



Arquitectura de Entornos Web

Curso	Arquitectura de Entornos Web (1800)
Formato	Manual de curso
Autor Institucional	Cibertec
Páginas	214 p.
Elaborador	López Aragón, Dámaso
Revisor de Contenidos	Morales Flores, Gustavo

Índice

Presentación	6
Red de contenidos	7

UNIDAD DE APRENDIZAJE 1: PROYECTO WEB

1.1 Tema 1 :	Fundamentos de un proyecto web	9
1.1.1 :	Internet	9
1.1.2 :	Página web	11
1.1.3 :	Hipertexto	12
1.1.4 :	Servidor web	12
1.1.5 :	Hosting	14
1.1.6 :	HTML	18
1.1.7 :	Navegadores de Internet	19
1.1.8 :	Proyecto web	21
1.1.9 :	Etapas de un proyecto web	21
1.1.10 :	Mapa de sitio	25
1.2 Tema 2 :	Elementos gráficos para la web	36
1.2.1 :	Wireframe	36
1.2.2 :	Beneficios de los wireframes	36
1.2.3 :	Tipos de wireframe: baja y alta fidelidad	37
1.2.4 :	Programas online para crear wireframes de baja fidelidad	38

UNIDAD DE APRENDIZAJE 2: HTML 5 Y CSS3

2.1 Tema 3 :	Estructura básica HTML	50
2.1.1 :	Definición	50
2.1.2 :	Características	50
2.1.3 :	Estructura de una página HTML	50
2.1.3.1 :	Etiquetas: html, head, meta, title, body	51
2.1.4 :	Etiquetas básicas	54
2.1.4.1 :	Encabezados <h1>, <h2>, ..., <h6>	54
2.1.4.2 :	Párrafos <p>	55
2.1.4.3 :	Líneas horizontales <hr>	55
2.1.4.4 :	Saltos de línea
	56
2.1.4.5 :	Comentarios <! -- comentario -->	56
2.1.4.6 :	Etiquetas de lista: Listas ordenadas, desordenadas y de definición	57
2.1.4.7 :	Texto enfatizado 	59
2.1.4.8 :	Texto reforzado 	59
2.1.5. :	Etiquetas para hipervínculos <a> y atributo target	59
2.1.6. :	Estilos en línea: uso del atributo style	61
2.1.6.1 :	Color de fondo	62
2.1.6.2 :	Color de texto	62
2.1.6.3 :	Alineación del párrafo	62
2.1.6.4 :	Uso de fuente: tamaño, decoración, estilo	62
2.1.7. :	Etiqueta de imagen y atributos src , alt , width y height	62
2.1.8. :	Etiquetas para figuras <figure> y <figcaption>	63
2.1.9. :	Uso de anclas: Enlace a un lugar de la misma página (id)	63

2.2 Tema 4	: Estilos CSS	81
2.2.1.	: Selector de estilo CSS	81
2.2.1.1	: Tipos de estilos	83
2.2.2	: Selectores CSS	86
2.2.2.1.	: Selector universal	86
2.2.2.2.	: Selector de etiqueta	86
2.2.2.3.	: Selector descendente	87
2.2.2.4.	: Selector de clase	87
2.2.2.5.	: Selector de id	88
2.2.2.6.	: Combinación de selectores	89

UNIDAD DE APRENDIZAJE 3: ESTRUCTURAS Y BLOQUES

3.1 Tema 5	: Modelo de cajas y posicionamiento CSS	106
3.1.1.	: Modelo de cajas	106
3.1.2.	: Posicionamiento de cajas: estático, relativo, absoluto, flotante; visualización: display, visibility, overflow, z-index	109
3.1.3.	: Etiquetas estructurales: header, nav, aside, main, section, article, footer	113
3.1.4	: Etiquetas flexibles: box-sizing, flexible box layout, grid	114

3.2 Tema 6	: Creación de menú	136
3.2.1	: Diseño de menú	136
3.2.2	: Creando un menú utilizando estilo a los elementos	137

UNIDAD DE APRENDIZAJE 4: MULTIMEDIA, TRANSICIONES Y ANIMACIONES

4.1 Tema 7	: Audio, video e iframe	151
4.1.1.	: Etiqueta de audio: propiedades y atributos	151
4.1.2.	: Etiqueta de video: propiedades y atributos	153
4.1.3.	: Etiqueta Iframe, reproducir videos de Youtube	154
4.2 Tema 8	: Animaciones, transiciones y transformaciones	172
4.2.1.	: Animaciones, propiedades	172
4.2.2.	: Transiciones, propiedades	173
4.2.3.	: Transformaciones, propiedades	179

UNIDAD DE APRENDIZAJE 5: FORMULARIOS

5.1 Tema 9	: Formularios	196
5.1.1	: Elementos de un formulario : form, input, button, textarea	196
5.1.2	: Nuevos atributos para la etiqueta INPUT : number, range, url, email, date, etc.	200
5.1.3	: Atributos del formulario	202
5.1.3.1	: Vistosidad de las entradas: placeholder, autofocus, etc	202
5.1.3.2	: Validación de las entradas: required, pattern, etc	202
Bibliografía		214

Presentación

La web 3.0 es la nueva evolución que han experimentado todos los navegadores modernos, los cuales, permiten marcar el contenido de forma semántica, para así poder ser indexado con mucha más fiabilidad y efectividad.

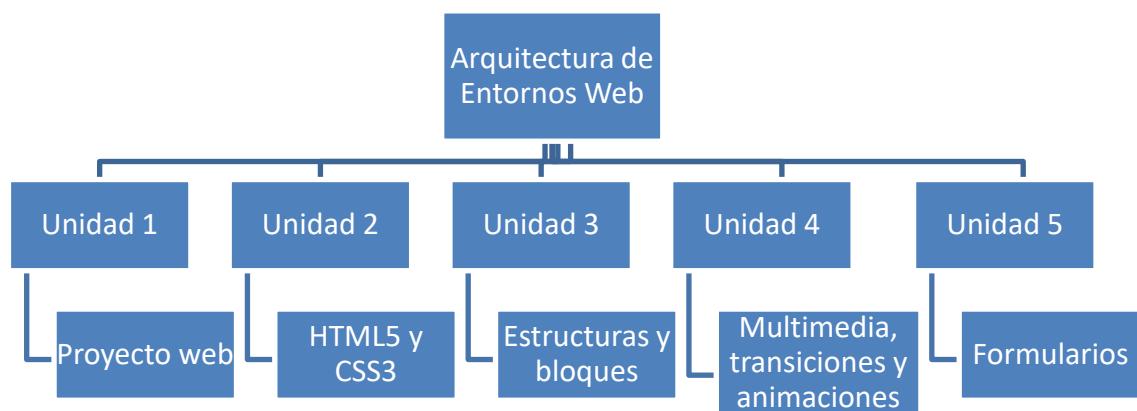
HTML5 es un nuevo lenguaje para el diseño de páginas web, que proporciona muchas más etiquetas de marcado de contenido entre los que destaca los bloques de contenidos, efectos multimedia y formularios.

Arquitectura de entornos Web pertenece a la línea de tecnología y se dicta en las carreras de tecnología de la institución. El curso brinda un conjunto de herramientas de diseño y programación para la creación de páginas web.

El manual para este curso ha sido diseñado bajo la modalidad de unidades de aprendizaje, las que se desarrolla durante semanas determinadas. En cada una de ellas, se hallará los logros que se deberá alcanzar al final de la unidad; el tema tratado, el cual será ampliamente desarrollado; y los contenidos, que debe desarrollar. Por último, encontrarás las actividades y trabajos prácticos que se desarrollará en cada sesión, los cuales permitirán reforzar lo aprendido en la clase.

El curso es eminentemente práctico: consiste en el diseño de sitios web y programación de páginas web utilizando HTML5 y formularios. La primera parte de este manual se explica acerca de los conceptos básicos de la web y los fundamentos de un proyecto web. Luego, se aprenderá a diseñar y construir páginas web para un sitio web aplicando las etiquetas HTML5, hojas de estilos CSS3, elementos de multimedia, animaciones y formularios. Por último, se aprenderá a publicar un proyecto web en un sitio web.

Red de contenidos





PROYECTO WEB

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno define qué es un proyecto web y elabora wireframes de alta y baja fidelidad.

TEMARIO

1.1 Tema 1 : Fundamentos de un proyecto web

- 1.1.1 : Internet
- 1.1.2 : Página web
- 1.1.3 : Hipertexto
- 1.1.4 : Servidor web
- 1.1.5 : Hosting
- 1.1.6 : HTML
- 1.1.7 : Navegadores de Internet
- 1.1.8 : Proyecto web
- 1.1.9 : Etapas de un proyecto web
- 1.1.10 : Mapa de sitio

1.2 Tema 2 : Elementos gráficos para la web

- 1.2.1 : Wireframe
- 1.2.2 : Beneficios de los wireframes
- 1.2.3 : Tipos de wireframe: baja y alta fidelidad
- 1.2.4 : Programas online para crear wireframes de baja fidelidad

ACTIVIDADES PROPUESTAS

- Los alumnos desarrollan los laboratorios de esta semana.
- Los alumnos desarrollan diseñan el mapa de sitio de su proyecto.
- Los alumnos desarrollan diseñan su wireframe de baja y alta fidelidad.

1.1. FUNDAMENTOS DE UN PROYECTO WEB

En la era de la globalización, la Internet se ha convertido en una herramienta indispensable para cualquier persona o profesional. Este medio se ha constituido como una de las principales vías de consulta de información, publicación y actualización de conocimientos, así como de comunicación entre profesionales.

En esta era, las empresas, organismos estatales, publicaciones, programas de televisión o individuos no existen sino tienen su dirección de correo electrónico o su sitio web; la triple W (www) y la @ (arroba) ya forman parte de la vida, y se están haciendo cada vez más imprescindibles.

Internet y Web son dos términos que a menudo se utilizan indistintamente de forma que se puede caer en la confusión y mezclar conceptos. Internet es un conjunto de redes de comunicación interconectadas entre sí, siguiendo el protocolo TCP/IP, que funcionan como una red única mundial y siendo la web un servicio más de Internet.

A continuación, se expone los conceptos principales acerca de las generalidades de la web.

1.1.1 Internet

Internet se define como una red mundial de computadoras, distribuidas por todo el mundo, las cuales se encuentran interconectadas a través de satélites, líneas telefónicas, fibra óptica, etc., donde utilizando un protocolo de comunicaciones común (TCP/IP), permite a millones de usuarios estar conectados para compartir, intercambiar, publicar y extraer información.

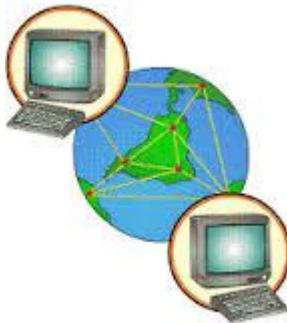


Figura 1: Internet
Fuente. – Tomado de <http://definicion.de/internet/>

En la actualidad es un error considerar a Internet solamente como una red de computadoras. En este caso Internet es una gran fuente de información, donde se puede enviar mensajes, descargar programas, archivos y ficheros de texto; consultar catálogos de bibliotecas, solicitar libros, realizar compras, etc.

Hay que tener en cuenta que todos los recursos que se pueden encontrar en Internet existen porque muchas personas, de forma voluntaria, han dedicado su tiempo en generarlos.

Protocolo de Internet

El protocolo de Internet es un conjunto de reglas formales que permite la transmisión de los datos entre dos computadoras. El ordenador conectado a una red usa protocolos para que envíe y reciba mensajes.



Figura 2: Protocolos de Internet

Fuente. - Tomado de http://www.hipertexto.info/documentos/internet_tecn.htm

El protocolo más utilizado es el TCP/IP (Protocolo de Control de Transmisión TCP y Protocolo de Internet IP). Este protocolo especifica cómo las computadoras de puerta encaminan la información desde la computadora emisor hasta el computador receptor.



