Tomando en cuenta esto, debemos insertar el CSS de Bootstrap en nuestro documento HTML lo que se realizaría de la siguiente manera (para el caso de los ejemplos, consideraremos que esta línea **reemplazará** nuestra anterior inserción de estilo): `<link rel="stylesheet" href=css/Bootstrap.css>`

Al refrescar nuestra página se verá de la siguiente manera:



titulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

titulo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

barra lateral

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Pie de pagina - plantilla ipp

No tuvo grandes cambios, esto es debido a que no utilizamos las clases que Bootstrap nos entrega.

EJEMPLO

Por ejemplo, podríamos dejar la imagen centrada. según la documentación de Bootstrap que se encuentra en el siguiente enlace:

<https://getbootstrap.com/docs/4.0/content/images/>

Una imagen que querríamos centrar para el caso de nuestro proyecto sería el logo del IPP, lo que requeriría de la siguiente aplicación de código, usando la clase "text-center":

```
<div class="text-center">
  
</div>
```

- inicio
- menu 1
- menu 2



titulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

titulo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

barra lateral

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Pie de pagina - plantilla ipp

Como vemos la imagen siempre estará al medio, independiente el tamaño de la página.

EJEMPLO

Ahora agregaremos un botón, veamos la documentación de bootstrap:

<https://getbootstrap.com/docs/4.0/components/buttons/>

Agregaremos un botón verde de los predefinidos por bootstrap:

```
<button type="button" class="btn btn-success">Success</button>
```

- inicio
- menu 1
- menu 2



titulo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

titulo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

barra lateral

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Success

Pie de pagina - plantilla ipp

De esta manera se pueden ir agregando elementos que ya tienen un estilo estandarizado. Si se quisiera modificar los estilos tenemos que colocar nuestros estilos propios después de los de Bootstrap, como vemos en la imagen siguiente:

```
3 <head>
4   <meta charset="UTF-8">
5   <title>Mi primera pagina Web</title>
6   <link rel="stylesheet" href='css/bootstrap.css'>
7   <link rel="stylesheet" href='estilo.css'>
8 </head>
```

Entonces nuestra página web, quedaría de la siguiente forma.



Recuerda que el navegador es un interpretador de etiquetas HTML, que leerá la información de forma secuencial. En el caso leerá los estilos de Bootstrap, cambiará la página según esto y después leerá los nuestros y los aplicará.

IMPORTANTE

Se le insta a que pruebe otras componentes de Bootstrap, creando una presentación de diapositivas usando carousel como un mini ejercicio de profundización antes de continuar con el texto. Basta con copiar el código que quieran usar de los disponibles en el link al final de este párrafo en un documento html con la estructura básica necesaria de utilización de Bootstrap (se aconseja de aquí en adelante usar el starter template indicado en esta página <https://getbootstrap.com/docs/4.0/getting-started/introduction/> como base para la utilización de Bootstrap) dentro de la etiqueta body (donde va el contenido, no lo olviden) y reemplazar los atributos src con las imágenes que quieran en su presentación <https://getbootstrap.com/docs/4.0/components/carousel/>

Javascript

Javascript está en la primera línea respecto a la interacción que tiene el usuario con la página web. Este es un lenguaje de programación que corre en el computador del usuario, permitiendo crear animaciones, validaciones y comportamiento de la página web, es decir, permite una muy necesaria capa de inteligencia y procesamiento de información.

Al igual que en CSS, los códigos Javascript, **desde ahora JS**, se pueden escribir directamente en el código HTML, pero por un tema de orden siempre se recomienda crearlos en archivos externos. Aunque para el primer ejemplo lo realizaremos directo en una página web.

EJEMPLO

```
<!DOCTYPE HTML>
<html lang= 'es'>
  <head>
  </head>
  <body>
    <h1> Esta es una pagina muy bonita. </h1>
    <script>
      alert("Bienvenido a IPP, nuestra nueva pagina web");
    </script>
  </body>
</html>
```

Así se vería nuestra página web:

Esta es una pagina muy bonita.

Esta página dice

Bienvenido a IPP, nuestra nueva pagina web

Aceptar

Como se puede notar, las etiquetas **script** de HTML permiten escribir, dentro de ellas, código js. Para el caso usamos la función **alert**, que permite mostrar un mensaje emergente. Ahora usaremos el mismo script pero lo escribiremos en un archivo externo.

