

## Tags Encabezados

<html>: Define un documento HTML

<body>: Define el *body* del documento

<head>: Define el *head* del documento

<h1> a <h6>: Define encabezados, donde el <h1>es el de mayor importancia y <h6> el de menor importancia

<hr>: Define una línea horizontal

## Tags Encabezados

<p>: Define un párrafo

<br>: Inserta un salto de línea simple

<pre>: Define un texto con formato pre definido, con texto justificado, tipografía Courier y respeta espacios y saltos de línea.

## Tags Formato

<b>: Define el texto en **Negrita**.

<em>: Define un texto enfatizado para textos o palabras con *importancia semántica*.

<i>: Define el texto en *Cursiva*.

<small>: Define el texto más pequeño.

<strong>: Define el texto con **mayor importancia**.

<sub>: Define el texto <sub>subíndice</sub>.

<sup>: Define el texto con <sup>superíndice</sup>.

<del>: Define el texto ~~tachado~~.

<mark>: Define el texto **resaltado**.

# Listas

HTML permite agrupar elementos que tienen más significado de forma conjunta. El menú de navegación de un sitio web por ejemplo está formado por un grupo de palabras. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su significado conjunto es mayor que por separado. Esto es denominado como listas

## HTML define 3 tipos de listas:

1. Listas no ordenadas
2. Listas ordenadas
3. Listas de definición

## Listas

- `<ol>`: Define una lista ordenada de artículos.
- `<ul>`: Define una lista de artículos sin orden.
- `<li>`: Define un artículo de una lista.

### Ejemplo de un menú de navegación hecho con una lista:

```
<ul id="menu">
  <li><a href="empresa.html">Empresa</a></li>
  <li><a href="productos.html">Producto</a></li>
  <li><a href="servicios.html">Servicios</a></li>
  <li><a href="contacto.html">Contacto</a></li>
</ul>
```

Dicho ejemplo se visualiza:

- [Empresa](#)
  - [Producto](#)
  - [Servicios](#)
  - [Contacto](#)
- 

## Listas de definición

- `<dl>`: Define una lista de definiciones, es decir, una lista de términos y sus definiciones asociadas.
- `<dt>`: Representa un término definido por el siguiente `<dd>`
- `<dd>`: Representa la definición de los terminos listados antes que él.

```
<dl>
  <dt>SGML</dt>
  <dd>Metalenguaje para la definición de otros lenguajes de marcado</dd>

  <dt>XML</dt>
  <dd>Lenguaje basado en SGML y que se emplea para describir datos</dd>

  <dt>RSS</dt>
  <dt>GML</dt>
  <dt>XHTML</dt>
  <dt>SVG</dt>
```

```
<dt>XUL</dt>
<dd>Lenguajes derivados de XML para determinadas aplicaciones</dd>
</dl>
```

Se visualiza de la siguiente forma:

SGML

    Metalenguaje para la definición de otros lenguajes de marcado

XML

    Lenguaje basado en SGML y que se emplea para describir datos

RSS

GML

XHTML

SVG

XUL

    Lenguajes derivados de XML para determinadas aplicaciones

# Tablas

Una tabla es un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos. HTML dispone de una gran variedad de etiquetas y atributos para crear tablas.

Sirven para representar información tabulada, en filas y columnas.

## Etiquetas básicas para tablas en HTML

Las tablas son definidas por las etiquetas `<table>` y `</table>`.

Las tablas son descritas por líneas de arriba a abajo (y luego por columnas de izquierda a derecha). Cada una de estas líneas, llamada fila, es definida por otra etiqueta y su cierre: `<tr>` y `</tr>`

Dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otro par de etiquetas: `<td>` y `</td>`. Dentro de estas etiquetas colocaremos el contenido.

- `<table>`: Representa datos con más de una dimensión.
- `<caption>`: Representa el título de una tabla.
- `<colgroup>`: Representa un conjunto de una o más columnas de una tabla.
- `<col>`: Representa una columna de una tabla.
- `<tbody>`: Representa el bloque de filas que describen los datos concretos de una tabla.
- `<thead>`: Representa el bloque de filas que describen las etiquetas de columna de una tabla.
- `<tfoot>`: Representa los bloques de filas que describen los resúmenes de columna de una tabla.
- `<tr>`: Representa una fila de celdas en una tabla.
- `<td>`: Representa una celda de datos en una tabla.
- `<th>`: Representa una celda encabezado en una tabla.