Figura 3: TCP / IP

Fuente. – Tomado de <http://www.taringa.net/posts/info/12588433/Protocolos-de-Internet.html>

Actualmente existen varios protocolos de Internet, se explica los más utilizados:

IPv6, será la próxima generación de protocolos de Internet, este protocolo se ha desarrollado para ampliar la capacidad de conexión debido al crecimiento de dispositivos y el aumento de equipos portátiles.

FTP, es un protocolo de transferencia de archivos de un ordenador a otro, de un servidor remoto a uno local y viceversa. Se precisa de un servidor de FTP y un cliente de FTP, estos pueden ser de acceso con una cuenta (login) o anónimo, la cual accede sin ninguna contraseña.

HTTP, protocolo de transferencia de Hipertextos, es el protocolo más utilizado por los servidores de la World Wide Web, el cual permite el intercambio de información hipertextuales (enlaces) de las páginas web.

SMTP, protocolo de transmisión de correo simple, es el protocolo que permite recibir correos electrónicos y, junto con el protocolo POP (protocolo de oficinas de correo) es usado por los ordenadores personales para administrar el correo electrónico, permitirá descargar los mensajes a la computadora.

TELNET, es un protocolo que permite la conexión remota a otro ordenador, el cual permite manejarla como si se estuviera físicamente frente al ordenador.

Otros conceptos:

URLs, localizador uniforme de recursos, representa la dirección electrónica para poder acceder a los recursos del servidor remoto. El tipo más común es el de las páginas web, con la dirección `http://`; también existen otras direcciones como `ftp://` para ubicar un recurso FTP con la finalidad de transferir archivos.

1.1.2 Página Web

Una página web es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes, hipervínculos y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web.

En Internet existen más de mil millones de páginas web de diversas índoles y contenido variado, provenientes del mundo entero y en los principales idiomas hablados. Esto representa el principal archivo de información de la humanidad que existe actualmente, almacenado a lo largo de miles de servidores a lo largo del planeta, a los que son posible acceder velozmente gracias a un sistema de protocolos de comunicación (HTTP).

Las páginas Web se encuentran programadas en un formato HTML o XHTML, y se caracterizan por su relación entre unas y otras a través de hipervínculos: enlaces hacia contenidos diversos que permiten una lectura compleja, simultánea y diversa, muy distinta a la que se halla en los libros y revistas.

Las páginas Web cumplen básicamente con la tarea de brindar información de cualquier índole y en cualquier estilo o grado de formalidad.

Algunas, al mismo tiempo, permiten distintos grados de interacción entre usuarios o con alguna institución, como son las páginas de foros, servicios de citas o redes sociales, las páginas de compra y venta de bienes, las páginas de consulta o de contacto con empresas, instituciones gubernamentales o con ONGs, e incluso las páginas de soporte técnico especializado.

En principio, las funciones de una página Web son tan amplias como la demanda de los usuarios y la oferta de sus creadores.

Tipos de página Web

Existen dos tipos de página Web, conforme al modo en que se genera su contenido:

Páginas Web estáticas. Operan mediante la descarga de un fichero programado en código HTML, en el que están todas las instrucciones para que el navegador reconstruya la página Web, accediendo a las ubicaciones de sus elementos y siguiendo un orden preconcebido, rígido, que no permite la interacción con el usuario. Este tipo de páginas son meramente informativas, documentales, no interactivas.

Páginas Web dinámicas. A diferencia de las anteriores, las páginas Web dinámicas se generan en el momento mismo del acceso del usuario, empleando para ello algún lenguaje interpretado (como el PHP), lo cual le permite recibir solicitudes del usuario, procesarlas en bases de datos y ofrecer una respuesta acorde a sus requerimientos.

1.1.3 Hipertexto

El hipertexto es una estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos y redes sociales.

El término **hipertexto** se ha utilizado con diferentes acepciones. La primera de ellas, con un carácter general, para hacer referencia a una forma de organizar la información. La segunda, como un término concreto usado en dos sentidos, o bien para referirse a las herramientas software que permiten crear y gestionar este tipo de organización conceptual, o bien para hacer referencia a los documentos creados por estas herramientas software. En última instancia, y tal como indica el W3C, **hipertexto** “es un texto que no está limitado a ser lineal”.

Aunque el término hipertexto fue acuñado por Ted Nelson en 1965 (Connolly 2000) para hacer referencia a una estructura del texto no secuencial que permitiera al lector elegir aquello que considerara más adecuado en una pantalla interactiva, lo que realmente se asocia en la actualidad con hipertexto es el modelo creado por Tim Berners-Lee con el objetivo de compartir documentos en internet, desarrollado en colaboración con Robert Cailliau en 1989, al que se le dio el nombre de HTML Tags (HyperText Markup Language), lo cual dio origen a lo que hoy conocemos como la Web (Coremans 2015).

Este lenguaje de etiquetas, HTML, fue una simplificación de un metalenguaje, o marco general para la descripción de lenguajes de marcado, denominado SGML (Standard Generalized Markup Language), el cual, por su extrema complejidad, no era utilizable para el entorno Web (Lamarca Lapuente 2013c).

Desde que vio la luz el primer documento en HTML en 1991 hasta la actualidad, se han llevado a cabo sucesivas versiones de este estándar definido por el W3C como la “lengua franca” para la publicación de hipertexto en el World Wide Web. En el año 1995 se hizo cargo de su desarrollo el World Wide Web Consortium en la versión 3.0 de este lenguaje de marcas. En la actualidad, a partir del año 2014, se encuentra disponible la versión 5.0 en la categoría de estándar, el cual es ya soportado por la mayoría de clientes (navegadores web) que existen en el mercado.

1.1.4 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

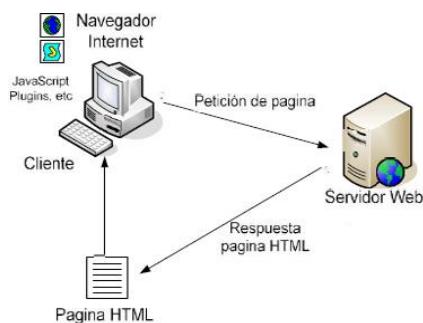


Figura 4: Servidor Web

Fuente. – Tomado de <http://trimestrecuartodeamarelysmaita.blogspot.com/2013/06/>

El funcionamiento de un servidor web básico basado en ejecutar infinitamente el siguiente bucle:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

Un servidor web que siga el esquema anterior, cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir ficheros estáticos.

En este punto es necesario aclarar lo siguiente: comúnmente se utiliza la palabra **servidor** para referirse a una computadora con un software servidor instalado, el cual se encarga de “alojar” los sitios web que se visita en la Internet y son estas computadoras las que proporcionan al usuario la interacción con el sitio en cuestión.

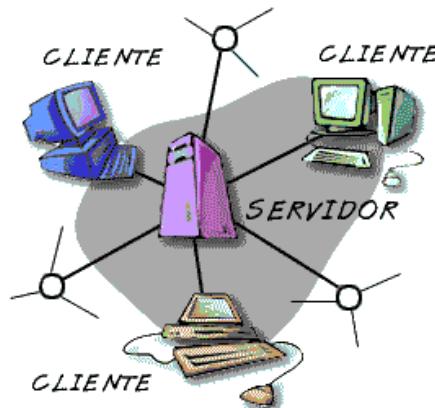


Figura 5: Servidor HTTP

Fuente. - Tomado de <https://vikastheengineer.wordpress.com/2012/10/03/types-of-computer-networks/>

Porqué son necesarios los servidores

Los servidores son como la columna vertebral de la estructura de Internet. La industria del Web hosting es simplemente la forma de alquilar esos espacios de memoria y administración de datos.

Sin este sistema de leasing, la Internet sería sólo una pequeña parte de lo que es hoy por hoy, ya que sería imposible para la mayoría de los dueños de sitios Web comprar un servidor para uso propio.

Tipos de servidores

Servidor Dedicado: se trata de una computadora dedicada exclusivamente al sitio del cliente, para aplicaciones de alta demanda. Entre las características se tiene:

- Son de pago superior a fin de tener un servidor para sí.
- Mayor capacidad en el disco duro, alta tasa de transferencia.

- Son más difícil de manejar

Servidor Conjuntamente: es un tipo de servidor similar al dedicado. La diferencia es que tú eres dueño del servidor, es decir, lo compras en lugar de alquilarlo de una empresa de servicio hosting. Entre las características se tiene:

- No son gratuitos, se paga una tarifa por el ancho de banda y tasas de mantenimiento.
- Se beneficia por contar con una alta tasa de velocidad.
- El servidor se encuentra ubicado físicamente en un centro de datos.

Servidor Compartido: se trata de un ordenador que aloja más de un sitio web. Entre las características se tiene:

- La mayoría de los servidores de este tipo son gratuitos.
- Todos los sitios web deben compartir recursos de una misma computadora, de manera desigual.
- Se controlan fácilmente.
- Uno de los inconvenientes es que algunos sitios web requieren de más recursos por ser de más rendimientos que las otras.

Servidor Virtual Privado (VPS): se trata de un ordenador que divide todos sus recursos entre los sitios web que aloja. Entre las características se tiene:

- No son gratuitos y su precio no es muy caro.
- Su manejo es sencillo, está casi todo hecho.
- Si el procesador se sobrecarga, fallarán los sitios web.
- Se comparte el HDD y la RAM pero no el procesador

Tipos de servidores Web

Apache: es el más común y utilizado en todo el mundo. Además, es gratuito y de código abierto, lo que implica que corre en cualquier plataforma.

Microsoft IIS: solamente funciona sobre la plataforma Windows. Para que trabaje sobre otro sistema tendrás que utilizar una máquina virtual.

Nginx: es un servidor web ligero y corre sobre un sistema Unix y Windows

Lighttp: está especialmente pensado para hacer cargas pesadas sin perder balance, utilizando poca memoria RAM y poca de CPU.

Sun Java System Web Server: es multiplataforma y se distribuye con licencias de código abierto.

1.1.5 Hosting

El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

Esta palabra es una analogía de “hospedaje o alojamiento en hoteles o habitaciones” donde uno ocupa un lugar específico, en este caso la analogía alojamiento de páginas web, se refiere al lugar que ocupa un sitio web, sistema, correo electrónico, archivos etc. en un servidor que por lo general hospeda varias aplicaciones o páginas web.



Figura 6: Hosting

Fuente. - Tomado de <http://chiapashosting.com/hosting-o-alojamiento-web.html>

Principales ventajas de un Hosting

- Tener un servicio de hosting disminuye los costos a los usuarios, puesto que un grupo de usuarios (los cuales acceden al mismo servicio) se encargará de compartir los gastos del servidor, mantenimiento y sus actualizaciones
- Algunos de los servidores de web hosting no cuentan con un administrador determinado, hecho que también se encarga de disminuir los costos de manera significativa; también existen otros servidores de web hosting que cuentan con equipos de profesionales y con personal de servicio al cliente, especialmente dedicados a responder las necesidades más específicas de los clientes.
- Los servicios de web hosting son indispensables para todo tipo de empresas que surgen en la web, no importa si son servicios de web hosting gratuitos, dedicados o compartidos, lo importante es acceder al hosting adecuado para las necesidades.

Qué se debe tener en cuenta antes de contratar un servicio hosting

Como en la mayoría de los distintos servicios, todo depende de lo que precises. Hay servicios de hosting muy económicos, que no poseen ningún tipo de garantía sobre tus datos, ni te aseguran una estabilidad mínima de tu servicio; y también hay otros servicios que incluyen seguramente aplicaciones y servicios que no se necesita.

Por supuesto, los servidores de pago prestan un mejor servicio, más personalizado y ofrecen soporte técnico, paneles de control amigables, así como plantillas y demás comodidades completamente instaladas en el servicio de alojamiento.

