PRÁCTICA N°1 - HTML



Cátedra: Entornos gráficos

4º año Ingeniería en sistemas de información

**Profesores:**

Butti, Julián

Diaz, Daniela

**Integrantes:**

Roma, Daiana - 43673 - [dairoma5@gmail.com](mailto:dairoma5@gmail.com)

Trilla, Melody - 43715 - trillamelo@gmail.com

Zancocchia, Camila - 43671 - [camila.zancocchia@gmail.com](mailto:camila.zancocchia@gmail.com)

# 

**Contenido**

[**Ejercitación 1**](#_b0bbjx3ufdgk) **3**

[**Ejercitación 2**](#_a68s4tqcv04p) **7**

[2.a)](#_bex426fqjbst) 7

[2.b)](#_pp135ltis789) 7

[2.c)](#_kf3i0e7g3yw3) 8

[2.d)](#_a0uovrmd6syk) 8

[2.e)](#_xwod6gcor2uu) 9

[2.f)](#_tychs6x1kc53) 9

[**Ejercitación 3**](#_1efzxa7hmkii) **11**

[3.a)](#_jfxhzxh3xxai) 11

[3.b)](#_c4uxygqq6794) 11

[3.c)](#_fnfuapuyb58l) 12

[3.d)](#_5uih32n5khgt) 13

[3.e)](#_70md8k543um) 14

[3.f)](#_nspl63bz4erz) 15

[3.g)](#_obvsrzk53gyd) 16

[3.h)](#_1k43a56nh3sw) 17

[3.i)](#_iyxp7o2058r2) 19

[3.j)](#_w0s5s76qod2j) 19

[3.k)](#_ppmy5djf0477) 20

# 

# 

# Ejercitación 1

**1- Qué es HTML, cuándo fue creado, cuáles fueron las distintas versiones y cuál es la última?**

HTML (HyperText Markup Language) es un lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones. Define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

Durante sus primeros cinco años (1990-1995), HTML pasó por una serie de revisiones y experimentó una serie de extensiones, alojadas principalmente en el CERN (Organización Europea para la Investigación Nuclear), y luego en el IETF (Internet Engineering Task Force). Con la creación del W3C, el desarrollo de HTML cambió de lugar de nuevo. Un primer intento frustrado de extender el HTML en 1995 conocido como HTML 3.0 luego dio paso a un enfoque más pragmático conocido como HTML 3.2, que se completó en 1997. HTML 4.01 siguió rápidamente más tarde ese mismo año. Al año siguiente, la membresía del W3C decidió dejar de evolucionar HTML y en su lugar comenzó a trabajar en un equivalente basado en XML, llamado XHTML. Este esfuerzo comenzó con una reformulación de HTML 4.01 en XML, conocido como XHTML 1.0, que no agregó nuevas funciones, excepto la nueva serialización, y que se completó en 2000. Después de XHTML 1.0, el enfoque del W3C se volvió más fácil para otros grupos de trabajo para extender XHTML, bajo el título de XHTML Modularization.

Paralelamente, el W3C también trabajó en un nuevo lenguaje que no era compatible con los lenguajes HTML y XHTML anteriores, llamándolo XHTML 2.0. Alrededor de la época en que la evolución de HTML se detuvo en 1998, parte de la API para HTML desarrollada por los proveedores de navegadores se especificaron y publicaron bajo el nombre DOM Level 1 (en 1998) y DOM Level 2 Core y DOM Level 2 HTML (comenzando en 2000 y culminando en 2003). Estos esfuerzos luego se agotaron, con algunas especificaciones DOM nivel 3 publicadas en 2004, pero el grupo de trabajo se cerró antes de que se completaran todos los borradores de nivel 3.

En 2003, la publicación de XForms, una tecnología que se posicionó como la próxima generación de formularios web, despertó un renovado interés en la evolución del HTML en sí mismo, en lugar de encontrar reemplazos para él. Este interés surgió de la constatación de que la implementación de XML como tecnología web se limitaba a tecnologías completamente nuevas (como RSS y Atom más tarde), en lugar de un reemplazo de las tecnologías implementadas existentes (como HTML).

Una prueba de concepto para demostrar que era posible extender los formularios de HTML 4.01 para proporcionar muchas de las funciones que introdujo XForms 1.0, sin requerir que los navegadores implementaran motores de procesamiento que fueran incompatibles con las páginas web HTML existentes, fue el primer resultado de este interes renovado.

En esta etapa inicial, aunque el borrador ya estaba disponible públicamente, y ya se estaban solicitando comentarios de todas las fuentes, la especificación solo estaba bajo los derechos de autor de Opera Software. La idea de que la evolución de HTML debería reabrirse se probó en un taller W3C en 2004, donde se presentaron algunos de los principios que subyacen al trabajo HTML, así como el anteproyecto de propuesta antes mencionado que cubre solo las características relacionadas con formularios. el W3C conjuntamente por Mozilla y Opera. La propuesta fue rechazada debido a que la propuesta entraba en conflicto con la dirección previamente elegida para la evolución de la Web; el personal y la membresía del W3C votaron para continuar desarrollando sustitutos basados en XML en su lugar.

Poco después, Apple, Mozilla y Opera anunciaron conjuntamente su intención de continuar trabajando en el esfuerzo bajo el paraguas de un nuevo lugar llamado WHATWG. Se creó una lista de correo pública y el borrador se movió al sitio WHATWG. El derecho de autor se modificó posteriormente para ser propiedad conjunta de los tres proveedores, y para permitir la reutilización de la especificación. El WHATWG se basó en varios principios básicos, en particular, que las tecnologías deben ser compatibles con versiones anteriores, que las especificaciones y las implementaciones deben coincidir incluso si esto significa cambiar la especificación en lugar de las implementaciones, y que las especificaciones deben ser lo suficientemente detalladas como para que las implementaciones interoperabilidad completa sin ingeniería inversa entre sí. Este último requisito en particular requiere que el alcance de la especificación HTML incluya lo que se había especificado previamente en tres documentos separados: HTML 4.01, XHTML 1.1 y DOM Nivel 2 HTML. También significó incluir muchos más detalles de los que se habían considerado anteriormente como la norma. En 2006, el W3C indicó su interés en participar en el desarrollo de HTML 5.0 después de todo, y en 2007 formó un grupo de trabajo autorizado para trabajar con WHATWG en el desarrollo de la especificación HTML. Apple, Mozilla y Opera permitieron al W3C publicar la especificación bajo los derechos de autor del W3C, manteniendo una versión con la licencia menos restrictiva en el sitio WHATWG. Durante varios años, ambos grupos trabajaron juntos bajo el mismo editor: Ian Hickson. En 2011, los grupos llegaron a la conclusión de que tenían diferentes objetivos: el W3C quería trazar una línea en la arena para las características de una Recomendación HTML 5.0, mientras que WHATWG quería continuar trabajando en un estándar de vida para HTML, manteniendo continuamente la especificación y agregando nuevas características. A mediados de 2012, se presentó un nuevo equipo de edición en el W3C para que se encargue de crear una Recomendación HTML 5.0 y preparar un Borrador de Trabajo para la próxima versión HTML.