**A continuación veremos un ejemplo del código de una tabla:**

```
<table class="table">
  <caption>Lista de alumnos.</caption>
  <thead>
    <tr>
      <th>#</th>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Usuario</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1</th>
      <td>Daniel</td>
      <td>López</td>
      <td>@danlopez</td>
    </tr>
    <tr>
      <th>2</th>
      <td>Laura</td>
      <td>Martínez</td>
      <td>@lauram</td>
    </tr>
    <tr>
      <th>3</th>
      <td>Simón</td>
      <td>Luna</td>
      <td>@simonlunar</td>
    </tr>
  </tbody>
</table>
```

# Formularios

## El elemento `<form>`

Todos los formularios comienzan con la etiqueta `form`:

```
<form action="enviador.php" method="post">
</form>
```

El elemento **form** define un contenedor, al igual que un `div`, pero también **soporta atributos específicos para configurar su comportamiento**.

**Action:** define la locación del archivo que recibe los datos del formulario y procesará el envío.

**Method:** define que método de HTML se utilizará para el envío (puede ser GET o POST), por lo general utilizaremos POST.

## Etiquetas propias de los formularios:

### Las etiquetas `<fieldset>` y `<legend>`

La etiqueta **fieldset** tiene como objetivo crear grupos de elementos del formulario que poseen un mismo propósito. La etiqueta **legend**, define formalmente el propósito del elemento `fieldset`.

Se estructura de la siguiente manera:

```
<form>
  <fieldset>
    <legend>Talle de remera</legend>
    <p>
      <input type="radio" name="size" id="S" value="small" />
      <label for="S">Small</label>
    </p>
    <p>
      <input type="radio" name="size" id="M" value="medium" />
      <label for="M">Medium</label>
    </p>
    <p>
      <input type="radio" name="size" id="L" value="large" />
      <label for="L">Large</label>
    </p>
  </fieldset>
</form>
```

---

### La etiqueta `<label>`

La etiqueta **label** define formalmente a cada elemento de un formulario, esta etiqueta es de mucha ayuda para generar un formulario accesible. El principal atributo de la etiqueta `label` es “**for**” quien va a referenciar a “`label`” con su elemento del formulario. Para lograr esta referencia, el valor del atributo `for` debe ser igual al valor del atributo `id` ó `name` del elemento.

```
<form>
  <label for="nombre_apellido">Nombre:</label>
  <input type="text" name="nombre_apellido">
</form>
```

---

### El elemento `<input>`

Este elemento tiene múltiples propósitos según el valor que tome el atributo “**type**”

**Los posibles valores del atributo type son:**

- **text:** Es el valor por defecto, el más común y el input toma la estructura de una línea de texto.

```
<label for="nombre">Nombre:</label>
<input type="text" name="nombre">
```

Nombre:

- **email:** El elemento input acepta como valor una dirección de mail, al momento de ser validada requiere que el texto ingresado tenga formato de correo electrónico.

*Con el atributo **placeholder**, podemos colocar un texto de ayuda para el usuario, dándole una pista de cómo completar el campo*

```
<label for="correo">Correo electrónico:</label>
<input type="email" name="correo" placeholder="nombre@ejemplo.com">
```

Correo electrónico:

- **password:** Este valor es utilizado para campos de contraseñas, el texto ingresado no es visible, se reemplaza por asteriscos o bullets.

```
<label for="contrasena">Contraseña:</label>
<input type="password" name="contrasena">
```

Contraseña:

- **search:** El atributo type toma este valor cuando el formulario será para realizar búsquedas.
- **tel:** Es utilizado para indicar que se ingresará un teléfono.
- **checkbox:** Es un controlador con valor booleano que sólo puede estar deshabilitado o habilitado.

```
<label for="acepta">Acepta términos y condiciones</label>
<input type="checkbox" value="1" id="acepta">
```

Acepta términos y condiciones

- **file:** Este atributo convierte al input para que el usuario pueda seleccionar un archivo de su computadora.

```
<label for="foto">Foto:</label>
<input type="file" name="foto">
```