Para llamar a un código Javascript que esté escrito dentro de un archivo .js desde HTML usaremos una línea de este tipo:

```
<script src="archivo.js"></script>
```

ENTONCES AHORA EL CÓDIGO QUEDA DE LA SIGUIENTE FORMA

```
<!DOCTYPE HTML>
<html lang= 'es'>
  <head>
  </head>
  <body>
    <h1> Esta es un pagina muy bonita. </h1>
    <script src='mov.js'></script>
  </body>
</html>
```

El archivo mov.js contendría la siguiente línea:

```
alert('hola mundo');
```

Y el resultado será exactamente el mismo, solo que con el mensaje de alerta diciendo “hola mundo”.

Variables en Javascript

En Javascript, como en todo otro lenguaje de programación, las variables se utilizan para guardar valores.

Todas las variables empiezan con la palabra var, y la estructura para definir una sería:

```
var nombre_variable = valor asignado;
```

Dependiendo de qué valor sea asignado, tratará el dato de forma correspondiente.

EJEMPLO

```
var saludo = "hola mundo IPP";
alert(saludo);
```

Esto nos mostraría el mensaje **hola mundo IPP**, ya que al ingresar el mensaje entre comillas, la variable saludo considera que lo guardado es un string (texto).

Ahora pongámonos en el caso de que queremos que el usuario ingrese un valor y que después la página lo salude refiriéndose a él por su nombre. Pensando en esto podríamos usar la función **prompt**, esta función muestra un mensaje con una línea de texto en donde se puede ingresar información, posibilitando posteriormente usarla en alguna otra tarea.

EJEMPLO

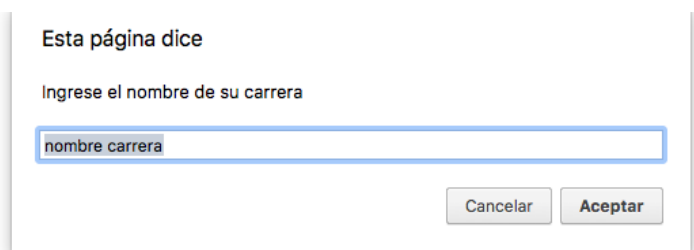
```
var carrera;  
carrera = prompt('Ingrese el nombre de su carrera','nombre carrera');  
  
var mensaje = 'Hola bienvenido a tu carrera de: ';  
mensaje = mensaje.concat(carrera);  
  
alert(mensaje);
```

Qué hace este código:

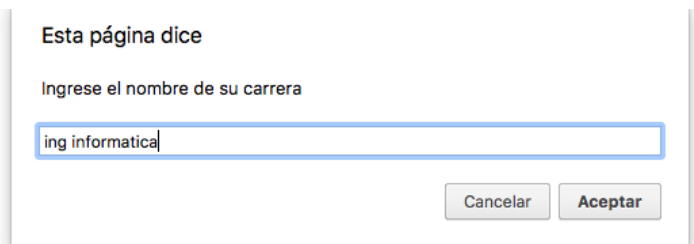
1. Primero, **prompt** pregunta cuál es tu carrera, usando su primer parámetro para definir la pregunta (los parámetros son los valores puestos entre los paréntesis de una función y se separan por comas). El segundo parámetro permite dejar un texto tipo en el espacio para escribir que genera prompt.
2. Lo que se escriba en el espacio proporcionado por prompt, será guardado en la **variable carrera** cuando se presione el botón Aceptar que genera el mismo.
3. Después, utilizando la función **concat** (que concatena (o une) datos de tipo string (o texto), une el texto 'Hola bienvenido a tu carrera de: ' con tu respuesta guardada en la variable **carrera**. Si han concatenado en otros cursos de la carrera, estos términos les parecerán familiares. La estructura de esta función es texto1.concat(texto2).
4. Finalmente, mostrará el texto concatenado y previamente guardado en la variable mensaje a través de una ventana emergente generada con **alert**.

Entonces lo que nos mostrara nuestro navegador en este proceso es:

Esta es una pagina muy bonita.



Esta es una pagina muy bonita.



Esta es una pagina muy bonita.