Desde entonces, W3C Web Platform WG ha estado recogiendo parches del WHATWG que resolvió errores registrados en la especificación W3C HTML o representó con mayor precisión la realidad implementada en los agentes de usuario. Hasta ahora los parches de la especificación HTML WHATWG se fusionaron hasta el 12 de enero de 2016. Los editores HTML del W3C también han agregado parches que resultaron de discusiones y decisiones tomadas por el W3C Web Platform WG, así como correcciones de errores de errores no compartidos por el WHATWG. HTML5 es el principal lenguaje de la web. Es uno de los primeros lenguajes que debes dominar si deseas dedicarte al diseño y desarrollo web.

**2. ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos con HTML?**

El objetivo de W3C es garantizar una Web universalmente accesible, más allá de las diferencias de idioma, navegador, sistema operativo, plataforma, localización geográfica o aptitudes tecnológicas.

Para lograr esta meta, el W3C propone a diseñadores, programadores, editores y empresas de software dedicadas a web browsers el cumplimiento de ciertas directivas de desarrollo. El W3C ofrece pautas sobre la mayoría de los lenguajes y tecnologías de uso común en la construcción de páginas y aplicaciones web, como HTML, XHTML, CSS, XML y sus distintas versiones, entre muchas otras.

**Perceptible**

Perceptible, se refiere que tanto los contenidos como la interfaz de un sitio web debe poder ser percibida por todos sus usuarios.

Los contenidos audiovisuales, por ejemplo, deben tener alternativas perceptibles para personas ciegas o sordas. Existen opciones como la audiodescripción de videos, subtítulos y lengua de señas para hacer accesible el contenido.

Además, la interfaz, imágenes, botones, reproductores de videos y otros elementos deben ser accesibles, perfectamente identificables y ejecutables por cualquier persona en cualquier situación, dispositivo y sistema operativo.

El color también forma parte del grupo de indicadores relevantes, ya que existe un porcentaje de la población que posee alguna condición que no puede percibirlos correctamente. Por ejemplo, si un formulario de contacto requiere que se completen de forma obligatoria los ítem de color rojo, es un impedimento para aquellos que no tienen la capacidad de percibir los colores de la forma correcta.

**Operable**

Esto quiere decir que un sitio web debe tener muchas maneras, y todas muy claras y advertidas, para realizar una acción o buscar un contenido. Entre más alternativas existan, mejor será su accesibilidad.

El que un sitio web sea operable según las reglas de la W3C se asocia a lo siguiente:

Asegurar toda su funcionalidad en base al teclado: Este punto es clave, ya que todo el mundo está acostumbrado a navegar los sitios web con el ratón, pero si éste falla en algún minuto, o un grupo de la población no puede usarlo por distintas razones, la navegación se hace casi imposible.

Dar un tiempo prudente para leer y entender el contenido: muchos sitios usan herramientas como los "slider", que van cambiando rápidamente sus contenidos (sea texto o imágenes), como en un carrusel. Los tiempos en los que la información cambia deben ser lo suficientemente estudiados, como para que todos los potenciales usuarios, independientemente de su condición o edad, puedan comprenderlos.

No diseñar contenidos o herramientas que puedan provocar convulsiones: Existen herramientas de diseño web que son preciosas, que cambian rápidamente de formas y colores, que le dan a un sitio un verdadero impacto, pero muchas de ellas pueden provocar ataques o convulsiones epilépticas, que pueden ser prevenidas.

Proporcionar medios para una navegación expedita: tanto en sitios web como en aplicaciones, a veces, menos es más, si es que se trata de hacer un sitio intuitivo, pero también es necesario que "todos los caminos lleven a Roma".

**Comprensible**

Para que un sitio sea considerado comprensible, debe tomar en cuenta los siguientes elementos:

Ser Legible y comprensible: Se refiere tanto a la forma, como el fondo de los textos de un sitio web. En su forma, deben poseer un tipo de letra que pueda ser leído por todos los usuarios, y si en el sitio existe una herramienta que permita al visitante ajustar el tipo de letra a su mejor tamaño, mucho mejor. En su fondo, se debe contemplar el correcto uso de abreviaturas, modismos, neologismos, para que sean comprendidos por todos. Hay que recordar que los sitios web pueden ser visitados en casi todo el mundo, y aunque un sitio web tenga por misión una cobertura local, no se debe dejar de lado lo global de las comunicaciones, y el uso de palabras que no puedan ser entendidas, debe tener una solución.

Predecible: un sitio web o aplicación, mientras más predecible sea, mucho mejor. Esto no tiene nada que ver con el concepto anticipable, que tiene que ver con las sorpresas que pueda tener un sitio web a sus usuarios. El concepto predecible, tiene relación con el funcionamiento de un sitio, con que sus potenciales usuarios no pierdan tiempo en tratar de adivinar para qué funciona una herramienta u otra, todo debe ser expedito para una mejor experiencia de navegación.

**Robusto**

Robusto se refiere a que los sitios web o aplicaciones deben ser compatibles con todos los navegadores web (Internet Explorer, Mozilla, Chrome, Safari, entre otros), en todos los sistemas operativos (Windows, Mac, Android, iOs, etc), y en todos los dispositivos (computadoras, teléfonos móviles, tabletas, e incluso televisores), pero también, con aplicaciones de tecnologías de apoyo, como lectores de pantallas para personas ciegas como lo son JAWS, NVDA, talkback, VoiceOver.

En palabras simples, un desarrollador debe respetar las normas sobre HTML y de las hojas de estilo, para que su sitio pueda ser visitado en cualquier tipo de plataforma. Entonces, si en código se escribe un enlace o un encabezado, una lista, una imagen, o lo que sea, se deben respetar los atributos de cada cual.

**3. En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y obsoleto?**

**Desaprobado**

Un elemento o atributo desaprobado es aquel que ha quedado anticuado por la presencia de estructuras nuevas. Los elementos desaprobados se definen en el manual de referencia en los lugares apropiados, pero claramente marcados como desaprobados. Los elementos desaprobados pueden declararse obsoletos en versiones futuras de HTML.

Los agentes de usuario deberían seguir dando soporte a los elementos desaprobados por razones de compatibilidad con versiones anteriores.

Las definiciones de elementos y atributos indican claramente cuáles son desaprobados.

