

**LABORATORIO N°01**

**DESARROLLO EN**

**PLATAFORMAS WEB**

# OBJETIVOS

* Diseñar páginas Web utilizando Etiquetas avanzadas y validación de formularios con hml5.

# SEGURIDAD

|  |  |
| --- | --- |
| **220px-DIN_4844-2_Warnung_vor_einer_Gefahrenstelle_D-W000.svg.png** | **Advertencia:**  En este laboratorio está prohibida la manipulación de hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas. |

# NORMAS EMPLEADAS

* + No aplica

# RECURSOS

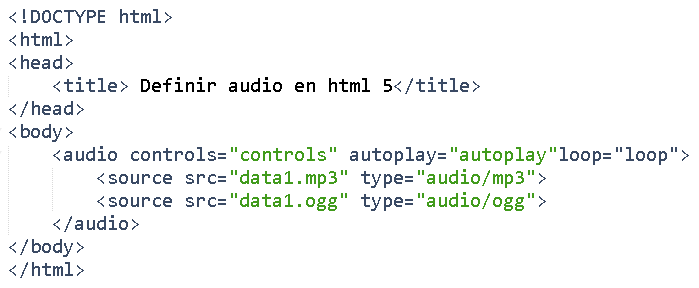
* + En el siguiente laboratorio se trabajará con un equipo que cuente con una conexión a Internet y un navegador web (Mozilla Firefox o Google Chrome).
  + También se requiere Visual Studio Code o Sublime Text.

# METODOLOGÍA PARA EL DESARROLLO DE LA TAREA

# El desarrollo del laboratorio es grupal (2 estudiantes).

# MARCO TEÓRICO Y PROCEDIMIENTO

* **Actividad 1 - Etiqueta de Audio**



Reproducción automática

Controles por defecto de HTML para video.

Tipo de archivo o formato

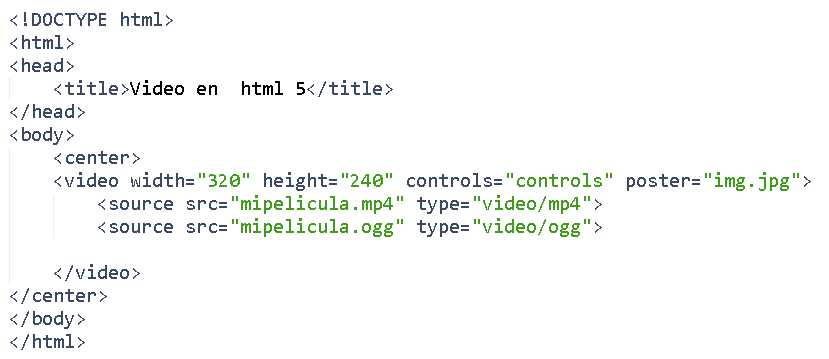
Ruta o dirección URL

Reproducción continua

En el ejemplo en donde se hace uso de la etiqueta audio, vemos el uso del atributo **controls**, con este disponemos de los controles que tiene por defecto para archivos de audio, seguido vemos el atributo **autoplay**, este nos permite que la reproducción del audio sea automática, es decir se reproducirá sin necesidad de poner play, y el último atributo que se usa es **loop,** cuya utilidad es que proporciona reproducción continua, lo que significa que una vez acabe la reproducción del audio este volverá a reproducirse.

En la etiqueta source hacemos uso de dos atributos, el primero es **src**, con la que especificamos la ruta o dirección URL del archivo, y el atributo **type** con la cual indicamos el tipo de archivo que recibirá. Es recomendable tener varios archivos de distintos formatos para que se haga uso del formato de acuerdo a la que el navegador soporte.

* **Actividad 2 - Etiqueta de Video**



Tipo de archivo o formato

Ruta o dirección URL

Thumbnail del video.

Controles por defecto de HTML para video.