Esta página dice

Hola bienvenido a tu carrera de: ing informatica

Aceptar

En vez de utilizar la función alert para mostrar el mensaje, podemos mostrarlo dentro de la página agregando texto a nuestro archivo HTML. Para hacer esto, utilizamos la función, **document.write(texto)**.

EJEMPLO

```
var carrera;  
carrera = prompt('Ingrese el nombre de su carrera', 'nombre carrera');  
  
var mensaje = 'Hola bienvenido a tu carrera de: '  
mensaje = mensaje.concat(carrera);  
  
document.write(mensaje);
```

Si realizamos este cambio, en vez de mostrar el último alert, la página se modificaría y mostraría lo siguiente:

Esta es una pagina muy bonita.

Hola bienvenido a tu carrera de: ing informatica

En este caso, escribe la respuesta indicada justo donde se encuentra el script de Javascript en el código.

Condicionales

Es importante tener claro que, en programación, la realización de verificaciones está basada en preguntas lógicas con la posibilidad de cumplir o no una condición. En la mayoría de los lenguajes de programación esto se hace mediante la estructura **if-else**, y este lenguaje no es la excepción.

Por ejemplo, en el siguiente código, se verificará la carrera ingresada por el usuario, generando un comportamiento diferente según la respuesta:

```
var carrera;  
carrera = prompt('Ingrese el nombre de su carrera','nombre carrera');  
  
document.write("Tu perteneces a una ");  
if (carrera == "informatica"){  
    document.write("gran carrera");  
}  
else {  
    document.write("pesima carrera");  
}
```

Si la persona ingresa: informática, le mostrará el mensaje: gran carrera:

Esta es una pagina muy bonita.

Tu perteneces a una gran carrera

Y en el caso de escribir otra carrera:

Esta es una pagina muy bonita.

Tu perteneces a una pesima carrera

Funciones

Una función es una porción de código que se ejecuta cuando es llamada por su nombre, sirve para encapsular código repetitivo u ordenar. Un ejemplo simple de cómo se crea una función sería:

```
function sumar( num1, num2 )  
{  
    return num1 + num2;  
}
```

Se comienza usando la palabra reservada **function** para indicar que se trata de una función, luego el nombre de la función seguido por los **argumentos** que están entre paréntesis y separados por comas, que no son más que valores que la función puede utilizar para realizar las actividades que tenga que realizar (si no se usan argumentos, se deja vacío). Finalmente, dentro de llaves se ponen las líneas de código que realizan lo que se requiere terminando con una sentencia **return**, que es un valor que se retorna al terminar de ejecutar la función.

Basándose en esa función, yo podría hacer una operación como la siguiente usando esta función

```
var resultado = sumar(3,4);
```

Al ejecutar la función **sumar** se le pasan como parámetros los números 3 y 4, por lo que, dentro de la función, estos números toman los nombres num1 y num2 respectivamente. Siguiendo con el desarrollo de la función, se suman estos números y esta suma se retorna, por lo que donde está la función se reemplaza por el valor de la suma entre 3 y 4, es decir 7, lo que se guarda en la variable **resultado**.

Esta función puede ser utilizada cuantas veces se quiera, pasando distintos parámetros y retornando siempre el resultado correspondiente.

Validar un formulario

Tomando en cuenta lo visto, este sería un ejemplo práctico que utiliza Javascript para verificar si se enviaron todos los datos antes de enviar un formulario.

Primero el formulario:

EJEMPLO

```
<html>
  <head>
    <title>Formulario Infalible</title>

    <script src='val.js'></script>

  </head>

  <body>
    <form method="post" action="form1.php" onSubmit="return validar();">
      Nombre: <input type="text" id="txt1" name="txt1" />
      Apellidos: <input type="text" id="txt2" name="txt2" />
      <input type="submit" id="btnEnviar" name="btnEnviar" value="Enviar" />
    </form>
  </body>
</html>
```

Aquí se puede ver como se llama a un js externo que contiene la forma de verificar las casillas del formulario, para esto utilizaremos el envío del formulario como un evento para llamar a una función llamada validar (el concepto de evento se explicará con más detalle luego), la cual analizaremos a continuación al ver el contenido del archivo val.js.