Foto: no file selected

- **number:** valida que la información ingresada por el usuario sea un número.
- **radio:** Radio button define un elemento con múltiples opciones que puede tomar sólo un valor como respuesta.

```
<form>
  <input type="radio" name="sexo" id="H" value="hombre" />
  <label for="hombre">hombre</label>
  <br>
  <input type="radio" name="sexo" id="M" value="mujer" />
  <label for="mujer">mujer</label>
</form>
```

hombre

mujer

- **submit:** Especifica un botón que tiene como comportamiento enviar el formulario.
- **reset:** Especifica un botón cuyo comportamiento es vaciar todos los campos del formulario.

```
<form>
  <input type="submit" value="Enviar formulario"/>
  <input type="reset" value="Limpiar formulario"/>
</form>
```

•

---

## El elemento <textarea>

Es un campo con múltiples líneas de texto, tiene un comportamiento similar al de un input de texto con la propiedad de aceptar saltos de líneas.

*El campo textarea acepta los siguientes atributos:*

- **cols:** Indica la cantidad de caracteres que podrá tener el campo a lo ancho.
- **row:** Indica la cantidad de líneas de texto.

## Los elementos <select>, <option> y <optgroup>.

La etiqueta **select** define un combo box, en el cual se podrá seleccionar una o más opciones según el valor de los atributos de este campo.

Cada opción dentro del elemento **select** se define como **option** y los mismos pueden ser agrupados con la etiqueta **optgroup**.

Si el elemento **option** tiene seteado el atributo “value” es el valor que se enviará al enviarse el formulario.

El elemento **optgroup** funciona como título de un conjunto de opciones, pero no es clickeable.

*Atributos del elemento option:*

- **label:** texto descriptivo de la opción.
- **selected:** es un valor booleano que indica si el elemento se muestra seleccionado por defecto.

```
<select name="talles">
  <optgroup label="Adultos">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </optgroup>
  <optgroup label="Niños">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </optgroup>
</select>
</form>
```

```
<form>
```

```
<select name="talles">
  <optgroup label="Adultos">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </optgroup>
  <optgroup label="Niños">
    <option value="L">Large</option>
    <option value="M">Medium</option>
    <option value="S">Small</option>
  </optgroup>
</select>
</form>
```



# Enlaces

Los enlaces (también conocidos como links o anchors) se utilizan para relacionar partes del documento con otros documentos o con partes del mismo documento. Por defecto, los enlaces se visualizan azules y subrayados.

Para crear un enlace es necesario utilizar la etiqueta de ancla `<a>` con el atributo **href**, que establecerá el destino al que apunta.

Sintaxis de un enlace:

```
<a href="productos.html">Productos</a>
```

El texto "productos", que figura entre las etiquetas `<a>`, es el que se visualizará en el navegador y que servirá como enlace a la url especificada en el atributo **href**.

## Enlaces relativos, absolutos e internos.

Los enlaces relativos son aquellos que apuntan a páginas ubicadas dentro del mismo proyecto.

Si la página referenciada se encuentra en el mismo directorio alcanza con mencionar el nombre de la misma para generar el enlace.

```
<a href="contacto.html">Contacto</a>
```

En caso de que el archivo se encuentre en un directorio específico, el mismo deberá ser mencionado.

```
<a href="imagenes/mapa.jpg">ver mapa</a>
```

En el ejemplo anterior, el destino al cual queremos acceder (en este caso una imagen) se encuentra dentro de la carpeta "imagenes" con lo cual hay que especificar esa ruta en el atributo "href".

Los enlaces absolutos son aquellos cuyo destino apunta a un documento que fuera del sitio y debe ser especificado utilizando la URL completa:

```
<a href="http://www.coderhouse.com/frontend">Curso de Frontend</a>
```

Los enlaces internos permiten referenciar secciones de nuestra página, para hacer esto se utiliza el id:

```
<a href="#pie">Ir al pie de página</a>
...
<footer id="pie"></footer>
```

También podemos usar como destino una sección específica una página distinta.

```
<a href="contacto.html#formulario">Formulario de contacto</a>
```

En el ejemplo anterior el enlace apunta a la sección que tiene el id formulario, dentro de la página "contacto.html".