Esta especificación incluye ejemplos que ilustran cómo evitar el uso de elementos desaprobados. En la mayoría de los casos estos ejemplos dependen de que el agente de usuario soporte hojas de estilo. En general, los autores deberían usar hojas de estilo para lograr efectos estilíticos y de formato de presentación en lugar de utilizar atributos presentacionales de HTML. Los atributos de presentación de HTML han sido desaprobados cuando existen alternativas con hojas de estilo

**Obsoleto**

Un elemento o atributo obsoleto es aquél para el cual no hay garantía de soporte por parte de un agente de usuario. Los elementos obsoletos han dejado de estar definidos en la especificación, pero se enumeran por motivos históricos en la [sección de cambios](http://html.conclase.net/w3c/html401-es/appendix/changes.html) del manual de referencia.

**4. ¿Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?**

Una definición de tipo de documento o DTD ( documento tipo definición) es una descripción de estructura y sintaxis de un documento [XML](https://es.wikipedia.org/wiki/XML) o [SGML](https://es.wikipedia.org/wiki/SGML). Su función básica es la descripción de la estructura de datos, para usar una estructura común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, dichos documentos pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento.

Un documento HTML válido declara qué versión de HTML se utiliza en el documento. La declaración del tipo de documento especifica la definición del tipo de documento (DTD) que se usa en el documento.

HTML 4.01 especifica tres DTDs, de modo que los autores deben incluir una de las siguientes declaraciones del tipo de documento en sus documentos. Los DTDs varían en cuanto a los elementos que soportan.

* El [DTD HTML 4.01 Estricto](http://html.conclase.net/w3c/html401-es/sgml/dtd.html) (Strict DTD) incluye todos los elementos y atributos que no han sido [desaprobados](http://html.conclase.net/w3c/html401-es/conform.html#deprecated) o que no aparecen en documentos con marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:  
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">
* El [DTD HTML 4.01 Transicional](http://html.conclase.net/w3c/html401-es/sgml/loosedtd.html) (Transitional DTD) incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados (la mayoría de los cuales están relacionados con la presentación visual). Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:  
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
* El [DTD HTML 4.01 para Documentos con Marcos](http://html.conclase.net/w3c/html401-es/sgml/framesetdtd.html) (Frameset DTD) incluye todo lo que incluye el DTD Transicional más los marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:  
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

**5. Qué son los metadatos y cómo se especifican en HTML?**

Los Metadatos son elementos HTML que muestran información sobre la propia página web que los contiene.

Son usados por los buscadores para definir la información principal de nuestra web (temática, descripción), por lo que será muy importante que lo tengamos correctamente configurados.

Estos elementos meta se encontrarán en el head.

El HTML permite a los autores especificar metadatos -- información sobre un documento más que contenido del propio documento -- de diferentes de maneras.

Por ejemplo, para especificar el autor de un documento, puede utilizarse el elemento [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META) como sigue:

<META name="Author" content="Dave Raggett">

El elemento [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META) especifica una propiedad (aquí "Author") y le asigna un valor (aquí "Dave Raggett").

Esta especificación no define un conjunto de propiedades legales de metadatos. El significado de una propiedad y el conjunto de valores legales para esa propiedad debería estar definida en un diccionario de referencia llamado [perfil](http://html.conclase.net/w3c/html401-es/struct/global.html#profiles). Por ejemplo, un perfil diseñado para ayudar a los motores de búsqueda a indexar documentos podría definir propiedades tales como "author", "copyright", "keywords", etc.

**Especificación de metadatos**

En general, la especificación de metadatos implica dos pasos:

1. Declaración de una propiedad y de un valor para esta propiedad. Esto puede hacerse de dos maneras:
   1. Desde dentro de un documento, por medio del elemento [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META).
   2. Desde fuera de un documento, vinculando los metadatos por medio del elemento [LINK](http://html.conclase.net/w3c/html401-es/struct/links.html#edef-LINK)
2. Referencia a un [perfil](http://html.conclase.net/w3c/html401-es/struct/global.html#profiles) en el que se definen la propiedad y sus valores legales. Para designar un perfil, se usa el atributo [profile](http://html.conclase.net/w3c/html401-es/struct/global.html#adef-profile) del elemento [HEAD](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-HEAD).

Obsérvese que al estar definido un perfil por el elemento [HEAD](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-HEAD), se aplica el mismo perfil a todos los elementos [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META) y [LINK](http://html.conclase.net/w3c/html401-es/struct/links.html#edef-LINK) de la cabecera del documento.  
Los agentes de usuario no necesitan soportar los mecanismos de metadatos. Para aquellos que opten por soportar metadatos, esta especificación no define cómo deberían interpretarse los metadatos.

# Ejercitación 2

HTML está compuesto por un conjunto de elementos que son la base de su estructura. Los elementos están compuestos por dos tags (el de apertura y el de cierre) y el contenido

en el medio (con excepción de los elementos vacíos ). Cada tag puede tener atributos (proporcionan ciertas características como altura, ancho, color, etc.) y eventos (asocian un script que se ejecuta cuando el evento ocurre).

Analizar los siguientes segmentos de código indicando en qué sección del documento HTML se colocan, cuál es el efecto que producen y señalar cada uno de los elementos, etiquetas, y atributos (nombre y valor), aclarando si es obligatorio.

## 2.a)

<!-- Código controlado el día 12/08/2009 →

Se utiliza para escribir comentarios.

**Elemento**: <!-- Código controlado el día 12/08/2009 →

**Etiqueta de apertura:**<!--

**Etiqueta de cierre:** -->

## 2.b)

<div id="bloque1">Contenido del bloque1</div>

**Elemento**: <div id="bloque1">Contenido del bloque1</div>

**Etiqueta de apertura:**<div id="bloque1">

**Etiqueta de cierre:** </div>

**Atributos:** id="bloque1"

**Nombre:** id

**Valor:** bloque1

## 2.c)

<img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32" longdesc="detalles.htm" />