EJEMPLO

```
function validar()
{
  var ok = true;

  if( document.getElementById("txt1").value == "" || document.getElementById("txt2").value == "" )
  {
    alert("Introduzca todos los datos");
    ok = false;
  }
  return ok;
}
```

El funcionamiento de la función validar se basa en un `if` que realiza una pregunta compuesta de varias partes, pero es importante ver toda la secuencia de funcionamiento:

1. Al apretar el botón de enviar en el formulario HTML la sección `onSubmit="return validar();"` del formulario ejecuta la función validar de js.
2. Dentro de esta función, se hace referencia a la función `document.getElementById(id).value` la que permite obtener el valor actual de la casilla con el id especificado.
3. En la pregunta se consideran los valores de los casilleros de texto del formulario y se pregunta por si alguno de ellos es igual a dobles comillas (`""`), que es lo mismo que decir que se pregunta si están vacíos (es el símbolo de un string o texto vacío). La barra doble (`||`) indica que, si la expresión de la derecha o la izquierda de estas barras son verdaderas, entonces toda la expresión completa se considera verdadera).
4. Si alguno de los casilleros estuviera vacío, se activa un alert para indicar que faltan datos y la variable `ok` pasa a ser falsa, hay que hacer notar que esta variable se retornará al `onSubmit` y si es que es falsa, el formulario no se enviará al ser presionado el botón.
5. En caso contrario (es decir, si todos los casilleros de texto están llenos) se permite el envío del formulario de manera normal.

The image shows a web form with two input fields: 'Nombre:' and 'Apellidos:'. The 'Apellidos:' field contains the text 'fghfg'. To the right of the 'Apellidos:' field is a blue button labeled 'Enviar'. Below the form, a modal alert box is displayed. The alert box has a title bar that says 'Esta página dice' and a message that says 'Introduzca todos los datos'. At the bottom right of the alert box is a button labeled 'Aceptar'.

Eventos

Disponemos de los siguientes **eventos** en Javascript que nos permiten ejecutar código cuando ocurren distintos sucesos:

- ▶ **onblur()**: se pierde el foco.
- ▶ **onchange()**: al modificar el valor del componente.
- ▶ **onbeforeunload()**: se lanza al intentar cerrar una ventana.
- ▶ **onclick()**: al hacer clic en un elemento.
- ▶ **ondblclick()**: al hacer doble clic.
- ▶ **onerror()**: si se produce un error al cargar la **página web** o una imagen.
- ▶ **onfocus()**: cuando un componente obtiene el foco.
- ▶ **onkeydown()**: al pulsar cualquier tecla.
- ▶ **onkeypress()**: al pulsar una tecla correspondiente a un carácter.
- ▶ **onkeyup()**: al levantar cualquier una tecla.
- ▶ **onload()**: se produce al terminar de cargarse la **página web** o una imagen.
- ▶ **onmousedown()**: al presionar un botón del ratón.
- ▶ **onmousemove()**: se ha movido el ratón.
- ▶ **onmouseout()**: el cursor del ratón sale del área ocupada por un elemento.
- ▶ **onmouseover()**: el cursor del ratón entra en el área ocupada por un elemento.
- ▶ **onmouseup()**: al dejar de presionar un botón del ratón.
- ▶ **onresize()**: al redimensionar una ventana o **frame**.
- ▶ **onselect()**: cuando se selecciona un texto.
- ▶ **onunload()**: al salir de la **página web**.

Para ver mejor la lógica de utilización de los eventos nos remitiremos a un ejemplo. Primero que todo mostraremos lo que iría dentro del archivo Javascript llamado eve.js:

EJEMPLO

```
window.onload = function()
{
    document.getElementById("btnEnviar").onmouseover = function() {
        this.value = "Vas a enviar el formulario";
    }
    document.getElementById("btnEnviar").onmouseout = function() {
        this.value = "Enviar el formulario";
    }
}
```

- NOTA: el **código fuente** para definir los **eventos** está dentro del evento **onload()** del **Objeto Window()**, el cual se procesa cuando se ha cargado todo el contenido de la **página web**: de no hacerlo así, existe el riesgo de que se intente definir el **evento** de un elemento antes de que haya sido creado, lo que generaría un error.
- NOTA 2: **this.value** es el valor actual del botón, por lo que cuando se le asigna un texto, el valor del botón cambia. El valor del botón es el texto dentro del mismo.