G HOSTINGROUP HOSTERIA EN COMPUTACIONES		Hosting	Soluciones	Servidores	Nosotros
C* 01748-5114					
 Plan Emprendedor <p>Hosting Peruano ideal para nuevos negocios.</p> <p>S/. 150.00 /año</p> <p>Precio no incluye IGV.</p> <p> Contratar este Plan »</p> <hr/> <p>2 GB Almacenamiento</p> <p>60 GB Transferencia Mensual</p> <p>60 Cuentas de Correo</p> <p>60 Cuentas FTP & MySQL</p> <p>99.999% Tiempo de Disponibilidad</p> <p>24/7/365 Soporte Técnico</p> <p>1 Dominio .COM Gratis</p> <p>No Alojar múltiples dominios</p> <p>Si Certificado de Seguridad SSL</p> <p>Copias de Seguridad Diarias</p> <p>Antivirus & Antisam</p> <p> WhatsApp Clic para Chatear</p>	 Mi Venta	 Plan PYME <p>Perfecto para pequeñas y medianas empresas</p> <p>S/. 200.00 /año</p> <p>Precio no incluye IGV.</p> <p> Contratar este Plan »</p> <hr/> <p>3 GB Almacenamiento</p> <p>100 GB Transferencia Mensual</p> <p>Ilimitadas Cuentas de Correo</p> <p>100 Cuentas FTP & MySQL</p> <p>99.999% Tiempo de Disponibilidad</p> <p>24/7/365 Soporte Técnico</p> <p>1 Dominio .COM Gratis</p> <p>Si Alojar múltiples dominios</p> <p>Si Certificado de Seguridad SSL</p> <p>Copias de Seguridad Diarias</p> <p>Antivirus & Antispam</p>	 Plan Empresarial <p>Máxima potencia para grandes empresas</p> <p>S/. 300.00 /año</p> <p>Precio no incluye IGV.</p> <p> Contratar este Plan »</p> <hr/> <p>5 GB Almacenamiento</p> <p>200 GB Transferencia Mensual</p> <p>Ilimitadas Cuentas de Correo</p> <p>Ilimitadas Cuentas FTP & MySQL</p> <p>99.999% Tiempo de Disponibilidad</p> <p>24/7/365 Soporte Técnico</p> <p>1 Dominio .COM Gratis</p> <p>Si Alojar múltiples dominios</p> <p>Si Certificado de Seguridad SSL</p> <p>Copias de Seguridad Diarias</p> <p>Antivirus & Antispam</p>		

Figura 7: Servicios de Hosting

Fuente. - Tomado de <https://www.hostinggroup.com/peru/hosting-web>

Para contratar un servicio hay que tener en cuenta una gran cantidad de factores:

- La cantidad de espacio en GB que ofrece el servicio para alojar las páginas web
 - La capacidad de gestionar los archivos vía FTP
 - Soporte para scripts perl y python, php, ASP, etc
 - Tipo de dirección web que ofrecen
 - Tasa de transferencia de archivos mensuales

Dominio

Un dominio de Internet es un nombre de un sitio web. Cada dominio tiene que ser único, por ejemplo “elcomercio.pe” es el nombre del dominio de la página web del diario El Comercio.



Figura 8: Dominio

Fuente. - Tomado de <http://blog.ozongo.com/cuantas-extensiones-necesito-para-el-dominio-de-mi-empresa/>

Todos los servidores y páginas de Internet tienen una dirección numérica que se conoce como dirección IP (Protocolo de Internet), por ejemplo 50.17.245.112

Los dominios fueron creados para evitar la necesidad de recordar las direcciones numéricicas de las páginas y servidores web. De forma que cuando se escribe en internet el dominio elcomercio.pe, el servidor de DNS (Servidor de Nombres de Dominio) proveedor de web hosting del dominio elcomercio.pe, proporciona la dirección IP 50.17.245.112 y el navegador se dirige directamente a esa dirección numérica.

Elementos de un dominio

Un dominio de Internet está compuesto por 2 elementos:

- Nombre del dominio
- Extensión o terminación del dominio

El nombre hace referencia a la marca o una palabra resumen que indique el tipo de actividad o información que brinda la web, por ejemplo www.pepsi.com.ar. Esta dirección hace referencia al nombre de una marca mundial de refrescos.

La terminación o extensión de los dominios, indica la ubicación geográfica donde se ubica el sitio web así como el tipo de organización o fin que tiene. Es aquí donde se distingue dos tipos de extensiones:



Figura 9: Dominio

Fuente. - Tomado de <http://www.quees.info/que-es-un-dominio.html>

Territoriales: indican a qué país o región pertenece la página, actualmente existen un total de 243 países y está compuesto por dos letras que corresponde a la abreviación del nombre del país:

- .es, España
- .pe, Perú
- .us, Estados Unidos

Internacionales: hacen referencia al tipo de actividad que ofrece el sitio web. A diferencia de los territoriales, éstos están compuestos por más de dos letras.

Dominio	Descripción
.biz	Prevista para ser usado por negocios
.com	Son los dominios más extendidos en el mundo. Sirven para cualquier tipo de página web y temática
.edu	Servicios de educación
.gov y .gob	Gobierno y entidades públicas
.info	Información
.mil	Departamento de Defensa de los Estados Unidos

.name	Nombres de personas
.net	Infraestructura de red e Internet
.org	Organizaciones
.tel	Servicios de comunicación por internet
.travel	Páginas de la industria de viajes y turismo
.aero	Sector aeronáutico
.edu	Universidades, educación
.jobs	Empleo

1.1.6 HTML

El HTML (Hyper Text Markup Language) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web).

Los navegadores se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado.

Actualmente se utiliza el HTML 5, está formado por muchos módulos distintos, cuyo grado de especificación está en niveles dispares. Por tanto, muchas de las características de HTML 5 están ya listas para ser implementadas, en un punto de desarrollo que se encuentra cercano al que finalmente será presentado.

Las versiones recientes de casi todos los navegadores, incluido el polémico Internet Explorer 8, implementan algunas de las características de HTML 5. Claro que, para que una web se vea bien en todos los sistemas, hay que utilizar sólo aquellas partes que funcionan en todos los navegadores, por lo que, a día de hoy, pocas son las utilidades realmente disponibles del lenguaje, si queremos hacer un sitio web compatible.

Novedades de HTML 5

HTML 5 incluye novedades significativas en diversos ámbitos. No sólo se trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

- Estructura del cuerpo: la mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Etiquetas para contenido específico: hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- Canvas: es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin.
- Bases de datos locales: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo parecido a

las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectadas a Internet.

- Web Workers: son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página. Para ello, se dispondrá también de un API para el trabajo con los Web Workers.
- Aplicaciones web Offline: existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- Geolocalización: las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.
- Nuevas APIs para interfaz de usuario: temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.
- Fin de las etiquetas de presentación: todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

1.1.7 Navegadores de internet

Un navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web, la cual puede estar alojada en un servidor dentro de la World Wide Web o en uno local.

El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.



Figura 10: Navegadores

Fuente. - Tomado de <http://www.adrformacion.com/cursos/dreamcs3/leccion1/tutorial3.html>

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, reproducir recursos multimedia incrustados y ejecutar acciones. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web).

Navegadores más utilizados



Figura 11: Navegadores

Fuente. - Tomado de <https://miracomosehace.com/mejor-navegador-internet-2020/>

1- Chrome

Este navegador de internet es uno de los más populares y utilizados en la actualidad. Afortunadamente el mismo ofrece una serie de ventajas que lo convierten en uno de los mejores, destacando que cuenta con una excelente velocidad, su interfaz es sencilla y dispone de un motor de JavaScript sensacional.

Otro de los beneficios que brinda es su compatibilidad con todos los sistemas operativos y adaptación con los servicios de Google, pudiendo ser personalizado según lo requiera el usuario. Ofrece extensiones nativas que facilitan diversos procesos.

Por otro lado, entre sus desventajas se tiene su gran uso de memoria RAM, posee un gestor de contraseñas sencillo y una parte de su código es privado; frecuentemente realiza actualizaciones, lo cual puede ser un poco molesto para los usuarios de esta plataforma web.

2- Mozilla Firefox

Otro de los navegadores que no se queda atrás en cuanto a funcionalidad es Firefox, ofreciendo una gran variedad de extensiones de gran utilidad, así como una interfaz personalizable. Pero esto no es todo, porque además dispone de software libre y es compatible con Smartphone.

Otra de sus ventajas es que consume menos memoria que Google Chrome, así como también trae incluidos servicios gratuitos, como lo son capturador de pantalla, envío de archivos (Firefox Send), Pocket, entre otros.

Entre sus desventajas se encuentran, menos atención de los diseñadores web, por ende, algunas páginas pueden generar errores o cualquier detalle molesto.

Los usuarios solo tienen derecho a tener una cuenta asociada.

Sus actualizaciones son lentas.

Dispone de herramientas innecesarias

3- Microsoft Edge

Este navegador ha llegado a apoderarse y reemplazar a Internet Explorer, logrando ofrecer mejores características que este último. Destacando como principal ventaja que ha sido diseñado por la compañía de Microsoft, de este modo está integrado a Windows 10; de igual manera cuenta con un excelente rendimiento de JavaScript.

Entre sus desventajas: sus funciones son similares a las de Chromium, cuenta con compatibilidad para Windows y MacOS, es decir que no tiene relevancia para los demás sistemas operativos de la actualidad y, por último, la mayor parte de su código es propietario.

4- Opera

Este navegador de internet dispone de un modo turbo, que lo convierte en el mejor para conexiones lentas. Dispone de gran compatibilidad con Smartphone e incluye algunas funciones como VPN, bloqueador de publicidad, entre otras. No obstante, se conoce que su código es propietario, sus actualizaciones son muy pocas, entre otras.

5- Safari

Por último, safari sorprende con una interfaz limpia, clara y sencilla, excelente compatibilidad con Mac OS X y iOS, sin embargo, sus extensiones y herramientas son muy básicas en comparación de los demás, es compatible solo con Apple y su código es privado.

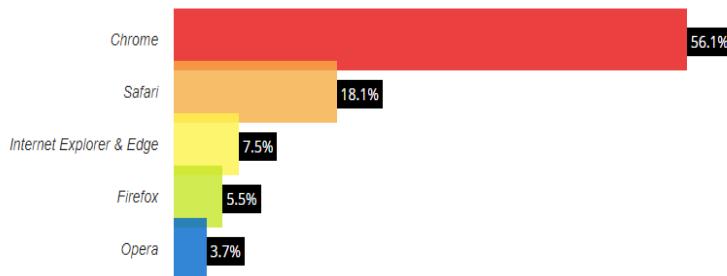


Figura 12: Estadísticas

Fuente de .- Tomado de <https://www.pcworld.es/mejores-productos/internet/mejores-navegadores-web-3672988>

1.1.8 Proyecto web

Debido a la transformación digital, las marcas necesitan estar cada vez más presentes full-time en la vida de los consumidores. Actualmente, quien trabaja con desarrollo de **sitios web** o aplicaciones móviles, han notado una demanda creciente. Y según se percibe en el mercado, la inversión digital no es más un extra, sino por el contrario, es primordial para el posicionamiento.

Un proyecto web es un emprendimiento temporal realizado para crear un producto o servicio único que, haciendo uso de una plataforma web, busca conseguir un objetivo específico.

El enfoque de proyecto web busca superar la visión de “diseñar una página web”. Se trata de concebir el desarrollo de una solución web, como un proyecto que debe ser formulado, gestionado y evaluado.

Para tener éxito en cualquier “proyecto web”, es fundamental dimensionar correctamente todos los pedidos, lo que contribuye a aumentar la credibilidad del profesional y tener clientes satisfechos.

1.1.9 Etapas de un proyecto web

Desarrollar un sitio web efectivo requiere una planeación bien elaborada. Puede ahorrar tiempo y dinero durante el proceso del proyecto si se logra entenderlo en su totalidad antes de comenzarlo.

A pesar de que los sitios varían en rango de contenido, desde unas cuantas páginas hasta sitios mucho más complejos con características sofisticadas, el proceso de desarrollo se mantiene igual.