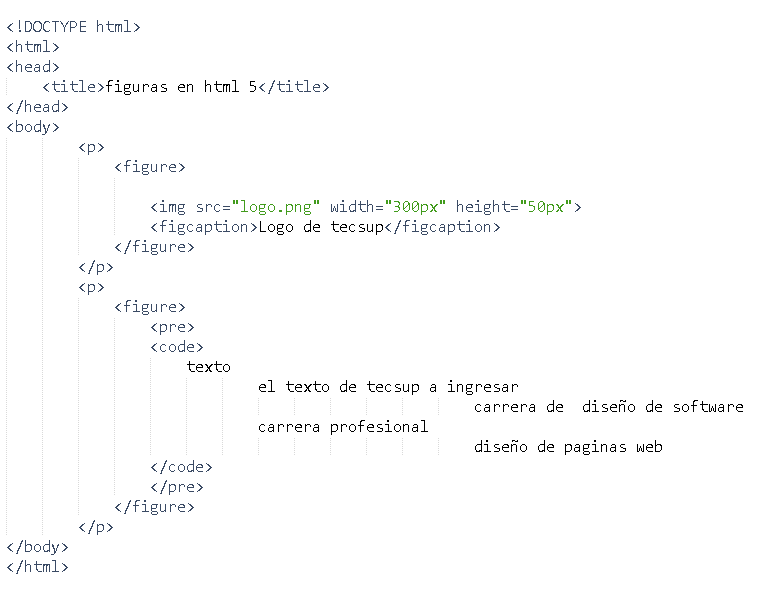
Altura y anchura (dimensiones).



En el ejemplo en donde se hace uso de la etiqueta video, vemos que están los atributos **width** y **height** en donde podemos indicar la anchura y la altura respectivamente que tendrá el video; asimismo vemos el atributo **controls**, el cual nos permite mostrar los controles por defecto de HTML para video, y el último atributo que encontramos en esta etiqueta es **poster**, con este atributo podemos establecer una imagen de preview o thumbnail del video.

Luego con la etiqueta source, útil para la especificación de recursos, hacemos uso del atributo **src**, en la que colocamos la dirección URL o la ruta del archivo de video, y seguido hacemos uso del atributo **type** para especificar el tipo de archivo. Como se ve en el ejemplo, hay dos etiquetas de source puesto que es recomendable contar con diferentes formatos de video para que el navegador puede hacer uso del formato que le sea posible.

* **Actividad 3 - Etiqueta Figure**

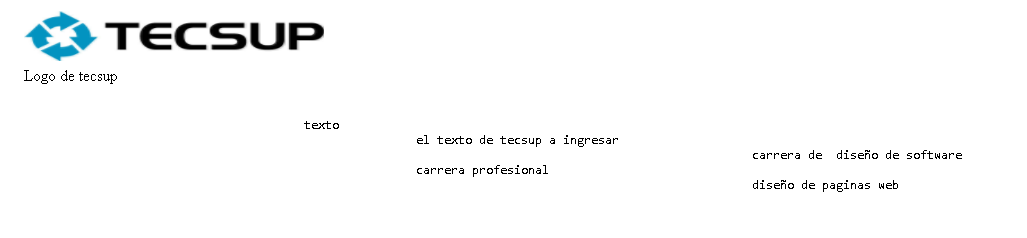


En el ejemplo, dentro de la etiqueta **p**, de párrafo, encontramos la etiqueta **figure**, dentro de esta podemos colocar contenido ya sea un texto, una ilustración o código, y a la vez nos ayuda a no alterar el flujo de contenido. Debajo vemos la etiqueta **img** junto con los atributos **src, width** y **height** para indicar la ruta, la anchura y altura respectivamente, y seguido vemos la etiqueta **figcaption** que nos permite colocar una descripción del contenido, en este caso de la imagen.