Explicando en detalle la estructura de la utilización de eventos:

1. Para el caso del ejemplo, lo primero que es necesario para utilizar eventos es que todo esté dentro del evento **onload**.
2. Para una acción o acciones en particular que se realizarán cuando suceda el evento, es necesario primero indicar cuál es el elemento que se va a revisar. Para el caso de **onmouseover**, necesitamos saber sobre qué elemento, al pasar el mouse por encima, activará la acción. Aquí se usa **document.getElementById("btnEnviar")** que permite identificar que estamos refiriendonos al un elemento dentro del documento cuyo atributo **id** será **btnEnviar**.
3. **Function** se utiliza aquí para agrupar código e indica que al suceder un evento en específico se realizará lo encerrado entre llaves. Para el caso se cambia el valor del botón (texto del botón).

Luego, esto actúa sobre un documento HTML como el siguiente:

EJEMPLO

```
<!DOCTYPE HTML>
<html lang= 'es'>
  <head>
    <script src='eve.js'></script>
  </head>
  <body>
    <h1> boton cambiante </h1>

    <input type="button" name="btnEnviar" id="btnEnviar" value="ok"/>

  </body>
</html>
```

Finalmente expliquemos que realiza este código completo. La página se trata de un botón que cambia su texto cuando se mueve el mouse por encima del mismo: al pasar el mouse por encima el texto cambia a "Vas a enviar el formulario" y cuando se saca el mouse de encima del botón el texto cambia a "Enviar el formulario".

Comentarios en los distintos lenguajes

No podemos terminar de revisar estas tecnologías sin mostrar como comentar en los distintos lenguajes utilizados, esto permitirá que puedan hacer una buena documentación de los códigos que realicen, ya sea para referencia personal o para poder mostrar a otros lo que construyan.

EJEMPLOS

Comentarios en Html: Se colocan entre los símbolos `<!--` y `-->` .

```
<!--
```

```
Esto es un comentario
```

```
<p>no aparecerá nada en la página ni las etiquetas HTML tendrán valor</p>
```

```
-->
```

Comentarios en CSS: Se colocan entre los símbolos `/*` y `*/` .

```
/* Este es un comentario en CSS
```

```
tampoco se ejecuta nada entre estos simbolos*/
```

Comentarios en Javascript: También se colocan entre símbolos `/*` y `*/`, pero además, cualquier línea de código que comience con doble slash (`//`) es un comentario.

```
/* Este sería un comentario
```

```
en Javascript*/
```

```
//este también es un comentario, de una sola línea
```

Instalación de Entornos de Trabajo

XAMPP

Para poder programar en PHP (en el próximo módulo) se debe ejecutar el código en un servidor. Para estos casos existen servidores locales, en los cuales uno puede trabajar libremente como por ejemplo LAMP, MAMP, WAMP, etc.

Para este curso se considerará el uso de XAMPP, dado su facilidad de instalación y trabajo, este se puede instalar libremente tanto en Windows, Linux y Mac, y de la misma forma (aunque no es una obligación si es más cómodo para ustedes usar algún otro programa que instale una combinación de apache + php + mysql).

IMPORTANTE

XAMPP se puede descargar desde la página web:

www.apachefriends.org

Método de instalación:

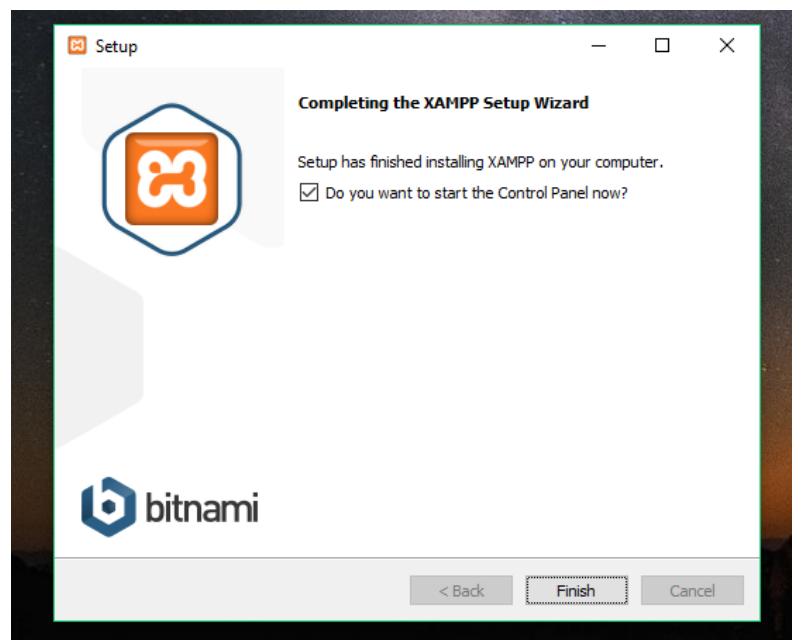
1. Seleccionas el sistema operativo de tu computador.