**Elemento**: <img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32" longdesc="detalles.htm" />

**Etiqueta de apertura:** <img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32" longdesc="detalles.htm" />

**Etiqueta de cierre:** no hay etiqueta de cierre

**Atributos:** src=""

**Nombre:** src

**Valor:**

**Atributos:** alt="lugar imagen"

**Nombre:** alt

**Valor:** lugar imagen

**Atributos:** id="im1"

**Nombre:** id

**Valor:** im1

**Atributos:** name="im1"

**Nombre:**  name

**Valor:**  im1

**Atributos:** width="32"

**Nombre:**  width

**Valor:**  32

**Atributos:** height="32"

**Nombre:**  height

**Valor:**  32

**Atributos:** longdesc="detalles.htm"

**Nombre:**  longdesc

**Valor:**  detalles.htm

## 2.d)

<meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />

**Elemento**: <meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

**Etiqueta de apertura:**<meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

**Etiqueta de cierre:**

**Atributos:** name="keywords"

**Nombre:**name

**Valor:**keywords

**Atributos:**lang="es"

**Nombre:**lang

**Valor:** es

**Atributos:** content="casa, compra, venta, alquiler "

**Nombre:** content

**Valor:** casa, compra, venta, alquiler

**Elemento**: <meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />

**Etiqueta de apertura:**<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />

**Etiqueta de cierre:**

**Atributos:** http-equiv="expires"

**Nombre:** http-equiv

**Valor:** expires

**Atributos:** content="16-Sep-2019 7:49 PM"

**Nombre:** content

**Valor:** 16-Sep-2019 7:49 PM

## 2.e)

<a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es" charset="utf-8" rel="help">Resumen HTML </a>

**Elemento**: <a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es" charset="utf-8" rel="help">Resumen HTML </a>

**Etiqueta de apertura:** <a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es" charset="utf-8" rel="help">

**Etiqueta de cierre:**</a>

**Atributos:** href="http://www.e-style.com.ar/resumen.html"

**Nombre:** href

**Valor:** http://www.e-style.com.ar/resumen.html

**Atributos:**  type="text/html"

**Nombre:** type

**Valor:** text/html

**Atributos:** hreflang="es"

**Nombre:**hreflang

**Valor:** es

**Atributos:** charset="utf-8"

**Nombre:**charset

**Valor:** utf-8

**Atributos:** rel="help"

**Nombre:**rel

**Valor:** help

## 2.f)

<table width="200" summary="Datos correspondientes al ejercicio vencido">

<caption align="top"> Título </caption>

<tr>

<th scope="col">&nbsp;</th>

<th scope="col">A</th>

<th scope="col">B</th>

<th scope="col">C</th>

</tr>

<tr>

<th scope="row">1º</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<th scope="row">2º</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

**Elemento**: <th scope="col">.......</th>

**Etiqueta de apertura:** <th>

**Etiqueta de cierre:** </th>

**Atributos:** scope="col"

**Nombre:**  scope

**Valor:**  col

**Elemento**: <tr>....</tr>

**Etiqueta de apertura:** <tr>

**Etiqueta de cierre:** </tr>

**Elemento**: <caption align="top"> Título </caption>

**Etiqueta de apertura:** <caption>

**Etiqueta de cierre:** </caption>

**Atributos:**  align="top"

**Nombre:** align

**Valor:** top

**Elemento**: <table width="200" summary="Datos correspondientes al ejercicio vencido">....</table>

**Etiqueta de apertura:** <table>

**Etiqueta de cierre:** </table>

**Atributos:**  width="200"

**Nombre:** width

**Valor:** 200

**Atributos:** summary="Datos correspondientes al ejercicio vencido"

**Nombre:** summary

**Valor:** Datos correspondientes al ejercicio vencido

# Ejercitación 3

En cada caso, explicar las diferencias entre los segmentos de código y sus visualizaciones

## 3.a)

<a href="http://www.google.com.ar">Click aquí para ir a Google</a>

Muestra un hipervínculo: “Click aquí para ir a Google”, el cual te direcciona a la página de Google al hacer click en él. Al abrirse Google, lo hace en la misma pestaña en la cual estaba el hipervínculo.

<a href="http://www.google.com.ar" target="\_blank">Click aquí para ir a Google</a>

A diferencia del segmento anterior, aquí se agrega el atributo target="\_blank", lo cual hace que la página de Google se abra en una pestaña diferente.

<a href="http://www. google.com.ar" type="text/html" hreflang="es" charset="utf-8" rel="help">

En este caso no se visualiza nada, es decir, la página se ve en blanco, esto se debe a que se abre la etiqueta, pero luego no hay contenido ni el cierre de la misma.

<a href="#">Click aquí para ir a Google</a>

Se visualiza un hipervínculo: “Click aquí para ir a Google”, pero al hacer clic en el mismo, no nos direcciona a la página de Google, debido a que en el atributo href, no se especifica la dirección url de dicha página, sino que se coloca un “#”, el cual sirve para movernos a diferentes partes de la misma página.

<a href="#arriba">Click aquí para volver arriba</a>

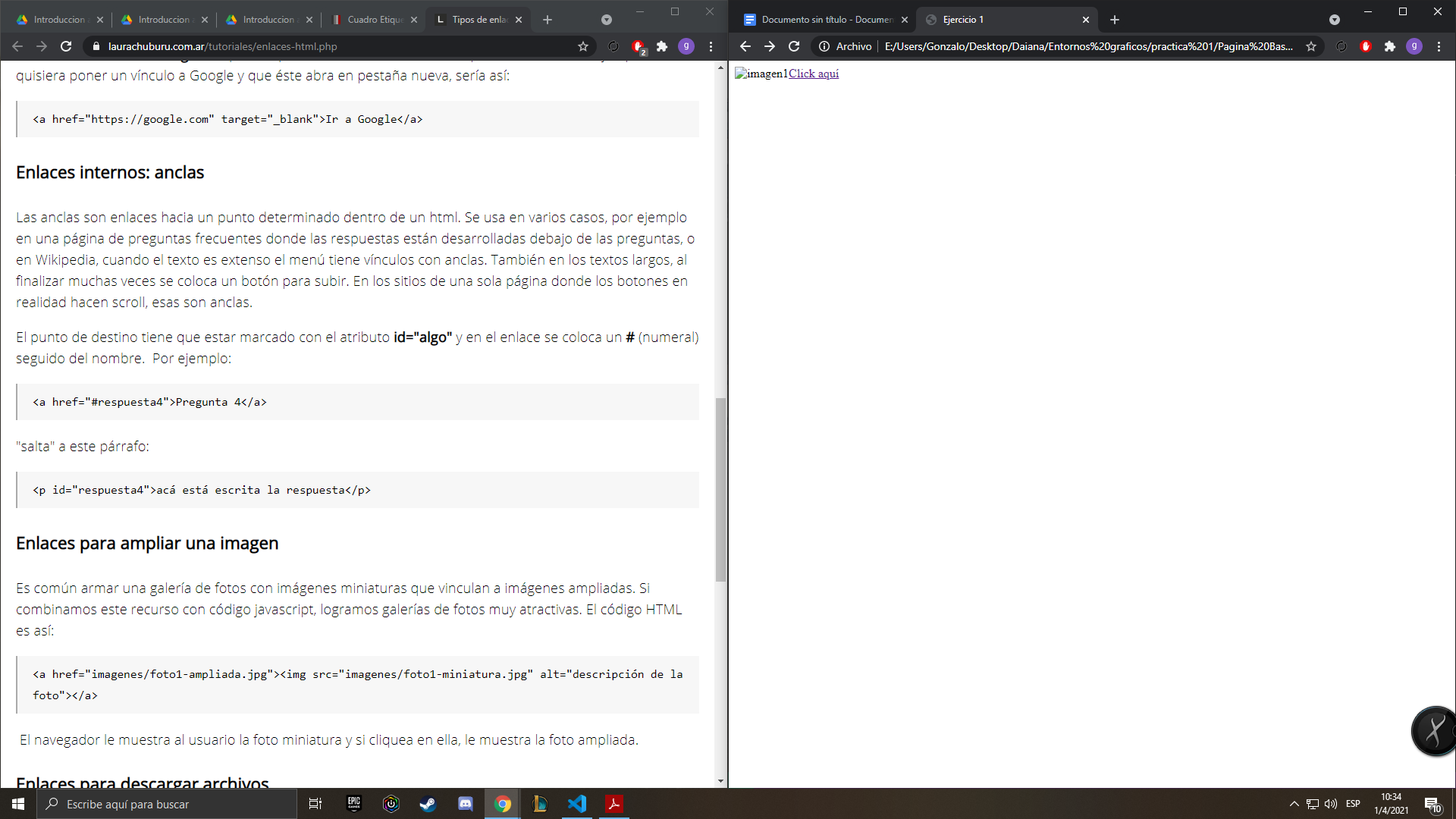
En este caso se visualiza un hipervínculo: “Click aquí para volver arriba”. Sin embargo, el atributo href="#arriba", requiere que exista otro segmento de código donde tenga un atributo ..id=”arriba”.., para que al hacer click en el hipervínculo nos direccione a esa parte de la página.