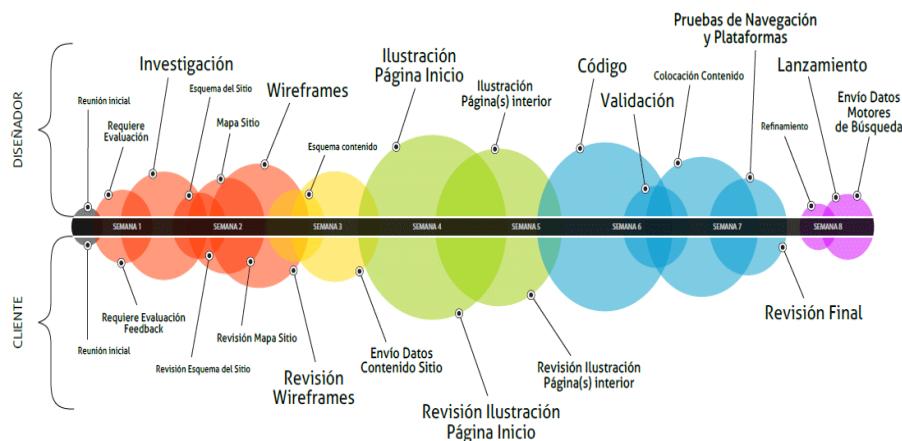


Figura 13: Etapas de un proyecto Web
Fuente. - Tomado de <http://www.n4gash.com/2014/infografia-pasos-diseno-pagina-web/>

A continuación, se presenta las etapas de un proyecto de diseño de sitios Web:

1. **Estrategia:** es la primera etapa a definir antes de tomar cualquier acción. Se necesita decidir cuál es el propósito del sitio web, cuáles son las necesidades del negocio que motivan el proyecto. Es importante definir:
 - Objetivos y metas.
 - Analizar y recoger la información
 - Lista de puntos para producción (tecnología, servidor web, presupuesto)
 - Designar un editor del sitio
2. **Benchmarking:** conocer qué está haciendo la competencia y como está posicionado su negocio en la Internet. Se analizan sitios web relacionados con el área del proyecto, determinando elementos y funcionalidades destacados que puedan ser utilizados para ofrecer un sitio web más competitivo.



Figura 14: Banchmanrkng
Fuente. - Tomado de https://www.enterurl.com/promotion_sub_profile.asp

3. **Arquitectura de la información:** con toda la información que se va a contar de las reuniones realizadas con los clientes, se crea un mapa del sitio (**SiteMap**) para el proyecto donde se muestra cómo será la estructura de secciones de la nueva web.

Este proceso ayuda a comprender qué y cómo se va a comunicar y establecer las funcionalidades que los usuarios necesitarán.

Los contenidos bien planificados y bien estructurados ayudarán a crear tanto el sistema de interacción como el diseño visual.

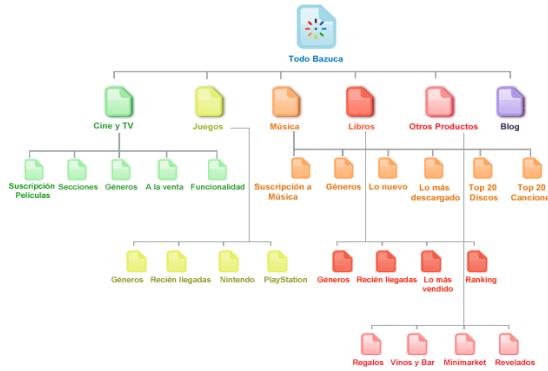


Figura 15: Arquitectura de Información

Fuente. - Tomado de <http://informaticayserviciosdeinternet.blogspot.com/2012/05/mapa-de-sitio.html>

4. **Diseño de prototipos:** a partir de la priorización de contenidos, se trabaja con el cliente a través de un proceso iterativo para diseñar un prototipado o boceto muy esquemático (en inglés, wireframe) de las principales páginas de la web.

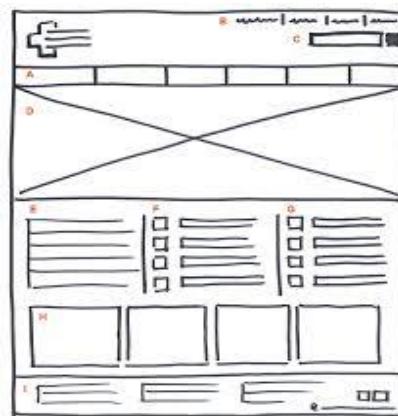


Figura 16: Diseño de prototipos

Fuente. - Tomado de <http://teamddm.com/articles/wireframes-getting-a-shared-understanding>

Esto ayudará al cliente a conocer cómo se organizará y será mostrado el contenido en dispositivos móviles, tablets y ordenadores de sobremesa.

5. **Construcción:** en este punto se crea la plantilla de la web con HTML5.



Figura 17: Construcción del sitio
Fuente. - Tomado de <http://laminaliagalicia.es/?p=1>

Este paso es a menudo un trabajo de colaboración estrecha entre diseñadores y desarrolladores. La codificación adaptativa empezará con el diseño web y se usa la lista de priorización de contenidos o el prototipado como mapa de ruta para priorizar los contenidos. Se incorpora los diferentes estilos de diseño según lo especificado en la guía de estilos mediante CSS3 y es importante asegurarse de que cumple con los estándares de W3C. Se usa JQuery y JavaScript para eventos interactivos, en lugar de Flash o Silverlight. También se aplica las mejores prácticas para facilitar la lectura online y la impresión de contenidos añadiendo el código específico en CSS.

6. **Pruebas de usabilidad y refinamiento:** las pruebas de usabilidad son evaluaciones que se realizan a sitios web con el objeto de determinar si son intuitivos y fáciles de navegar. Estas pruebas se realizan con usuarios reales y con usuarios expertos.



Figura 18: Pruebas
Fuente. - Tomado de <http://www.condo-consulting.com/test-de-usabilidad-que-son-y-para-que-me-sirven.b.aspx>

Realizar pruebas de usabilidad a la página web permitirá saber si la información está bien estructurada y si generará una experiencia positiva al usuario en cada visita. Al realizar esta prueba se evalúa la navegabilidad, arquitectura de la información, entradas y datos, formularios, credibilidad y confianza de su página web.

7. **Publicación en la Web:** este será el último paso, subir el proyecto a la Internet, para ello se debe tener en cuenta lo siguiente:
 - Contratar un hosting, la empresa dispone de distintas alternativas: alojamiento virtual, tener servidor propio, servidor dedicado.
 - Si se quiere generar tráfico hacia el sitio web, la empresa debe iniciar una labor de promoción del sitio.

- Para que el sitio sea visitado en forma continua, la empresa deberá realizar frecuentes actualizaciones y revisiones de sus páginas.



Figura 19: Publicación

Fuente. - Tomado de <http://www.callegranvia.com/empresariales/info/promover-su-negocio-con-la-publicacion-de-articulos/>

1.1.10. Mapa de sitio

Un Sitemap o Mapa del Sitio es la estructura de enlaces de páginas web que definirá la buena o mala navegación de un sitio web.

El mapa de un sitio web puede ser un documento que se utiliza para planificar el diseño de la web, o puede ser una página web donde se listan todas -o las más importantes- páginas web de un sitio (generalmente organizadas de alguna manera).

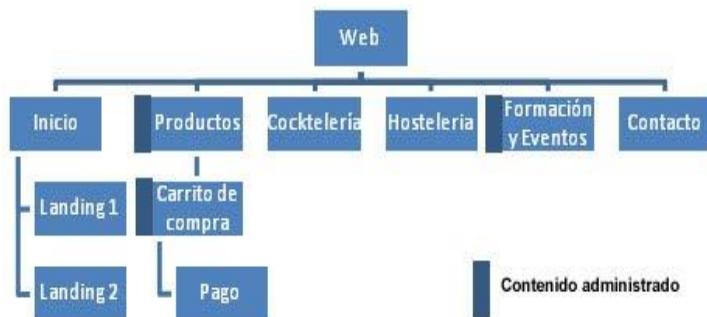


Figura 20: Mapa de Sitio

Fuente. - Tomado de <http://marketinges.es/tag/sitemap/>

Empezando por la página de Inicio hasta la página de contacto, todo sitio web debe tener un mapa para que la navegación del usuario web sea lo más fácil posible.



Figura 21: Mapa de Sitio

Fuente. - Tomado de <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacion-de-sitios-web>

Importancia

Un mapa de sitio es importante fundamentalmente porque ayuda a los buscadores a que encuentren tu web y a que indexen sus páginas más rápidamente.

Es como si a un buscador se le diera un índice del libro (libro es web) y le ahorras el esfuerzo de tener que buscar él mismo donde está cada cosa.

Características

- Consistentes: el sistema debe ser similar en todo el sitio, en lo referido a su ubicación y disposición en las páginas. Esto se aplica también a aquellas instituciones que pueden tener más de un sitio web.
- Uniforme: el sistema debe utilizar similares términos con el fin de que el usuario que lo vea en las páginas, confíe en que sus opciones llevan siempre hacia los mismos lugares dentro del sitio.
- Visible: debe ser una guía permanente en el área en que se encuentre el usuario dentro del sitio.

Ventajas

- Resumen de la información y accesibilidad en un enlace directo
- Facilidad de navegación
- El usuario permanece más en la Web
- La navegación resulta una experiencia agradable
- Una estructura clara y sencilla

Técnica para construir un sitio web

Una de las técnicas más populares utilizadas por los arquitectos de información es la conocida card sorting (tarjetas ordenadas).



Figura 22: Técnica para construir un sitio web

Fuente. - Tomado de <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacion-de-sitios-web>

Esta técnica se basa en la observación de cómo los usuarios agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con las diferentes categorías del sitio web.

De esta forma, partiendo del comportamiento de los propios usuarios, es posible organizar y clasificar la información de un sitio web conforme a su modelo mental.

LABORATORIO 1

Creando un mapa de sitio con Creately

Diseña un mapa de sitio utilizando el aplicativo Creately.

Acceso a la plataforma

Ingresa a la plataforma, escribe la siguiente dirección URL <https://app.creately.com/>

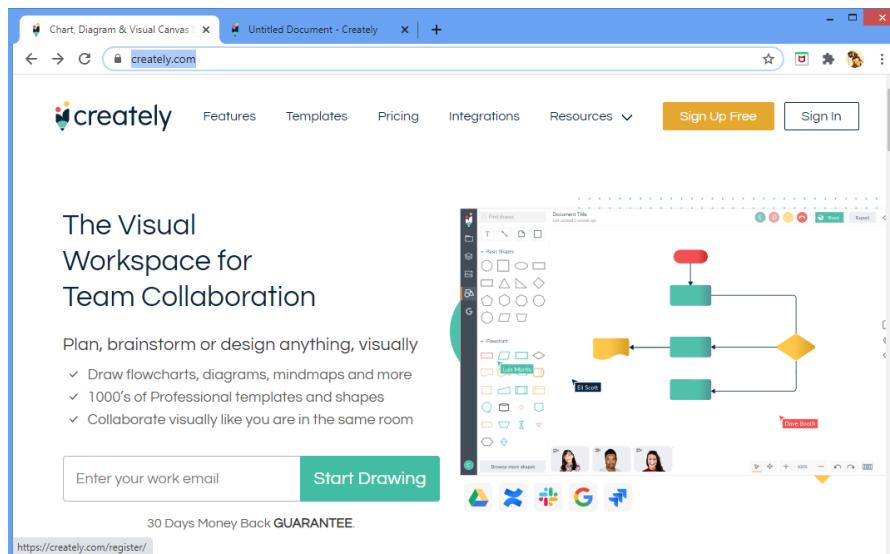


Figura 23: Captura de pantalla
Fuente. - Elaboración propia

Para registrar una cuenta libre, presiona el botón **Sign Up Free**, para registrarte.

En la página register, ingresa con una cuenta de google o el email de una compañía, en este caso ingresa con una cuenta de Gmail.