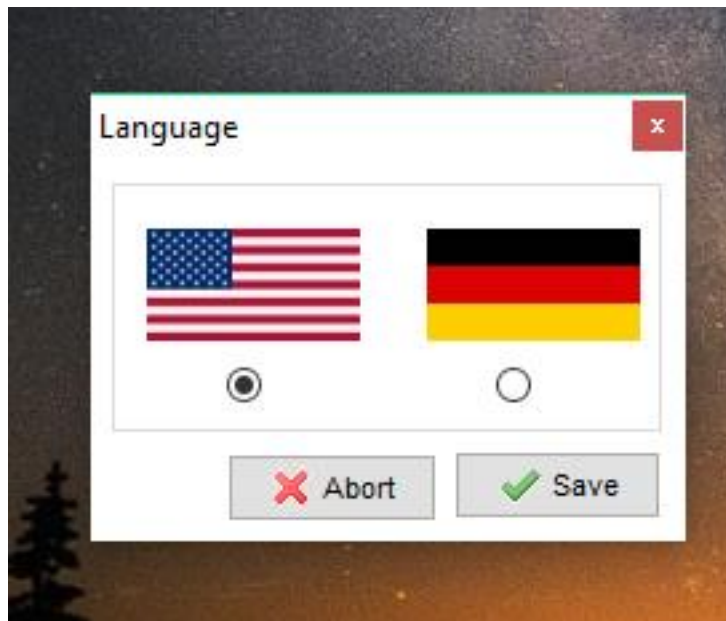
2. Posteriormente en la instalación se deben seguir las instrucciones dadas por el programa. Es importante ejecutar instalador siempre con permisos de Super Usuario (root o administrador).



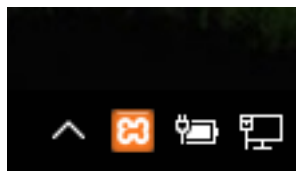
y Listo.



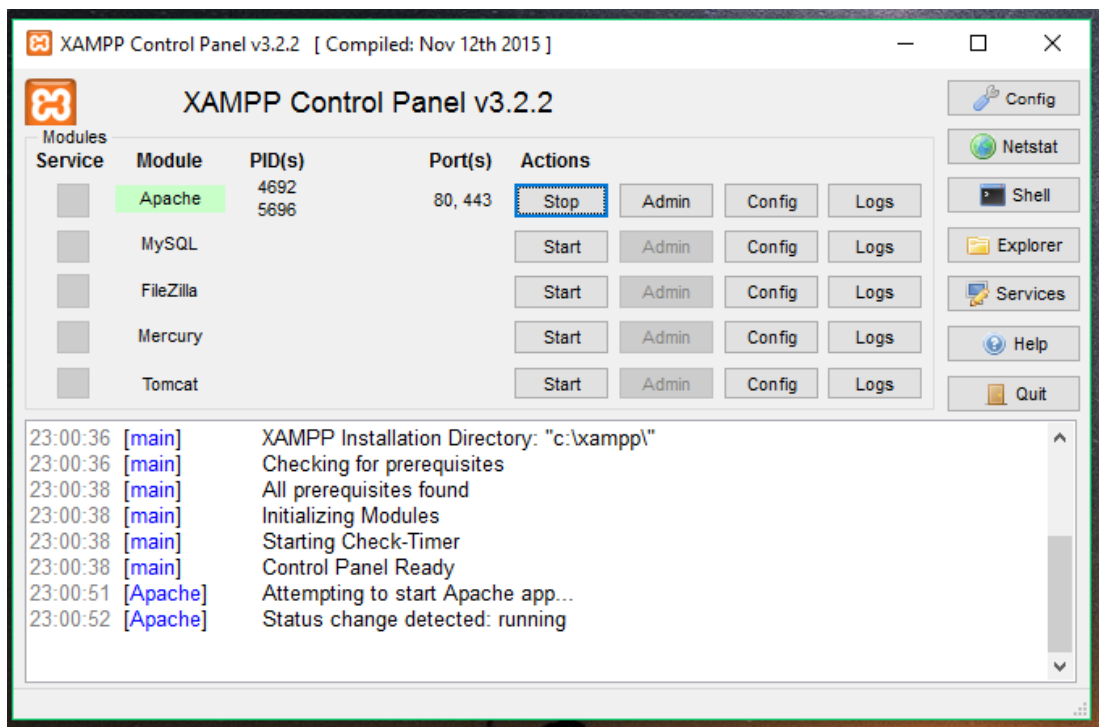
Solo falta elegir el lenguaje con el cual se trabaja.