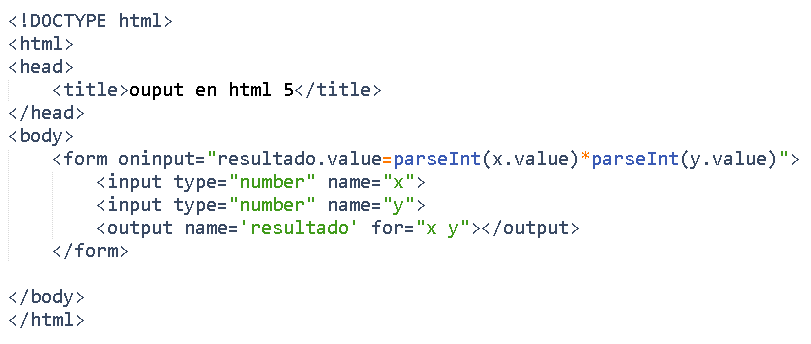
Se vuelve a abrir una etiqueta p, y luego una etiqueta figure, en este caso almacenará como contenido código; primero vemos la etiqueta **pre** y significa que lo que vaya dentro de esta etiqueta se mostrará tal y como se codifico en el archivo HTML, sin omitir espacios; luego haciendo uso de la etiqueta **code**, útil para mostrar código de programación o contenido relacionado.

Altura y anchura (dimensiones).