No sólo podemos agregar enlaces a texto, también podemos hacerlo con otros elementos. Por lo general se usan texto o imágenes. Ejemplo de enlaces con una imagen:

```
<a href="http://www.coderhouse.com/cursos.html#frontend">
  
</a>
```

En este caso la imagen será clickeable y enlazará al destino especificado en el atributo `href`.

# Audio y video

HTML5 soporta contenido multimedia gracias a los elementos `<audio>` y `<video>`

Insertando contenido multimedia

Insertar contenido multimedia en tus documentos HTML es muy sencillo:

**Para insertar un video video usaremos la siguiente sintaxis:**

```
<video src="videos/tutorial.ogg" controls>
  Tu navegador no implementa el elemento <code>video</code>.
</video>
```

**de la misma forma insertaremos audio con el siguiente código:**

```
<audio src="audio.ogg">
<p>Tu navegador no implementa el elemento audio.</p>
</audio>
```

El atributo src puede ser una URL del archivo de audio o la ruta al archivo en el sistema local.

**Atributos de la etiqueta `<audio>`:**

- controls: muestra los controles estándar de HTML5 para audio en una página web.
- autoplay: hace que el audio se reproduzca automáticamente.
- loop: hace que el audio se repita automáticamente.
- preload: es usado en el elemento audio para almacenar temporalmente (buffering) archivos de gran tamaño. Este puede tomar uno de 3 valores:
  1. "none" no almacena temporalmente el archivo
  2. "auto" almacena temporalmente el archivo multimedia
  3. "metadata" almacena temporalmente sólo los metadatos del archivo

Se pueden especificar múltiples fuentes de archivos usando el elemento `<source>` con el fin de proporcionar vídeo o audio codificados en formatos diferentes para diferentes navegadores. Por ejemplo:

```
<video controls>
  <source src="foo.ogg" type="video/ogg">
  <source src="foo.mp4" type="video/mp4">
  Tu navegador no implementa el elemento <code>video</code>.
</video>
```

Esto reproduce el archivo Ogg en navegadores que admiten el formato Ogg. Si el navegador no admite Ogg, el navegador usará el archivo MPEG-4.

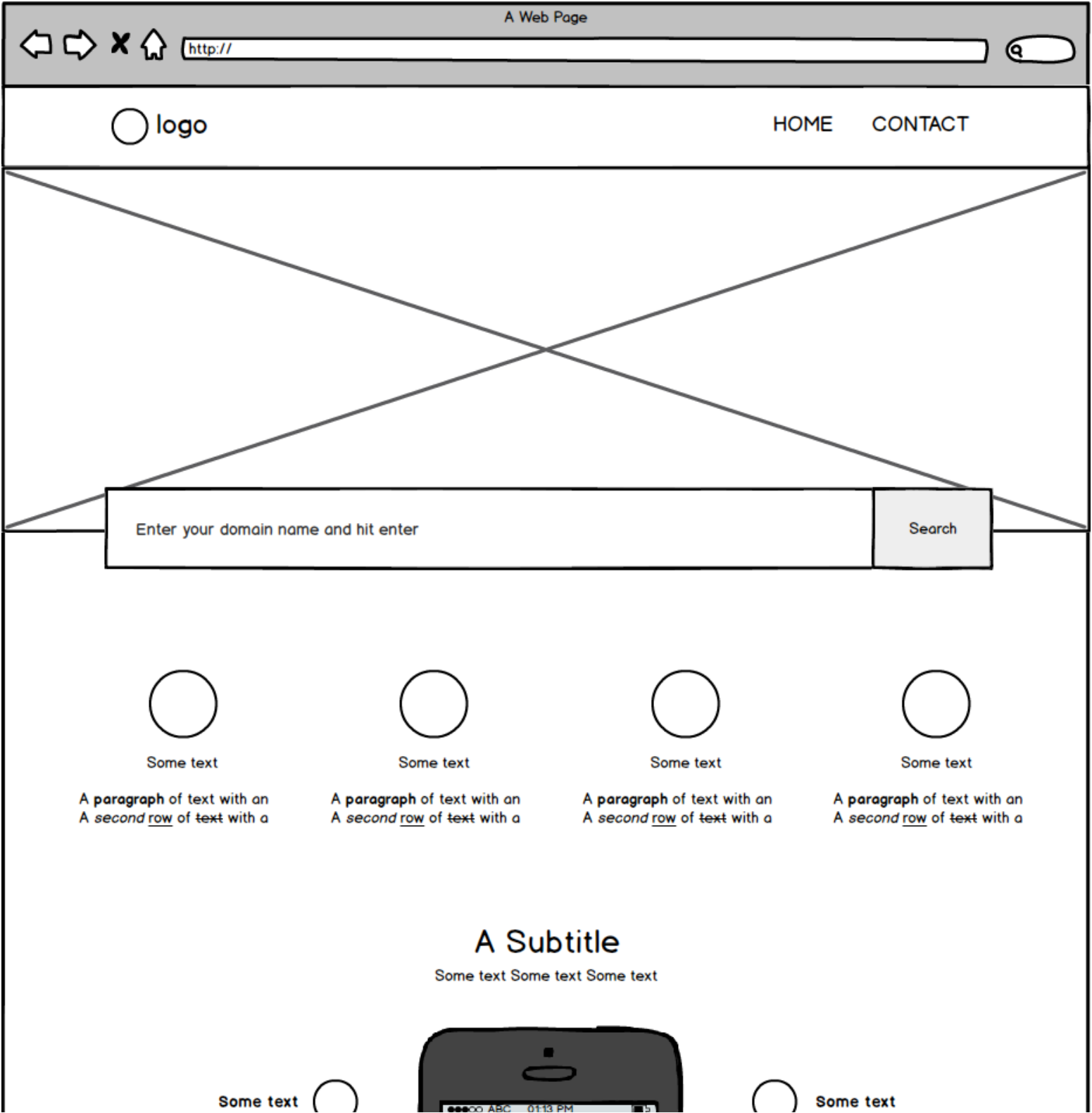
También puedes especificar qué codecs requiere el archivo multimedia; de esta forma el navegador tomará decisiones más inteligentes:

```
<video controls>
  <source src="foo.ogg" type="video/ogg; codecs=dirac, speex">
  Tu navegador no implementa el elemento <code>video</code>.
</video>
```