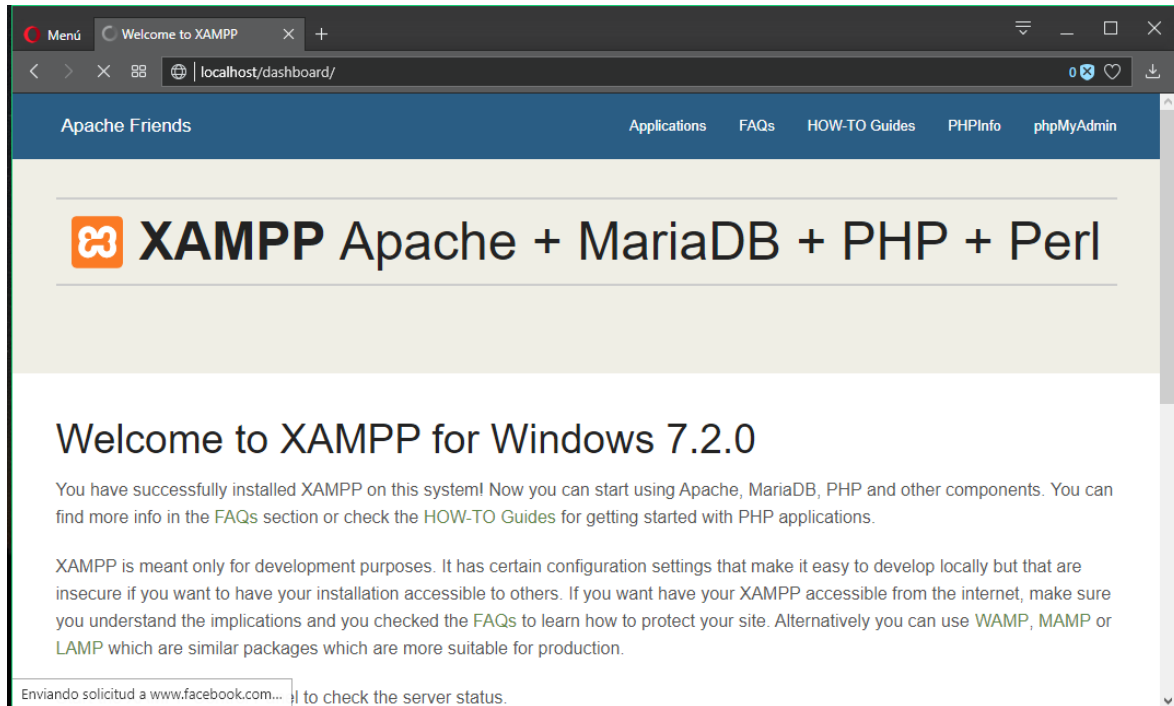
En Windows se debe ver este logo en tu barra de tareas.



Después dentro de XAMPP, ejecutamos el servicio de apache.



Para entrar a nuestro servidor web local accedemos a nuestro navegador y escribimos **localhost** o **127.0.0.1** en la barra de direcciones. El resultado debería ser el siguiente independiente del caso:



Si vez esta página es que está todo listo.

Instalación de Visual Studio

La instalación de Visual Studio será muy importante para el curso, dado que dentro de esta IDE programaremos todo lo relacionado con .NET.