<a name="arriba" id="arriba"></a>

No se visualiza nada ya que entre las etiquetas de apertura y cierre no hay contenido.

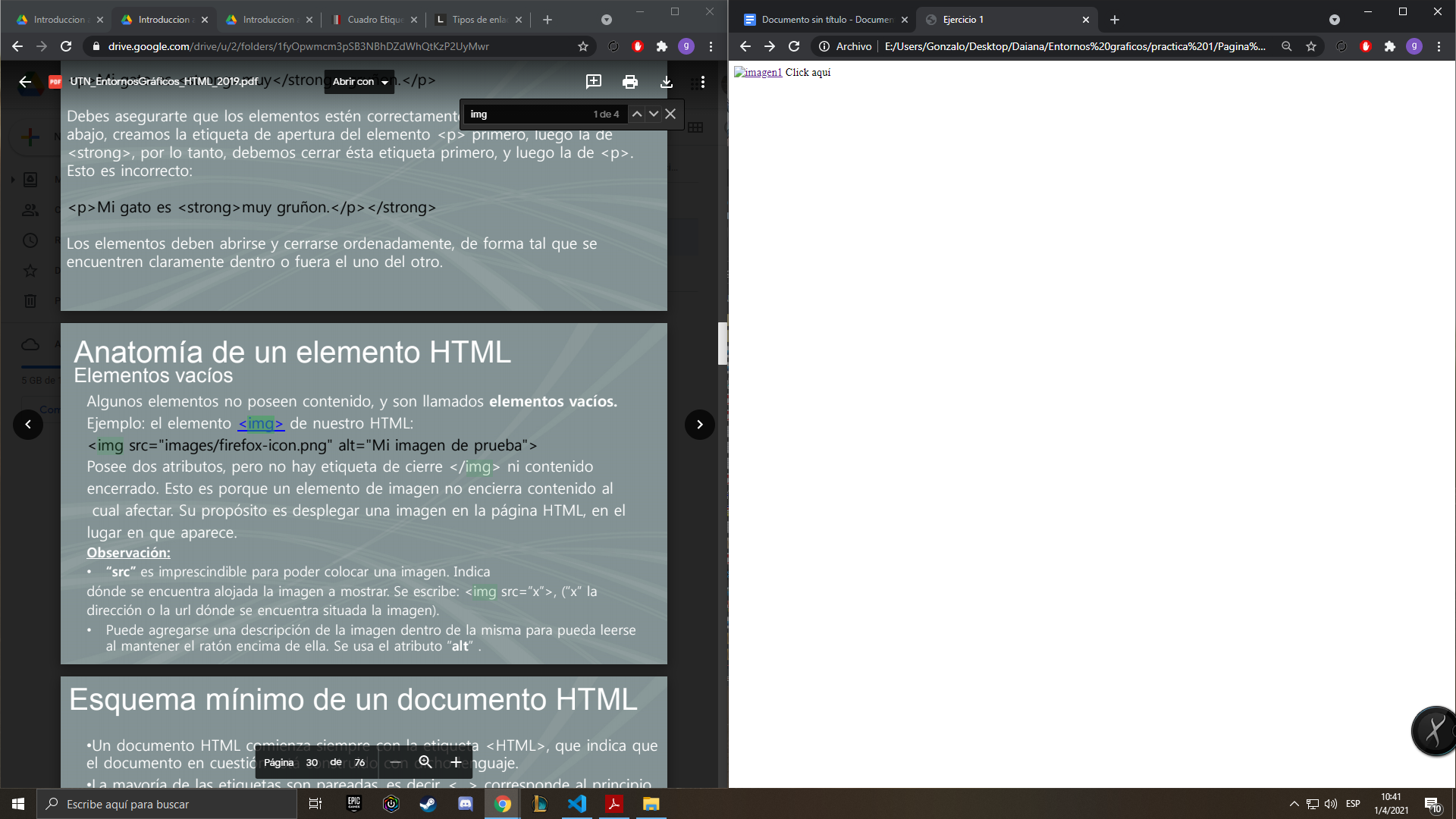
## 3.b)

<p><img src="im1.jpg" alt="imagen1" /><a href="http://www.google.com.ar">Click aquí</a></p>

Se visualiza: 

Dentro de la etiqueta <p>..</p> se coloca una etiqueta <img> y una etiqueta <a>..<a/>. En este caso, el contenido de la utlima es: Click aquí, el cual funciona como enlace para ingresar a la página de Google.