# Trabajo Práctico 2 HTML

## Descripción:

- 1. Seguir las indicaciones del mockup.
- 2. Respetar la estructura semántica del TP1 y agregar los nuevos elementos html5 vistos en clase.
- 3. Maquetar el formulario con sus elementos correspondientes.



asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs

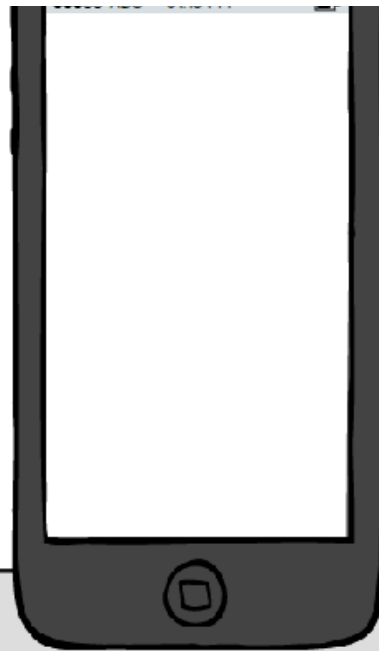
asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs

Some text  
asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs

Some text  
asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs

Some text  
asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs

Some text  
asdasd asdjfhasdj  
asldkjfhasdjkl asklfhs



## A Subtitle

Some text Some text Some text

Some text

\$95

Some text 1

Some text 1

Some text 1

Some text 1

Button

Some text

\$95

Some text 1

Some text 1

Some text 1

Some text 1

Button

Some text

\$95

Some text 1

Some text 1