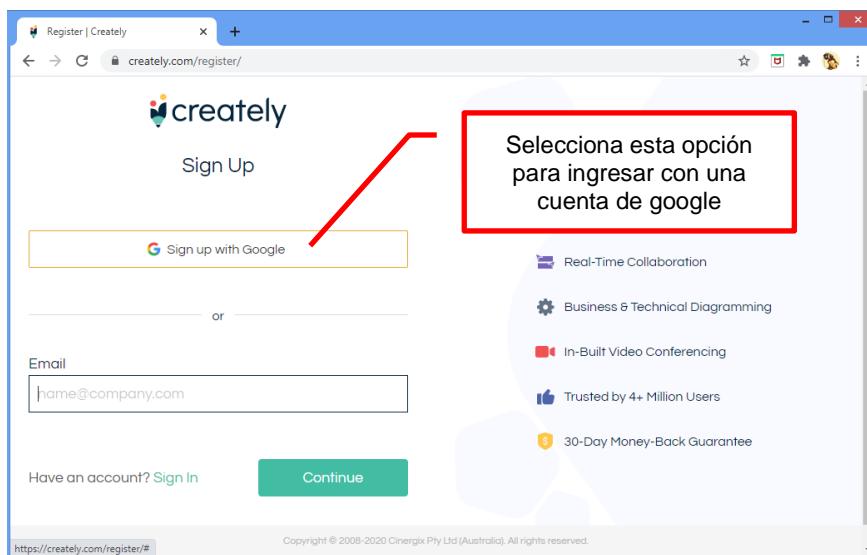


Figura 24: Captura de pantalla
Fuente. - Elaboración propia

En la ventana ingresa la cuenta de Gmail, en este caso selecciona una cuenta predefinida que está en google.

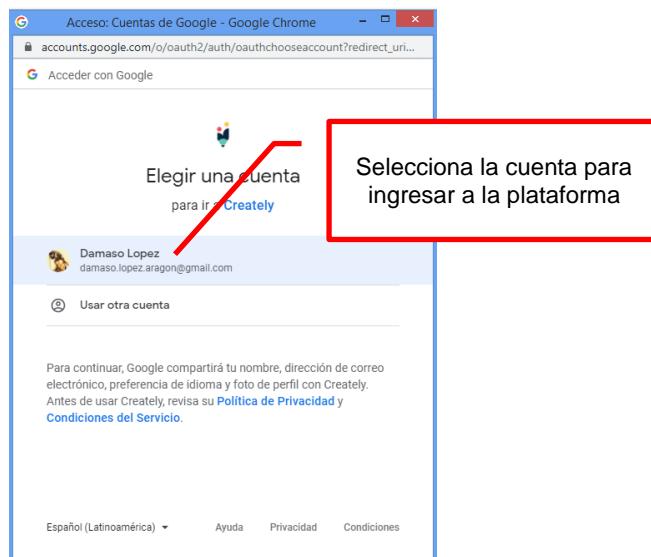


Figura 25: Captura de pantalla
Fuente. - Elaboración propia

Una vez que te has registrado, la plataforma te direcciona a la pantalla para agregar documentos (por ser un ingreso libre solamente puedes trabajar con tres documentos).

Para diseñar un sitemap, haz clic sobre el elemento **Untitled Document**.

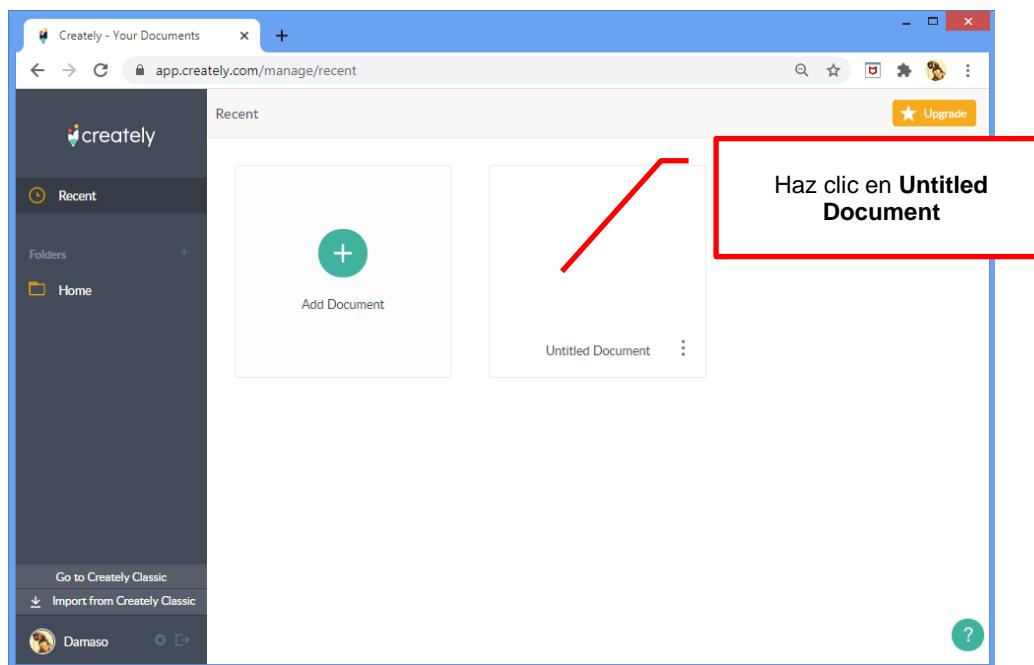


Figura 26: Captura de pantalla
Fuente. - Elaboración propia

A continuación, la plataforma carga el aplicativo para crear mapa de sitios, tal como se muestra.

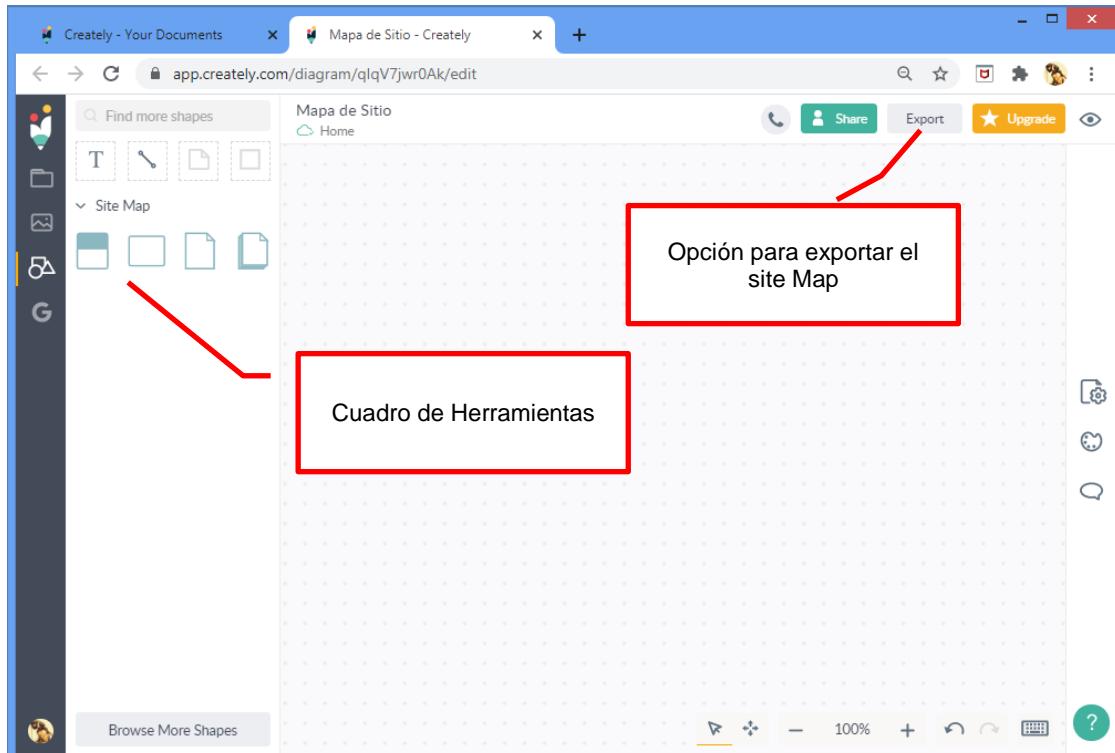


Figura 27: Captura de pantalla
Fuente. - Elaboración propia

Para insertar la página de inicio, selecciona y arrastra **Page Container** al documento. Para asignar un texto, selecciona la opción **Edit Text**, tal como se muestra.

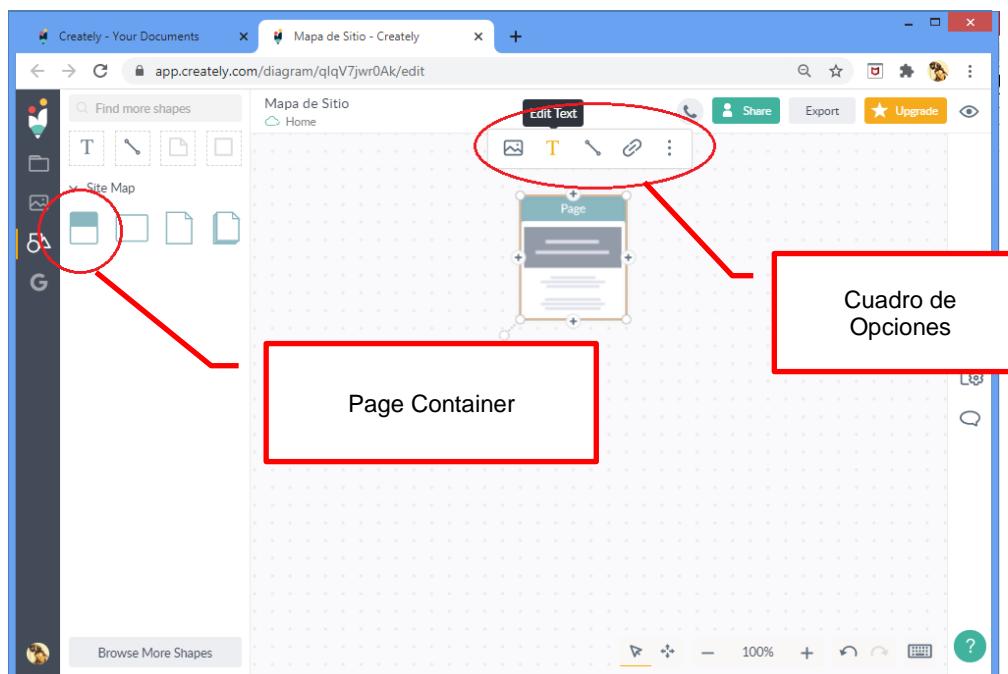


Figura 28: Captura de pantalla
Fuente. - Elaboración propia

A continuación, escribe un texto al objeto (Home). Para cambiar el color de letra, primero selecciona el texto, clic en la **Paleta de colores** y luego elige el color del texto.

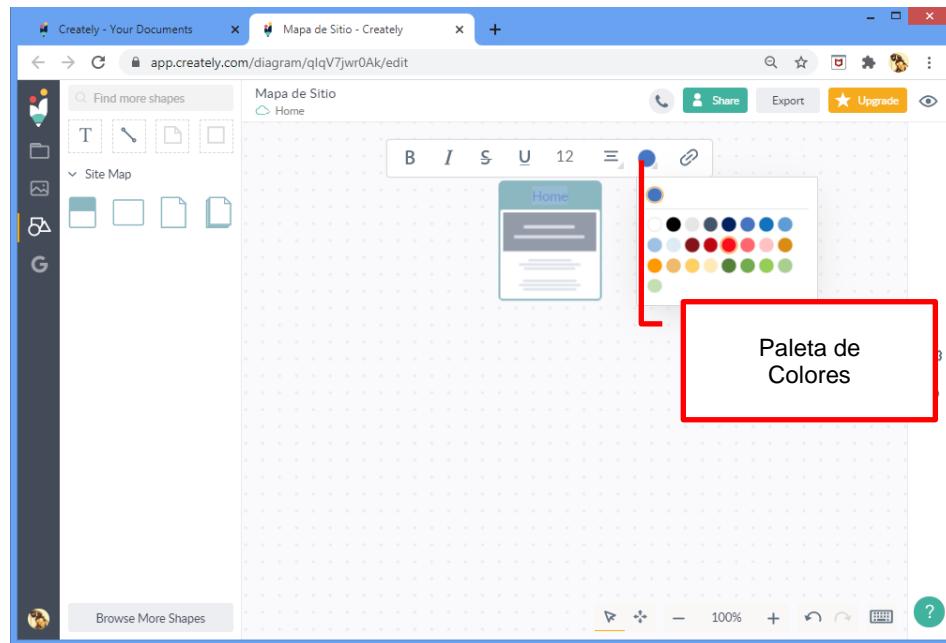


Figura 29: Captura de pantalla
Fuente. - Elaboración propia

Agrega tres page debajo del Page Container, los cuales representan las páginas principales del sitio.