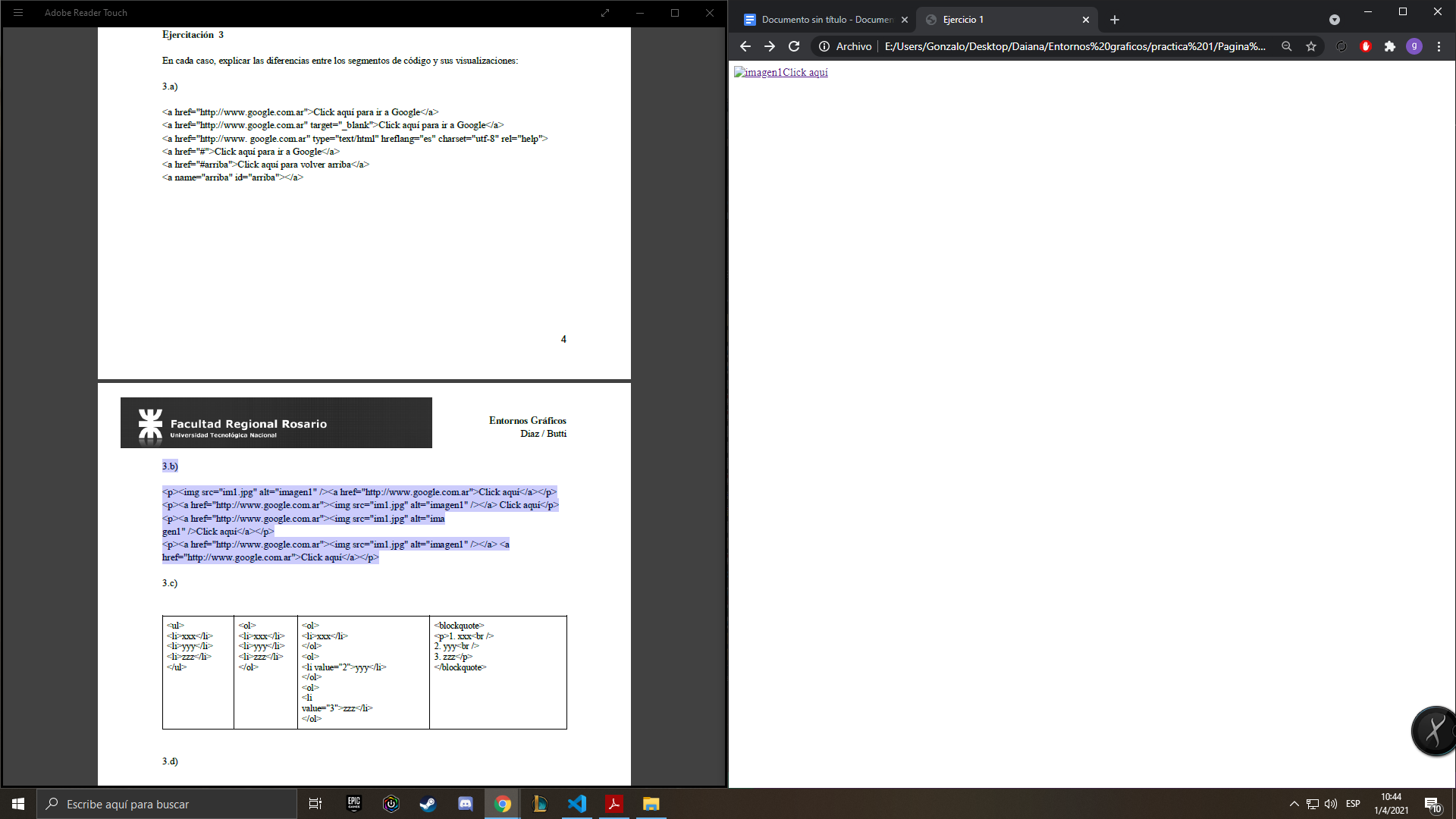
<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> Click aquí</p>

Se visualiza: 

En este caso, el enlace para ingresar a la página de Google se encuentra en la imagen, esto se debe a que dentro de la etiqueta <a>..<a/> se colocó la imagen. Mientras que: Click aquí, está fuera de la misma etiqueta, por lo que no se genera el enlace en la frase.

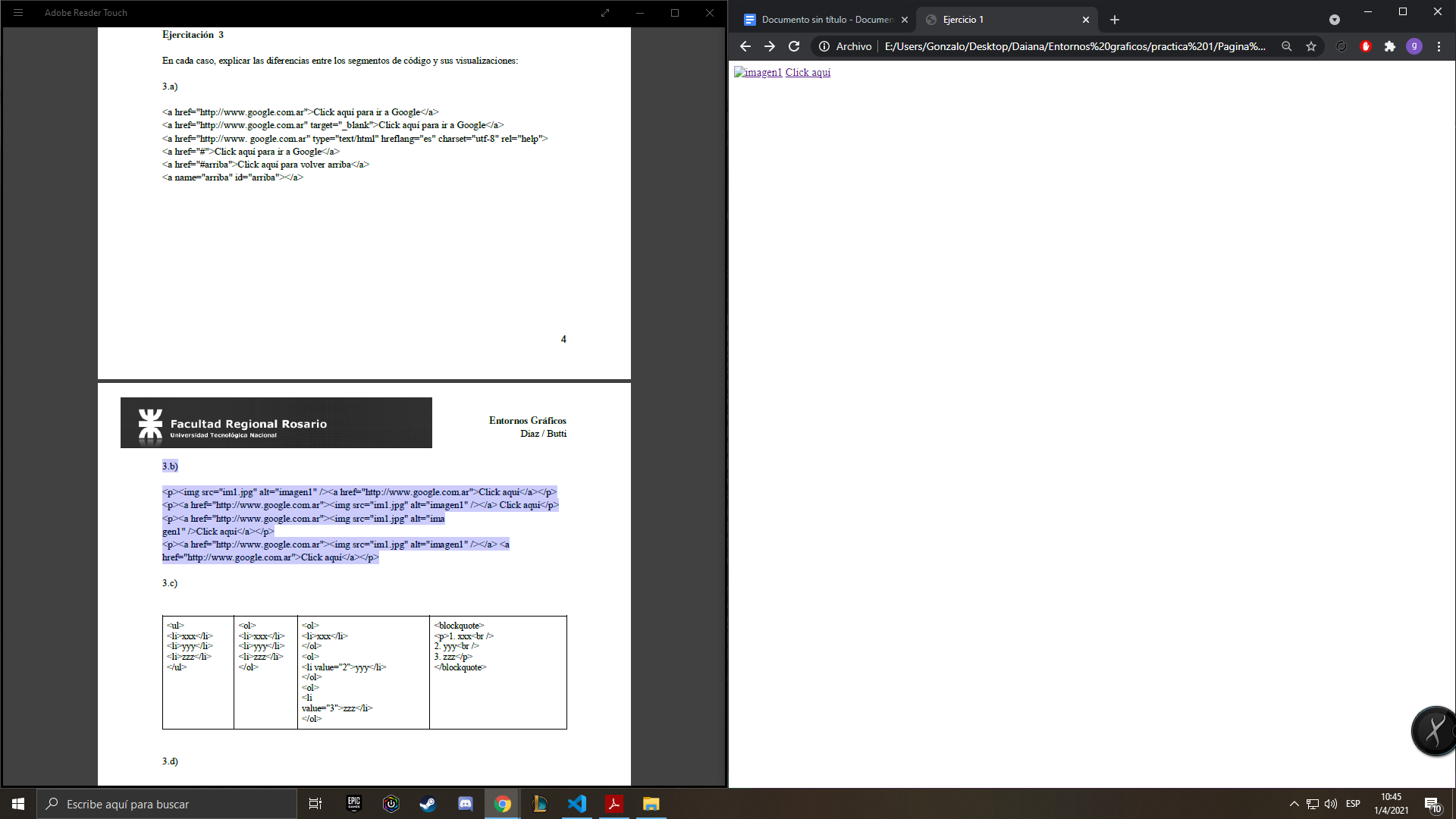
<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="ima

gen1" />Click aquí</a></p>

Se visualiza:

En este caso, tanto la imagen como Click aqui son enlaces para ingresar a Google ya que ambos se encuentran dentro de la etiqueta <a>..<a/>

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> <a href="http://www.google.com.ar">Click aquí</a></p>

Se visualiza: 

A diferencia del ejemplo anterior, aquí se utiliza dos veces la etiqueta <a>..<a/>, una vez para colocar la imagen y la otra para: Click aquí. Esto genera dos enlaces diferentes, aunque te direccionan a la misma página ya que en el atributo href se especifica para ambos casos, la misma dirección.

## 3.c)

<ul>

<li>xxx</li>

<li>yyy</li>

<li>zzz</li>

</ul>

Se visualiza:



Al utilizarse la etiqueta ul se crea una lista no ordenada

<ol>

<li>xxx</li>

<li>yyy</li>

<li>zzz</li>

</ol>

Se visualiza:



A diferencia del anterior, se utiliza la etiqueta ol, la cual corresponde a listas numeradas.

<ol>

<li>xxx</li>

</ol>

Se visualiza:



A diferencia del anterior, aquí se visualiza solo la primera fila.

<ol>

<li value="2">yyy</li>

</ol>

<ol>

<li

value="3">zzz</li>

</ol>

Se visualiza:



En este caso se hacen dos listas ordenadas y se utiliza el atributo “value” dentro de cada etiqueta “li”, para especificar el número de fila

<blockquote>

<p>1. xxx<br />

2. yyy<br />

3. zzz</p>

</blockquote>

Se visualiza:



En este caso no se utilizan listas, sino que las etiquetas <p>..</p> y <br>-

La etiqueta <P>...</P> inserta una salto de línea y una línea en blanco, por lo que, en la práctica, equivale a un nuevo párrafo.

En cambio la etiqueta <BR> inserta un salto de línea, pero no un salto de párrafo.

La etiqueta <blockquote>...</blockquote> se utiliza para insertar citas en forma de bloques de contenido, usualmente compuestas por un párrafo, un grupo de párrafos o un conjunto de muchos otros elementos, esto genera que el texto se tabule hacia la derecha.

## 3.d)

<table border="1" width="300">

<tr>

<th>Columna 1</th>

<th>Columna 2</th>

</tr>

<tr>

<td>Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

</table>

Se visualiza:



La etiqueta <table> </table> define el comienzo y final de la tabla.

<tr> </tr> define el comienzo y el final de una fila de la tabla.

<td> </td> define el comienzo y el final del contenido de las celdas.

La etiqueta <th> </th> se utiliza para las celdas que pertenecen a la fila de encabezado.

<table border="1" width="300">

<tr>

<td><div align="center"><strong>Columna 1</strong></div></td>

<td><div align="center"><strong>Columna 2</strong></div></td>

</tr>

<tr>

<td>Celda 1</td>

<td>Celda 2</td>

</tr>

<tr>

<td>Celda 3</td>

<td>Celda 4</td>

</tr>

</table>



La diferencia con el ejemplo anterior es que para la fila de encabezado en lugar de utilizar la etiqueta <th> </th>, utiliza la etiqueta <td> </td>, la etiqueta <div> </div> con el atributo align="center", para que la alineación sea centrada; y la etiqueta <strong> </strong> para que el formato de los caracteres sea en negrita. Con estas etiquetas anidadas se logra la misma tabla que con el código anterior.

## 3.e)

<table width="200">

<caption>

Título

</caption>

<tr>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

<tr>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">

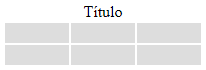
&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

</table>

Se visualiza:



<caption>..</caption>: esta etiqueta se coloca al comienzo de la tabla, justo antes del encabezado. Es la etiqueta que contiene el título de la tabla.

bgcolor="#dddddd" se utiliza para darle color de fondo a la celda.

&nbsp; sirve para representar un espacio en blanco.

<table width="200">

<tr>

<td colspan="3"><div align="center">Título</div></td>

</tr>

<tr>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

<tr>

<td bgcolor="#dddddd">&nbsp;</td>

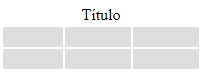
<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

</table>

Se visualiza:



En este caso se visualiza lo mismo que en el segmento de código anterior. Sin embargo, para insertar el título de la tabla utiliza las etiquetas <tr>..</tr>, <td>..</td>, <div>..</div> y los atributos colspan=”3” que sirve para unir celdas, en este caso une tres celdas horizontales; y el atributo align=”center” para que el título quede centrado.

## 3.f)

<table width="200">

<tr>

<td colspan="3"><div align="center">Título</div></td>

</tr>

<tr>

<td rowspan="2" bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

<tr>

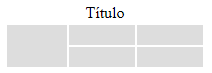
<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

</table>

Se visualiza:



A diferencia del ejercicio anterior, en lugar de utilizar el atributo colspan, se utiliza el atributo rowspan=”2” dentro de la etiqueta <td>..</td>. Sirve para combinar filas, en este caso, dos filas se combinan en una.

<table width="200">

<tr>

<td colspan="3"><div align="center">Título</div></td>

</tr>

<tr>

<td colspan="2" bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

<tr>

<td bgcolor="#dddddd">&nbsp;</td>

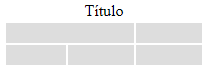
<td bgcolor="#dddddd">&nbsp;</td>

<td bgcolor="#dddddd">&nbsp;</td>

</tr>

</table>

Se visualiza la siguiente tabla



En este caso se combinan dos celdas horizontalmente, con el atributo colspan=”2”.