Some text 1

Some text 1

Button

Some text

\$95

Some text 1

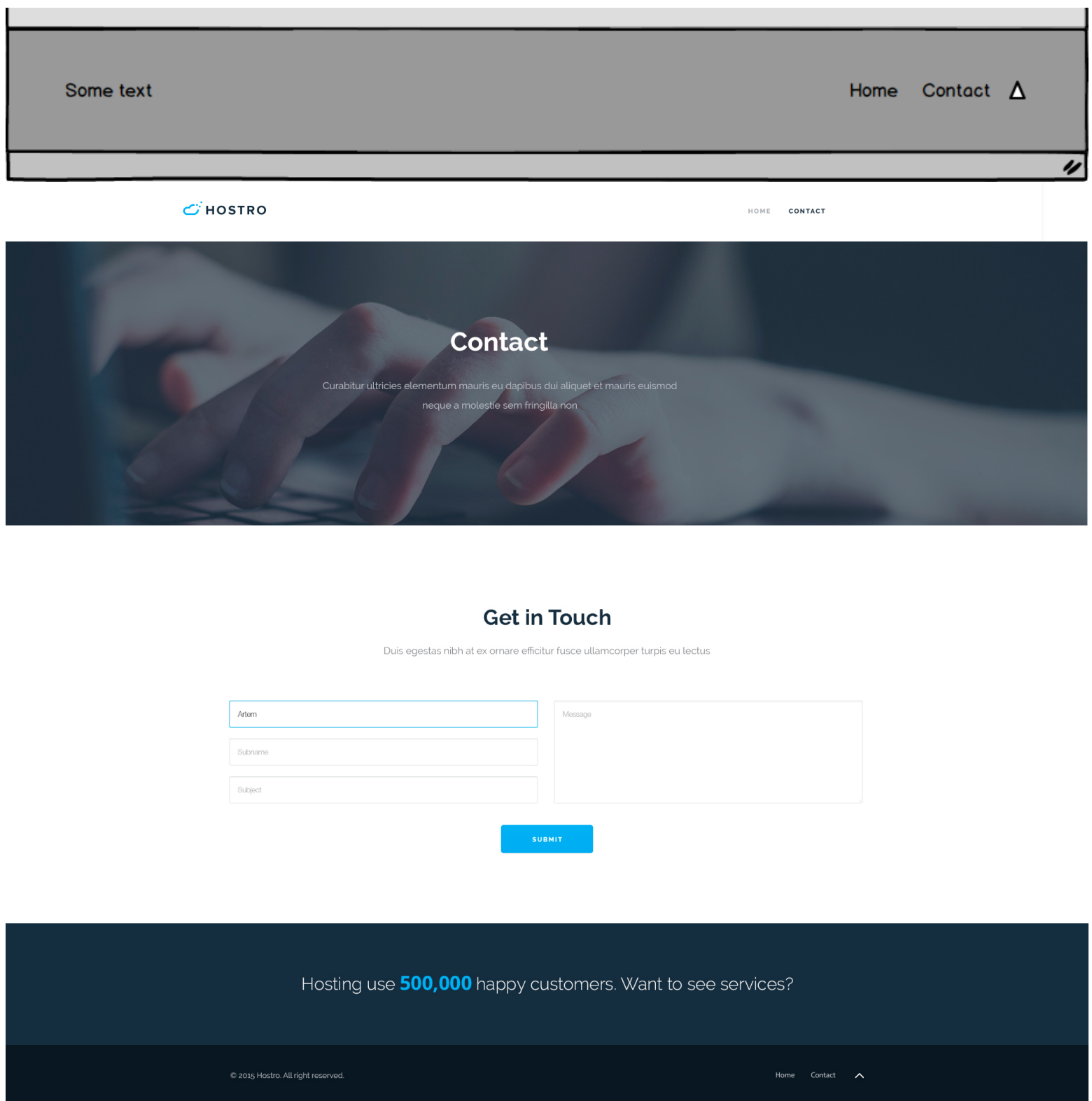
Some text 1

Some text 1

Some text 1

Button

Hosting use 500,000 happy customers. Want to see services?



## Objetivo:

Aprender a crear secciones básicas de cualquier sitio web moderno en html5.

## Requerimientos:

1. Haber realizado los trabajos anteriores.
2. Indentar correctamente.

3. Respetar la estructura semántica
4. Agregar los texto del mockup.  
Subir la resolución de tu trabajo a tu repositorio de Github en la subcarpeta “HTML TP2”

## **Bonus:**

**Conviértete en un Ninja de la Productividad: Aprende a crear etiquetas html con los superpoderes de emmet!**

1. [Instalando Emmet:](#)
2. [Conviertete en un verdadero ninja](#)

## **Recursos:**

3. [Elementos Heading](#)
4. [Elemento Nav.](#)
5. [Elemento Table.](#)
6. [Formularios en HTML](#)
7. [Cómo estructurar un formulario en HTML5.](#)