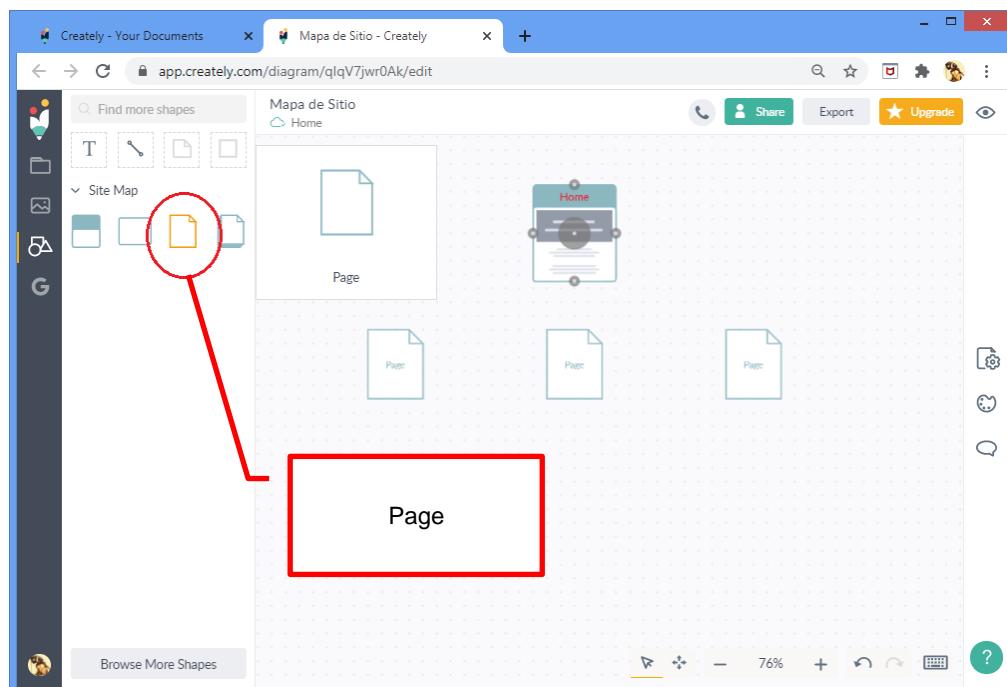


Figura 30: Captura de pantalla
Fuente. - Elaboración propia

Para darle estilo al Page, selecciona desde la opción **Style**, el estilo para dicho elemento.

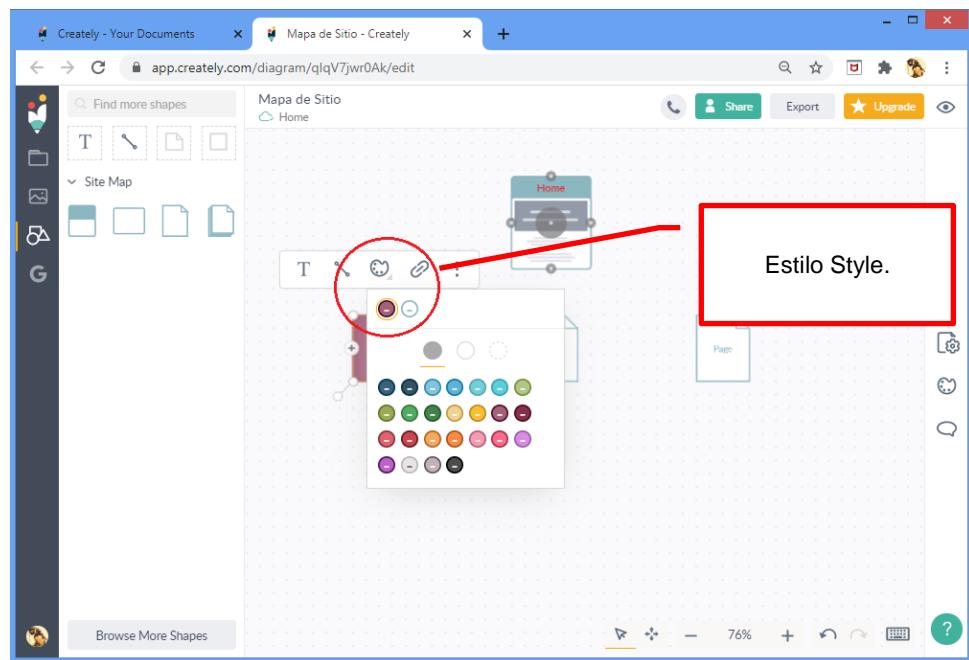


Figura 31: Captura de pantalla
Fuente. - Elaboración propia

Para realizar la conexión entre las páginas, selecciona la opción **Draw Connector** y luego arrastra uno de los border hacia la página HOME, tal como se muestra, las cuales se encontrarán conectadas.

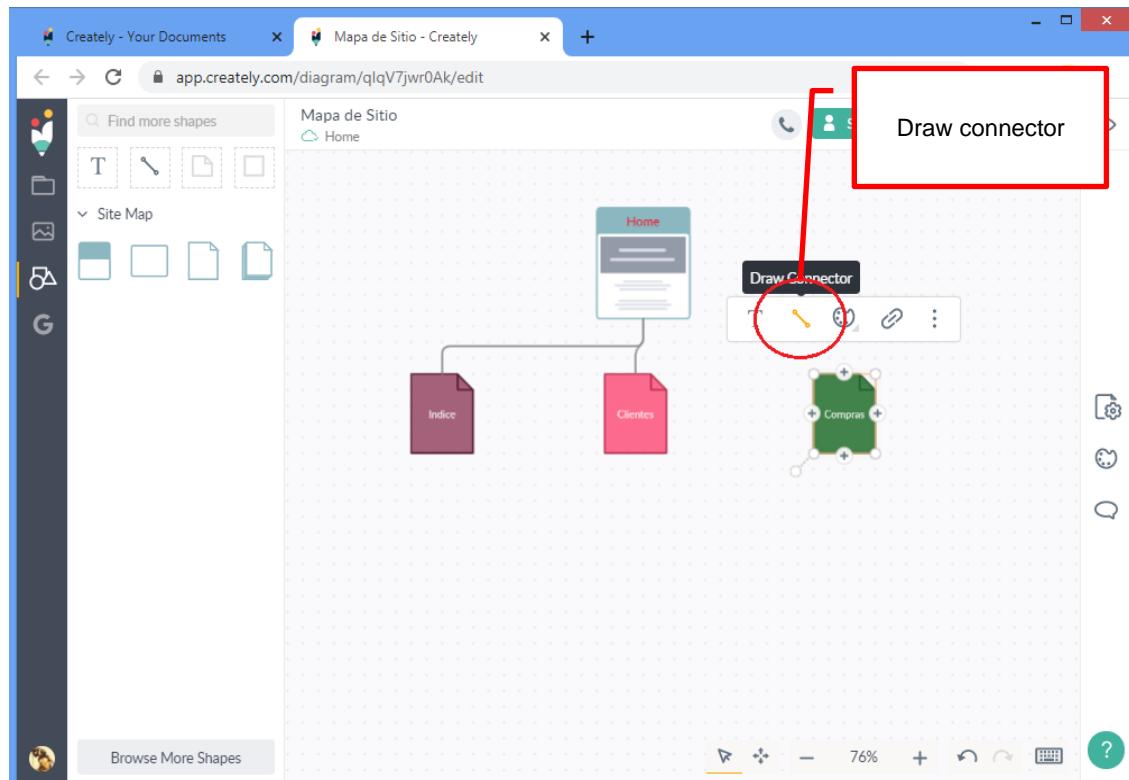


Figura 32: Captura de pantalla
Fuente. - Elaboración propia

Para cambiar el diseño del conector a tipo angular: selecciona el conector; luego, haz clic en la opción **Angled**, el conector se visualiza en forma lineal y angular en sus bordes.

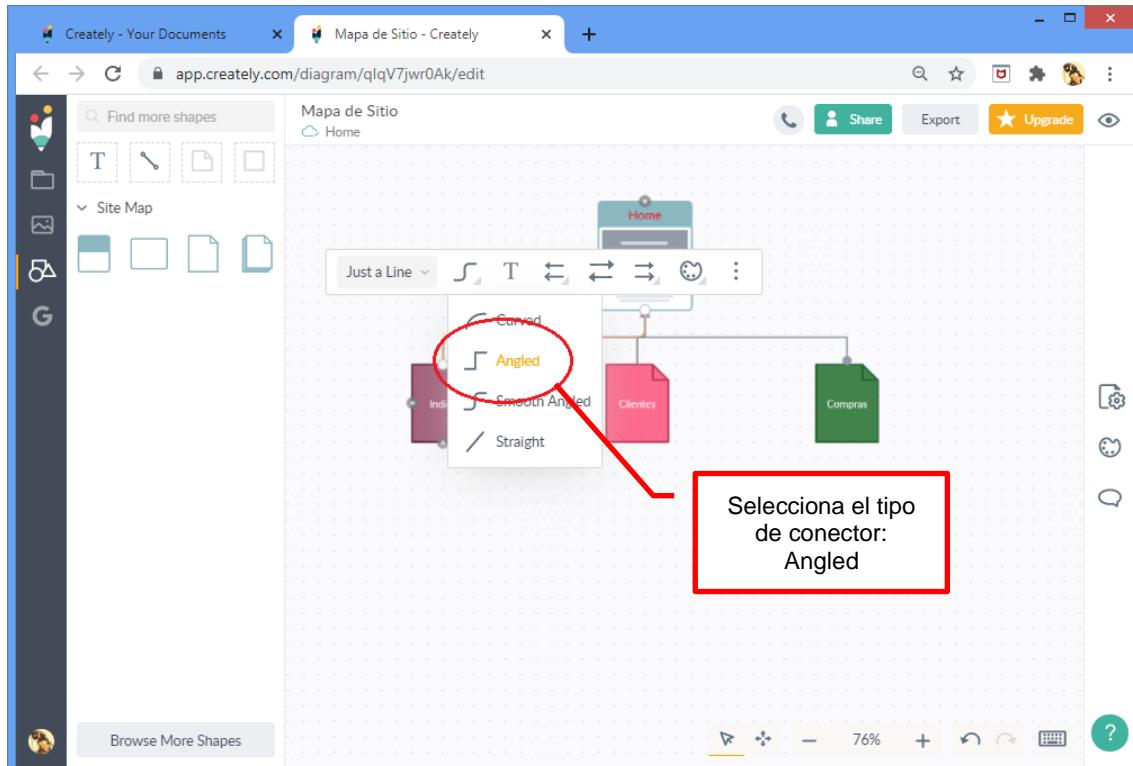


Figura 33: Captura de pantalla
Fuente. - Elaboración propia

Exportando el mapa de Sitio

Selecciona la opción **Export**, donde se visualiza varias opciones. Selecciona **Export as JPEG**, y se descarga el documento en formato JPG.