Ruta o dirección URL



* **Actividad 4 - Etiqueta Output**



Referencia de elementos

Salida de cálculo

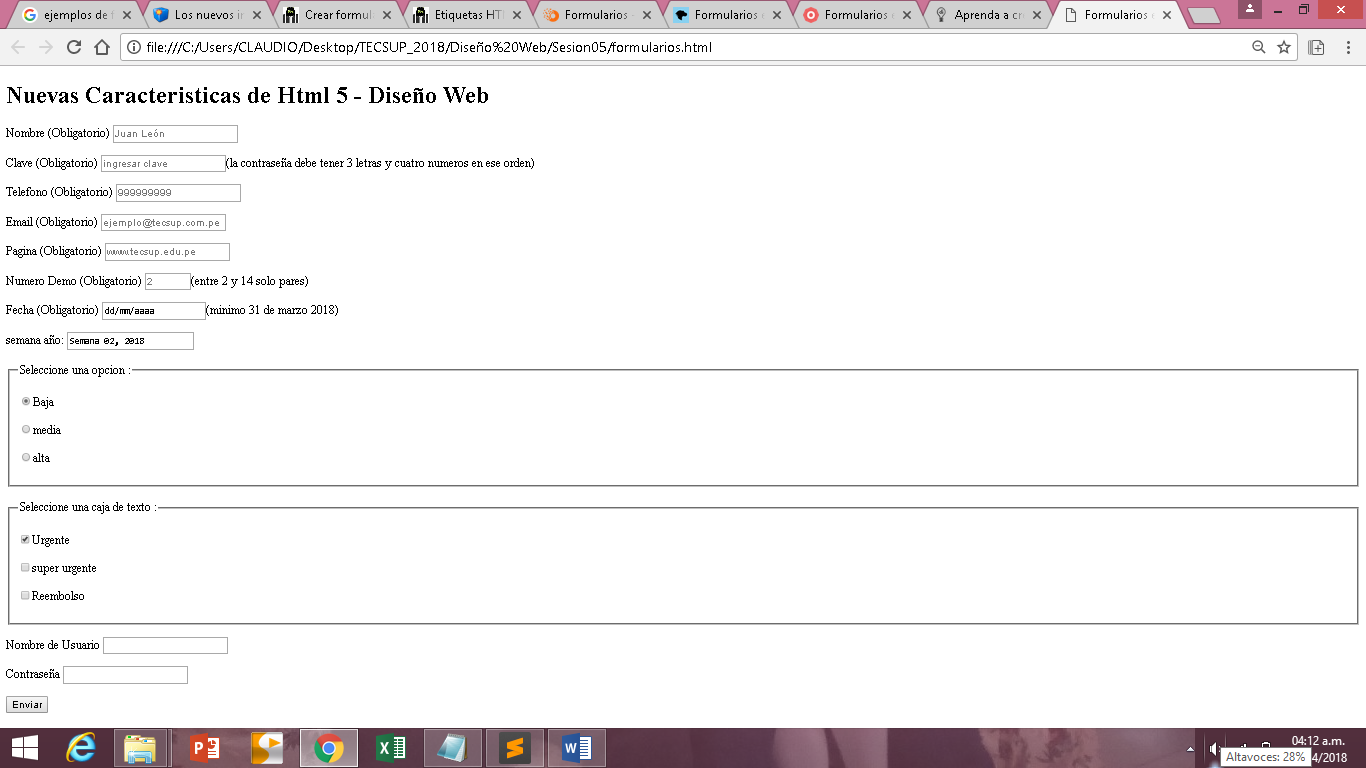


En el ejemplo la etiqueta form tiene como atributo **oninput**, que como vemos recibe una operación en donde asigna como valor a “resultado” el producto de dos valores, la primera es “x” cuyo valor será devuelto como entero y será multiplicado con el valor entero de “y”. Debajo vemos la etiqueta input, que nos posibilita la entrada de datos por parte del usuario y como atributos se usan **type** en la que indicamos el tipo de valor que recibirá y el nombre que tendrá este valor, es similar a declarar una variable cuyo nombre es “x”, lo mismo hacemos para “y”, y finalmente en una etiqueta output nombramos como “resultado” al valor que tendrá como salida y con el atributo **for** referenciamos los elementos que intervienen en la operación “x” e “y”.

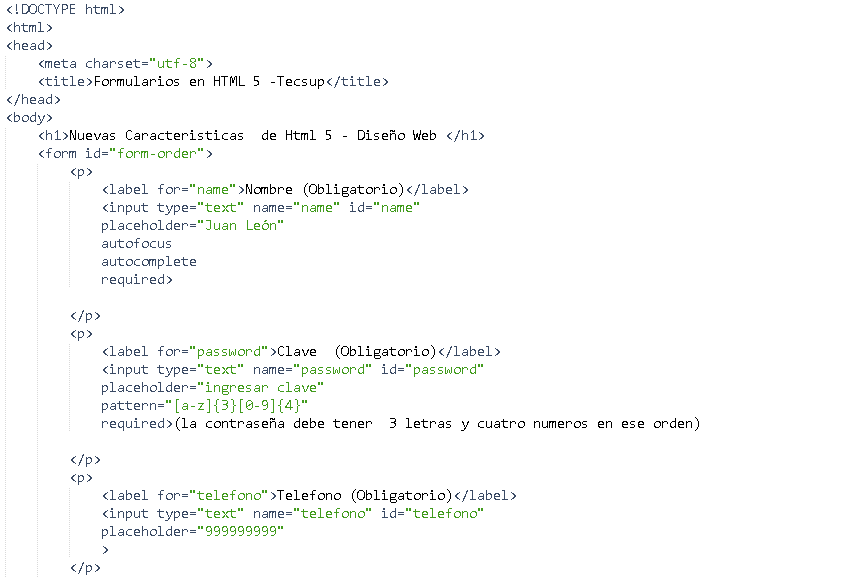
**Diferencia entre input y output**

Los inputs refieren a aquellos datos que provienen de entrada por parte del usuario, mientras que los outputs refieren a aquellos valores que se muestran de salida, producto de algún proceso u operación con datos ya definidos o datos input.

* **Actividad 5 - Formularios con HTML 5**



**Desarrollo:**



El atributo **pattern** nos posibilita hacer reglas para validar el valor introducido, en este caso en corchetes asignamos que los valores serán permitidos si son de la “a” a la “z” y solo se permiten tres letras, y luego podrá introducirse números, pero solo 4; siendo un total de 7 caracteres.

Para la etiqueta input, el valor del atributo **name** será el nombre simbólico que tendrá el valor introducido, y con **placeholder** nos apoyamos para usar un texto de sugerencia, con **autofocus** se coloca el cursor de manera inmediata en el la caja de texto, **autocomplete** velará si el valor introducido puede ser autocompletado por el navegador y con **required** nos aseguramos de tener el campo obligatoriamente lleno antes de enviar los datos del formulario.