## 

## 3.g)

<table width="200" border="1">

<tr>

<td colspan="3"><div align="center">Título</div></td>

</tr>

<tr>

<td colspan="2"rowspan="2">&nbsp;</td>

<td>&nbsp;</td>

</tr>

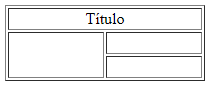
<tr>

<td width="50%">&nbsp;</td>

</tr>

</table>

Se visualiza



En este caso, para la segunda fila de la tabla se definen dos celdas

<table width="200" border="1" cellpadding="0" cellspacing="0">

<tr>

<td colspan="2"><div

align="center">Título</div></td>

</tr>

<tr>

<td rowspan="2">&nbsp;</td>

<td>&nbsp;</td>

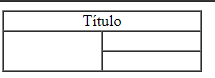
</tr>

<tr>

<td width="50%">&nbsp;</td>

</tr>

</table>



A diferencia del ejemplo anterior, se agregan los atributos:

Cellpadding: permite especificar la distancia entre el margen interior de la celda y su contenido. Cellspacing: permite especificar la distancia entre las celdas y el margen exterior de la tabla.

Entonces en este caso al ser el cellspacing="0" no tiene bordes ya que la distancia entre la las celdas y el margen exterior de la tabla es 0

Y al ser cellpadding="0" en este caso lo podemos visualizar en la palabra “Título” ya que es la única celda que contiene algo escrito, y aquí podemos ver que disminuye la distancia entre el contenido y el margen interior de la celda (el único borde que vemos, ya que no tiene margen exterior).

## 3.h)

<form id="form1" name="form1" action="procesar.php" method="post" target="\_blank">

<fieldset>

<legend>LOGIN</legend>

Usuario: <input type="text" id="usu1" name="usu1" value="xxx" /><br />

Clave: <input type="password" id="clave1" name="clave1" value="xxx" />

</fieldset>

<input type="submit" id="boton1" name="boton1" value="Enviar" />

</form>

Se visualiza:



<form> representa una sección de un documento que contiene controles interactivos que permiten a un usuario enviar información a un servidor web.

<fieldset> </fieldset> permite organizar en grupos los campos de un formulario. En este caso usuario y clave.

<legend> </legend> crea un título para el grupo de campos.

<input> se usa para crear controles interactivos para formularios basados en la web con el fin de recibir datos del usuario. Con el atributo type se indica si se va a ingresar texto, contraseña, si es un boton, etc. Con el atributo value se le da el valor inicial.

En este caso el <input> del botón enviar se encuentra fuera del <fieldset> </fieldset>, por lo que se puede ver que el botón queda fuera del recuadro.

<form id="form2" name="form2" action="" method="get" target="\_blank">

LOGIN<br />

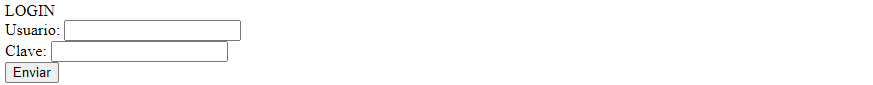
<label>Usuario: <input type="text" id="usu2" name="usu2" /></label><br />

<label>Clave: <input type="text" id="clave2" name="clave2" /></label><br />

<input type="submit" id="boton2" name="boton2" value="Enviar" />

</form>

Se visualiza:



En este caso no se utiliza la etiqueta <fieldset> </fieldset> , por lo que no se genera el recuadro para los campos.

Se utiliza la etiqueta <label> para los campos usuario y contraseña.

Además, en los inputs de usuario y contraseña no se especifica el atributo value, por lo que no hay ningún valor inicial.

<form id="form3" name="form3" action="mailto:xx@xx.com” enctype=text/plain method="p

ost" target="\_blank">

<fieldset>

<legend>LOGIN</legend>

Usuario: <input type="text" id="usu3" name="usu3" /><br />

Clave: <input type="password" id="clave3" name="clave3" />

</fieldset>

<input type="reset" id="boton3" name="boton3" value=

"Enviar" />

</form>

Se visualiza:



Es similar al ejemplo 1, aunque los input para usuario y clave, no contienen value, por lo que no contienen el valor inicial. Además, en la etiqueta <form> el atributo action contiene el correo "<mailto:xx@xx.com>”, que sirve para que el formulario se envíe a dicha dirección. En cambio en el ejemplo 1, ese atributo asume el valor: “procesar.php”; de esta manera, Cuando el usuario rellena este formulario y oprima el botón de envío, se llama a la página procesar.php.

Y el atributo enctype=text/plain especifica cómo se deben codificar los datos del formulario al enviarlos al servidor.

## 3.i)

<label>Botón 1

<button type="button" name="boton1" id="boton1">

<img src="logo.jpg" alt="Botón con imagen " width="30" height="20" /><br />

<b>CLICK AQUÍ</b></button></label>

Se visualiza:



Se utiliza la etiqueta <button>..></button>, la cual contiene una imagen de un logo y además un texto: “CLICK AQUÍ”. Todo esto se encuentra dentro de la etiqueta <label>...</label>

<label>Botón 2

<input type="button" name="boton2" id="boton2" value="CLICK AQUÍ" />

</label>

Se visualiza:



A diferencia del ejemplo anterior, aqui se utiliza un input de tipo button, al cual le da un value: “CLICK AQUÍ"

## 3.j)

<p><label><input type="radio" name="opcion" id="X" value="X" />X</label><br />

<label><input type="radio" name="opcion" id="Y" value="Y" />Y</label></p>

<p><label><input type="radio" name="opcion1" id="X" value="X" />X</label><br />

<label><input type="radio" name="opcion2" id="Y" value="Y" />Y</label></p>

En ambos lo primero que se visualiza es lo siguiente:



La diferencia es que en el primer ejemplo solo se puede elegir una opción, modificando las veces que se desee, mientras que en el segundo ejemplo se pueden elegir ambas opciones, pero una vez que se realizó la elección no se puede deshacer. Esta diferencia se debe a que para que suceda lo primero se le debe poner el mismo “name” a ambos inputs.

## 3.k)

<select name="lista">

<optgroup label="Caso 1">

<option>Mayo</option>

<option>Junio</option>

</optgroup>

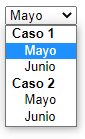
<optgroup label="Caso 2">

<option>Mayo</option>

<option>Junio</option>

</optgroup>

</select>



<select name="lista[]" multiple="multiple">

<optgroup label=" Caso 1">

<option>Mayo</option>

<option>Junio</option>

</optgroup>

<optgroup label=" Caso 2">

<option>Mayo</option>

<option>Junio</option>

</optgroup>

</select>



La diferencia entre ambos segmentos de códigos se puede ver en los atributos del select, en el segundo ejemplo agrega el atributo multiple, el cual permite elegir mas de una opción, a diferencia del ejemplo 1 que solo se puede elegir una.