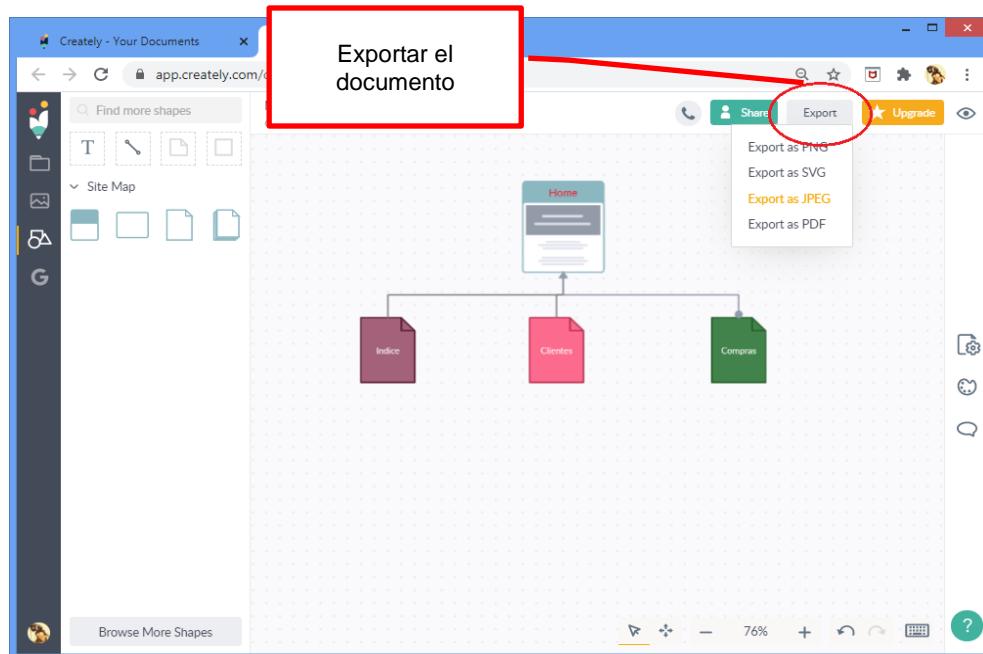


Figura 34: Captura de pantalla
Fuente. - Elaboración propia

A continuación, el archivo se descarga con el nombre del documento (Mapa de Sitio) y su extensión: JPG.

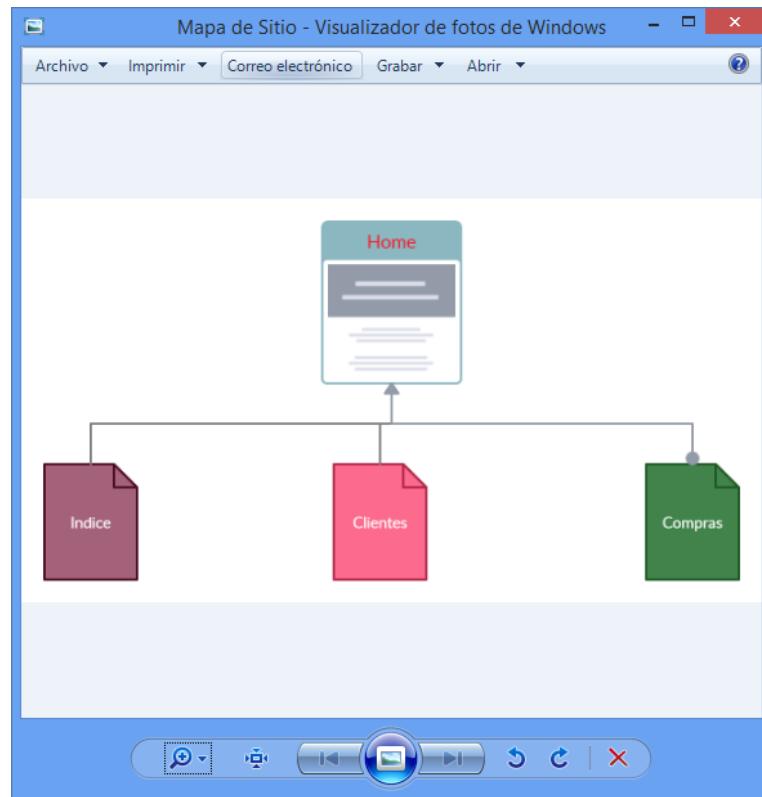


Figura 35: Captura de pantalla
Fuente. - Elaboración propia

Resumen

1. Internet se define como una red mundial de computadoras, distribuidos por todo el mundo, las cuales se encuentran interconectadas a través de satélites, líneas telefónicas, fibra óptica, etc., donde utilizando un protocolo de comunicaciones común (TCP/IP), permite a millones de usuarios estar conectados para compartir, intercambiar, publicar y extraer información.
2. Sitio Web o **Site**, es un conjunto de páginas web relacionadas a un dominio de Internet, el cual se puede visualizar a través de internet.
3. Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que solicita el usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo con los comandos solicitados.
4. El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.
5. Un dominio de Internet es un nombre de un sitio web. Cada dominio tiene que ser único en Internet, por ejemplo “elcomercio.pe” es el nombre del dominio de la página web del diario El Comercio.
6. Un navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web, la cual puede estar alojada en un servidor dentro de la World Wide Web o en uno local.
7. Etapa de un proyecto web permite desarrollar un sitio web efectivo, requiere una planeación bien elaborada. Puede ahorrar tiempo y dinero durante el proceso del proyecto si logra entenderlo en su totalidad antes de comenzarlo. A pesar de que los sitios varían en rango de contenido, desde unas cuantas páginas hasta sitios mucho más complejos con características sofisticadas, el proceso de desarrollo se mantiene igual.
8. Arquitectura web es una tarea que requiere conocimientos técnicos de construcción, funcionales y de diseño para sitios o páginas web. Con base en las preferencias del contenido se crea un mapa de la arquitectura del sitio para demostrar visualmente cómo se organiza el contenido y la estructura del sitio.
9. Un Sitemap o Mapa del Sitio es la estructura de enlaces de páginas web que definirá la buena o mala navegación de un sitio web.

Recursos

Pueden revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- http://www.hipertexto.info/documentos/internet_tecn.htm#IP
- <http://docentesistemas.webnode.es/contenidos-programaticos/grado-11/pagina-web-generalidades/que-es-un-hosting-/>
- <http://es.slideshare.net/josegregoriob/servidor-web-8451426>
- <http://www.duplika.com/que-es-el-hosting>
- <http://culturacion.com/como-escoger-un-servidor-para-alojar-mi-sitio-web/>
- <http://www.tiposde.org/internet/113-tipos-de-navegadores/>
- <http://spn314.blogspot.com/2013/11/cuantos-navegadores-de-internet-hay.html>
- <http://www.condo-consulting.com/test-de-usabilidad-que-son-y-para-que-me-sirven.b.aspx>
- <http://disenowebakus.net/pasos-clave-en-la-creacion-de-un-sitio-web.php>
- <http://novatosweb.blogspot.com/2010/12/etapas-que-debe-formar-tu-proyecto-web.html>
- <http://es.slideshare.net/cilp2884/etapas-de-un-proyecto-web>
- <http://es.slideshare.net/christian.bonner/arquitectura-de-la-informacion-de-sitios-web>

1.2. ELEMENTOS GRÁFICOS DE LA WEB

1.2.1. Wireframe

Como desarrolladores o diseñadores web, cuando se enfrenta a un nuevo proyecto con un cliente es de vital importancia lograr una buena comunicación y detectar realmente sus necesidades; pero esto es un problema cuando en algunos casos el cliente no sabe lo que necesita realmente o no puede explicarlo de manera adecuada.

Para evitar el tener que volver atrás en los proyectos y para mostrarle al cliente una mejor visualización del proyecto, nace la idea de los mockups, maquetas o wireframes.

Un wireframe básicamente es un boceto básico y de baja calidad del desarrollo de una página web o el diseño de una interfaz, cuya finalidad es mostrar al cliente un boceto rápido para facilitar la comunicación entre cliente y desarrollador.

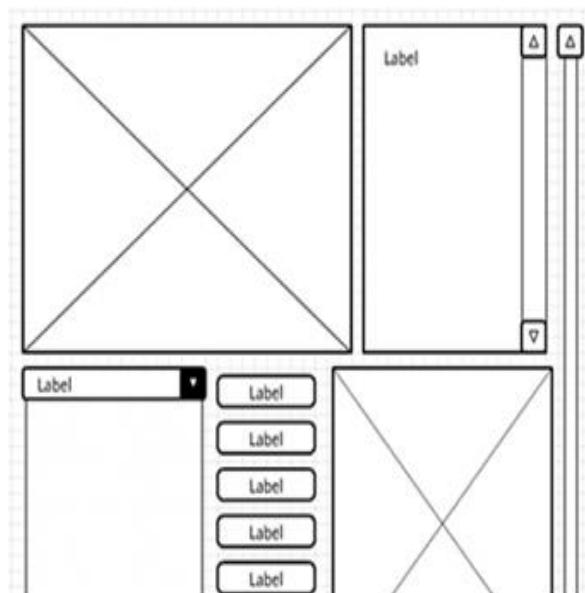


Figura 36: Wireframe

Fuente. - Tomado de <http://tecnologia.com/que-es-un-mockup-o-wireframe/>

El wireframe tiene como objetivo proporcionar la visualización del diseño o las funcionalidades de un site de manera netamente referencial y poder definir objetivos en una etapa temprana del proyecto, incluso hacer modificaciones de manera rápida, para evitar hacer estos cambios sobre un site o diseño real.

Para ciertos proyectos se debe tener en cuenta que, tan relevante como ubicar los elementos en pantalla es simular el funcionamiento, los wireframes funcionales, que no se limitan a una estructura estática, sino que también permiten la navegación.

1.2.2. Beneficios de los wireframes

- Comunicación entre todos los componentes del equipo de desarrollo y los usuarios.

- Participación, para integrar activamente a los usuarios en el desarrollo.
- Dan soporte a los diseñadores a la hora de escoger entre varias alternativas.
- Permiten a los diseñadores explorar diversos conceptos del diseño antes de establecer los definitivos.
- Permiten evaluar el sistema desde las primeras fases del desarrollo (facilitan la exploración de ideas sobre nuevos conceptos tecnológicos).
- Son esenciales para la documentación, tanto de conceptos funcionales del sistema como de tareas concretas del mismo.
- Son el primer paso para que ideas abstractas sean concretas, visibles y testables.
- Fomentan la iteratividad.
- Mejoran la calidad y la completitud de las especificaciones funcionales del sistema.
- Son herramientas de propósito general, pues sirven para comprobar la fiabilidad técnica de una idea, clarificar requisitos que quedaron “indeterminados” o ver cómo responde con el resto de la aplicación.

1.2.3. Tipos de wireframes: baja y alta fidelidad

Se puede tener distintos tipos de prototipos, que en ningún caso incluyen el diseño gráfico puesto que este se aplica en una etapa posterior.

1. **Wireframe de baja fidelidad:** son aquellos prototipos que representan cómo estarán organizados los elementos en las páginas del sistema sin entrar en detalle.
- **Sketching:** un sketch es un boceto rápido e informal que se realiza en papel para transmitir una idea o concepto con rapidez y claridad. Suele realizarse durante las entrevistas iniciales como técnica eficaz para comunicar al cliente conceptos o proponer alternativas a un problema; también es muy útil durante un brainstorming o en las reuniones con el equipo de trabajo. Es una forma eficaz de comenzar el prototipado de un sitio web, pues permite trabajar ágilmente con varias ideas y esquematizar las páginas.

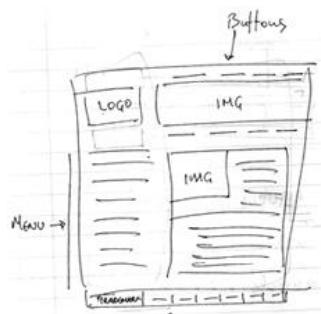


Figura 37: Tipos de Wireframe

Fuente. - Tomado de <http://tecnologia.com/que-es-un-mockup-o-wireframe/>

- **Wireframe:** permite crear una referencia visual de la estructura de una página web, definiendo al detalle el contenido y su distribución visual, organizando así la información a nivel de página.