El atributo **for** en la etiqueta label sirve para referenciar el input al que corresponderá, es por eso que el valor de este atributo debe ser igual al **id** de la etiqueta input.



En el atributo min se pone el valor de una fecha, puesto que el **type** es “date”, con esto, el calendario que tiene HTML solo mostrará fechas que no sean anteriores a la indicada.

Con el atributo **max** indicamos el valor máximo que puede aceptar la caja de texto, con **min** el valor mínimo y con **step**

Indicamos el intervalo entre los valores válidos, en este caso tomaría solo números pares.



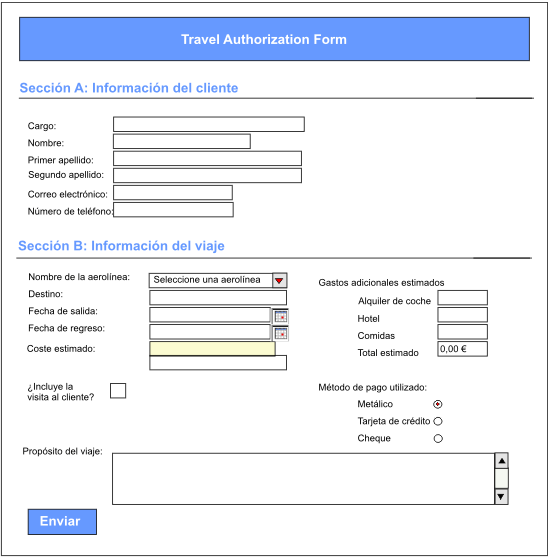
Con el atributo **checked** en la etiqueta input, el campo se mostrará por defecto marcado o seleccionado, asimismo el usuario puede cambiar de selección.

La etiqueta **fieldset** nos permitirá contener campos dentro de una sola agrupación colocando una línea alrededor, y la etiqueta **legend** que nos posibilitará colocar un nombre a dicha agrupación

# TRABAJO PROPUESTO

Según el formulario posterior **(Imagen referencial):**

1. Definir el campo cargo solo en mayúsculas y obligatorio mínimo 4 caracteres máximo 100 obligatorio
2. Definir los campos nombre y apellidos solo con mayúsculas y obligatorio mínimo 4 caracteres máximo 100
3. El correo electrónico es obligatorio y debe cumplir con su formato.
4. El número telefónico solo debe permitir 9 valores numéricos obligatorio.
5. Colocar mínimo 4 aerolíneas.
6. El destino es obligatorio y debe ser solo texto.
7. Las fechas deben ser a partir de enero del 2020.
8. El costo estimado debe estar entre 100 y 1000 obligatorio solo números.
9. El alquiler del coche solo debe permitir el ingreso de numero entre 100 y 600.
10. El campo comidas solo debe permitir valores numéricos



Presentar el o los archivos de la actividad propuesta

# OBSERVACIONES

* Nos agradaría que hubiera ejemplos en donde se ponga en práctica la elaboración del diseño responsive.
* El caso practico lo encontramos muy fructífero al momento de recordar y aprender, por lo que nos gustaría que se nos dejara ejercicios de este tipo en cada laboratorio.
* Los ejemplos podrían tener un poco más de complejidad, seria beneficioso para aprender más.

# CONCLUSIONES

* Se entendió y recordó el uso de etiquetas en HTML que involucran una correcta maquetación, asimismo hicimos uso de los diferentes atributos que proporciona para hacer validaciones en formularios.
* Se comparó que cuando hacemos uso de los atributos de validación de HTML, estos pueden ser modificados por algún usuario, lo que representa que no son muy seguros, mientras que, si hacemos uso de funciones de validación en JS, ningún usuario podría modificar estas validaciones.
* Entendimos la diferencia entre una página web dinámica (aplicativo web) y una página de web estática, con la primera se puede interactuar posibilitando la entrada y salida de datos; mientras que la segunda muestra información idéntica a todos sus usuarios y la interacción no permite entrada de datos por parte del usuario.