Aquí tienes los pasos de instalación:

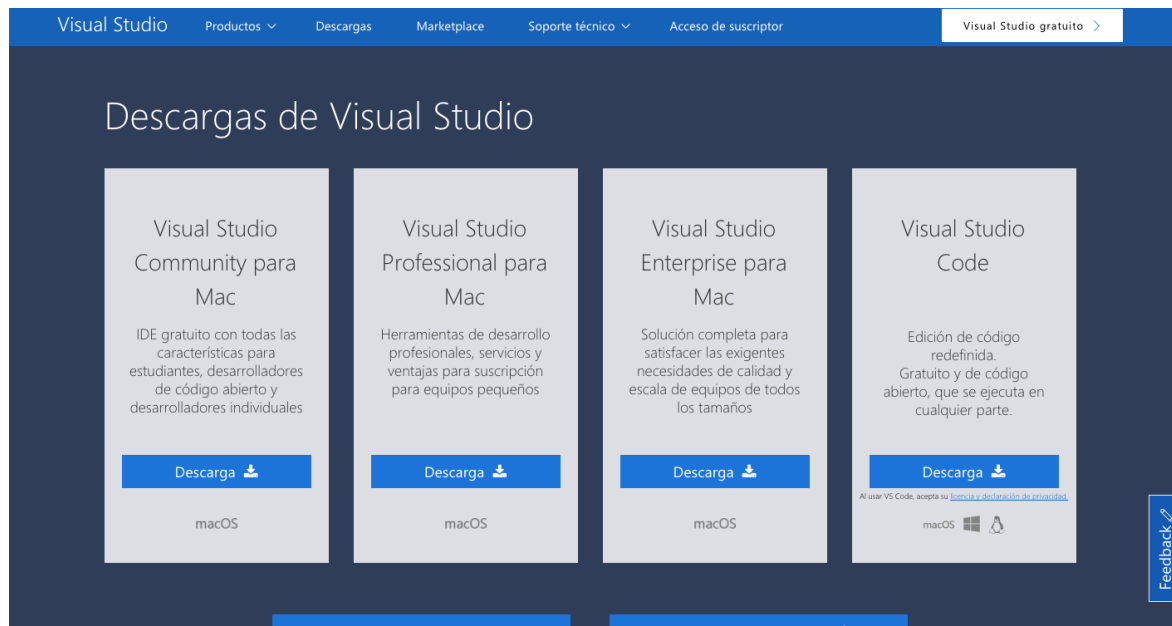
<https://docs.microsoft.com/es-es/visualstudio/install/install-visual-studio>

Y desde aquí puedes bajar el instalador:

https://www.visualstudio.com/es/downloads/?utm_source=mscom&utm_campaign=msdocs

Esta página del instalador mostrará las descargas adecuadas dependiendo del Sistema Operativo que estén usando, por ejemplo:

Vista desde un Mac:



Vista desde un computador Windows:



En esta página pueden elegir tanto la versión Visual Studio Community o la Visual Studio Code para el trabajo dentro del curso. Si están trabajando en Linux solo tiene como opción Visual Code, pero de ser posible, se recomienda utilizar Visual Studio Community.

Cierre

En este módulo revisamos las bases de la creación de una página web, partiendo desde la estructura básica de HTML tanto en la base del lenguaje como de distribución de información, pasando por los estilos visuales aplicables a esta base con CSS y Bootstrap, los cuales nos permiten aplicar estilos a todo nuestro contenido, y culminando con la integración básica de inteligencia y mejora de la interacción con el usuario que da Javascript, pudiendo tener nuestros primeros acercamientos a la verdadera interacción de la web 2.0. Ya que podemos crear una página web estática, en los próximos módulos integraremos verdadera inteligencia a nuestras páginas web, pudiendo convertirlas en verdaderos sistemas web que se transformen en aplicaciones que tengan gran valor agregado y utilidad, razón con la cual se cierra el módulo dando las instrucciones para instalar las nuevas herramientas necesarias para que pronto nos adentremos en el mundo de PHP y .NET, el próximo desafío.

APORTE A TU FORMACIÓN

Hoy en día se necesitan cada vez más profesionales orientados al desarrollo web, debido a que se afirma que todas las empresas, hasta la más pequeña, estarán conectadas a un sistema web, propio o externo, para poder tener el control total en todo momento de sus movimientos.

Con lo aprendido en este módulo podrás entender y por ende trabajar en la creación de sitios web estáticos para una empresa o proyectos personales creando alguna página web con información de tu propio interés.