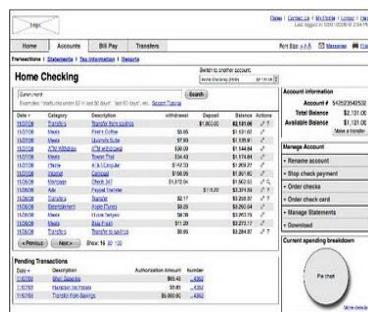


Figura 38: Diseño de Wireframe
Fuente. - Tomado de <http://olgacarreras.blogspot.com.es/2007/02/wireframes.html>

2. **Wireframe de alta fidelidad:** son prototipos o maquetas dinámicas, normalmente en (X)HTML, que simulan o tienen implementadas partes del sistema final a desarrollar. Pueden ser de muy distintos tipos, completo o parcial, horizontal o vertical, etc.

1.2.4. Programas online para crear wireframe de baja fidelidad

Existe una variedad de softwares que permiten crear tipos de wireframes:

Nombre	Ubicación
Lucidchart	https://www.lucidchart.com
Mockingbird	https://gomockingbird.com/
Mockflow	http://www.mockflow.com/
Moqups	https://moqups.com
Wireframe.cc	https://wireframe.cc

Figura 39: Tabla de programas online para wireframe
Fuente. – Elaboración Propia

LABORATORIO 2

Diseñando un WireFrame con MOCKINGBIRD

Se pide diseñar un WIREFRAME utilizando el aplicativo MOCKING BIRD.

Para ingresar al **Mockingbird**, ingresa la siguiente dirección: <https://gomockingbird.com/>

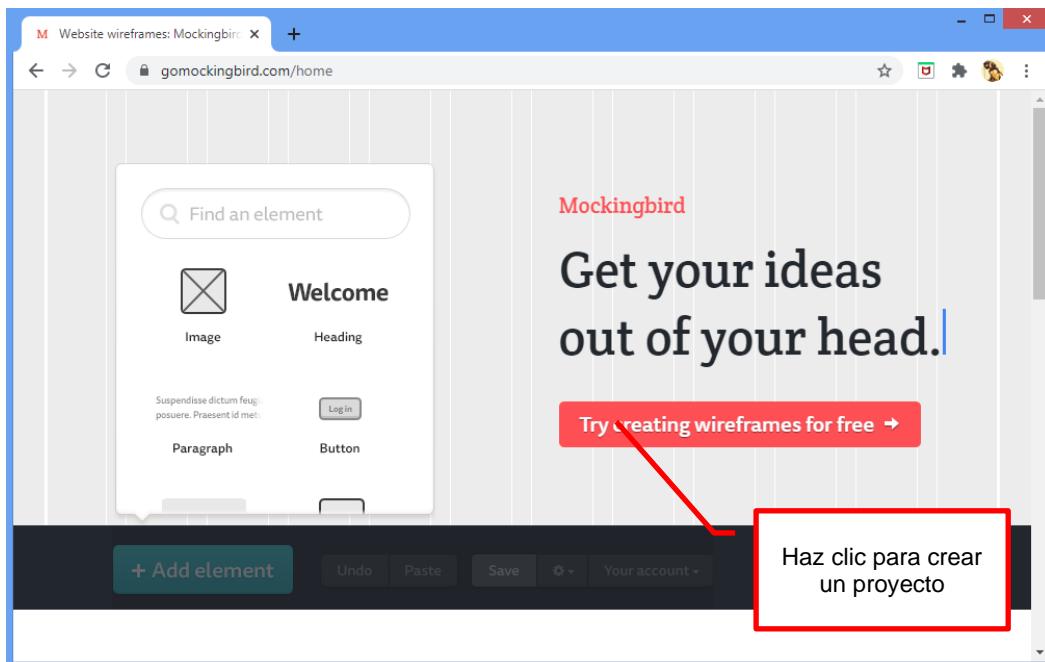


Figura 40: Captura de pantalla
Fuente. - Elaboración propia

Cargando la página **MockingBird**.

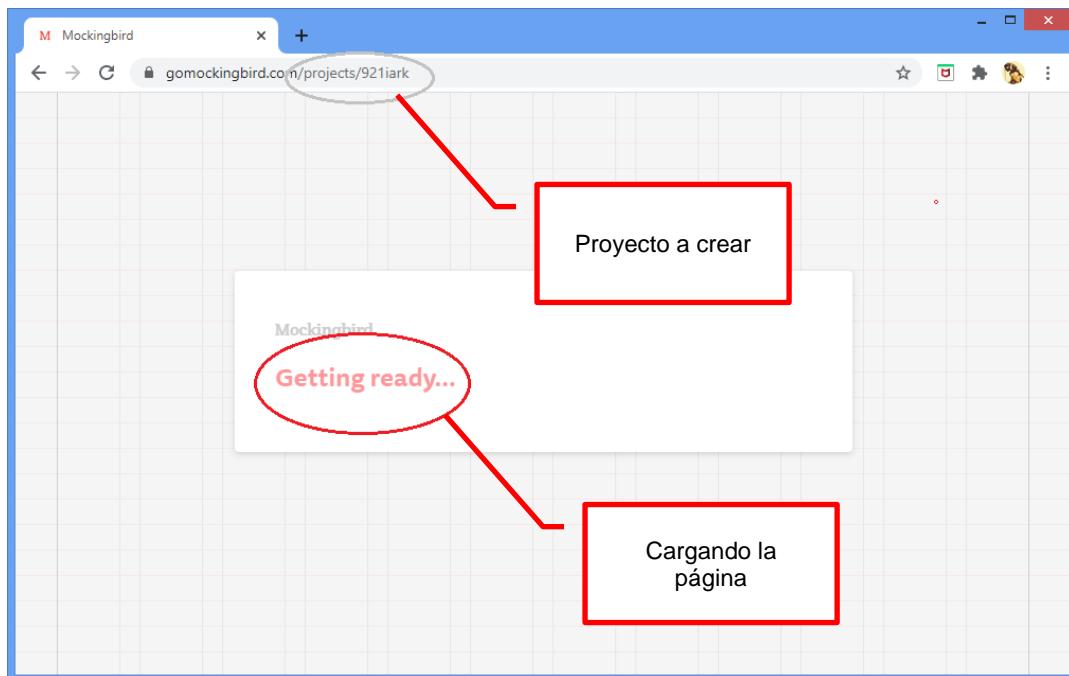


Figura 41: Captura de pantalla
Fuente. - Elaboración propia

IDE del MockingBird

Al cargar el aplicativo se visualiza el siguiente IDE, tal como se muestra.

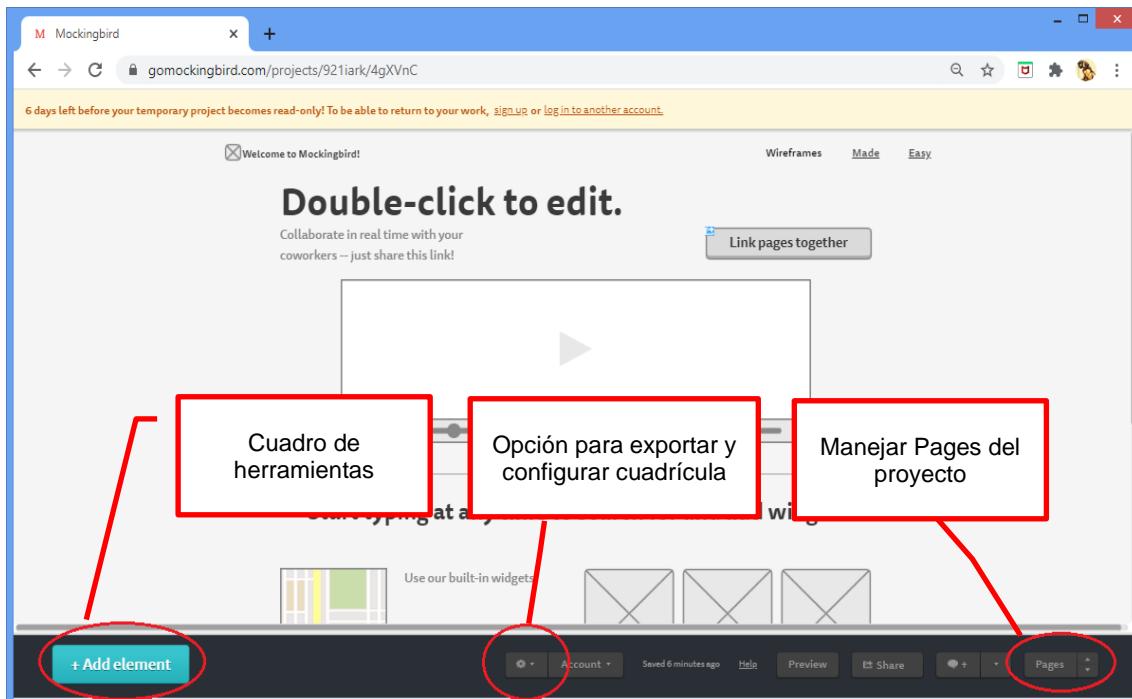


Figura 42: Captura de pantalla

Fuente. - Elaboración propia

Agregando una página de diseño

Despliega la opción **Page**. Allí se visualiza una opción a **New page**, escribe el nombre y presiona **ENTER**.

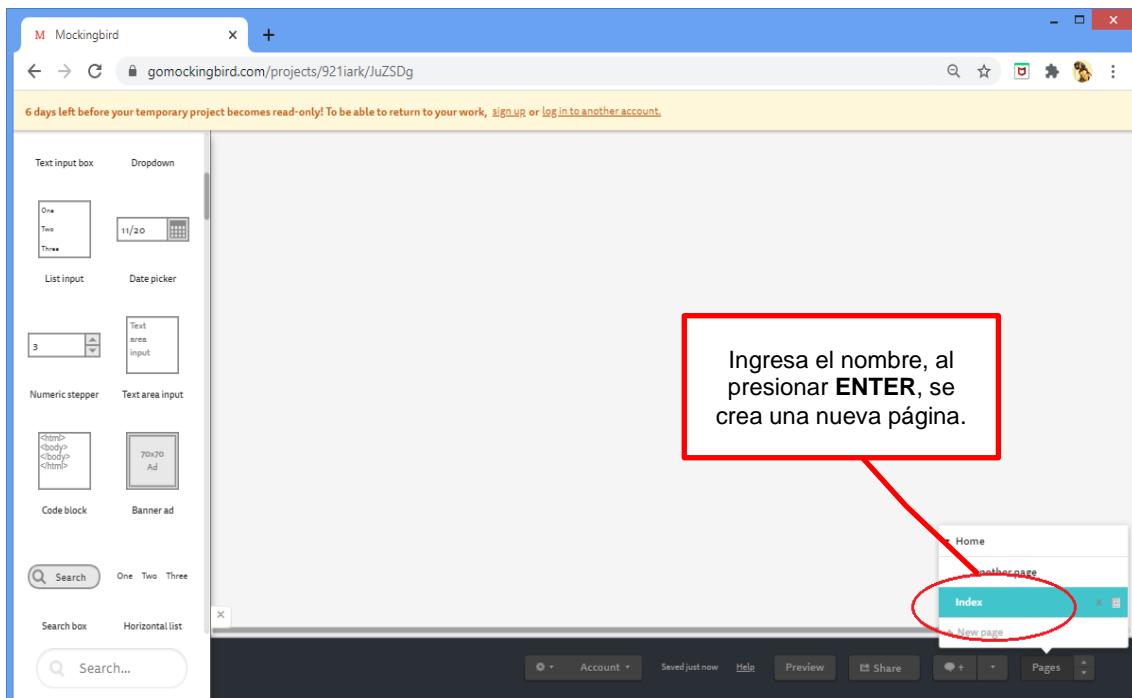


Figura 43: Captura de pantalla

Fuente. - Elaboración propia

Para configurar la vista del wireframe en cuadrículas, desde la opción **Configuration**:

1. Selecciona la opción **Snap to grid**.
2. Selecciona desde la opción **Columns: 960gs (12 columns)**, para visualizar 12 columnas en la cuadrícula.

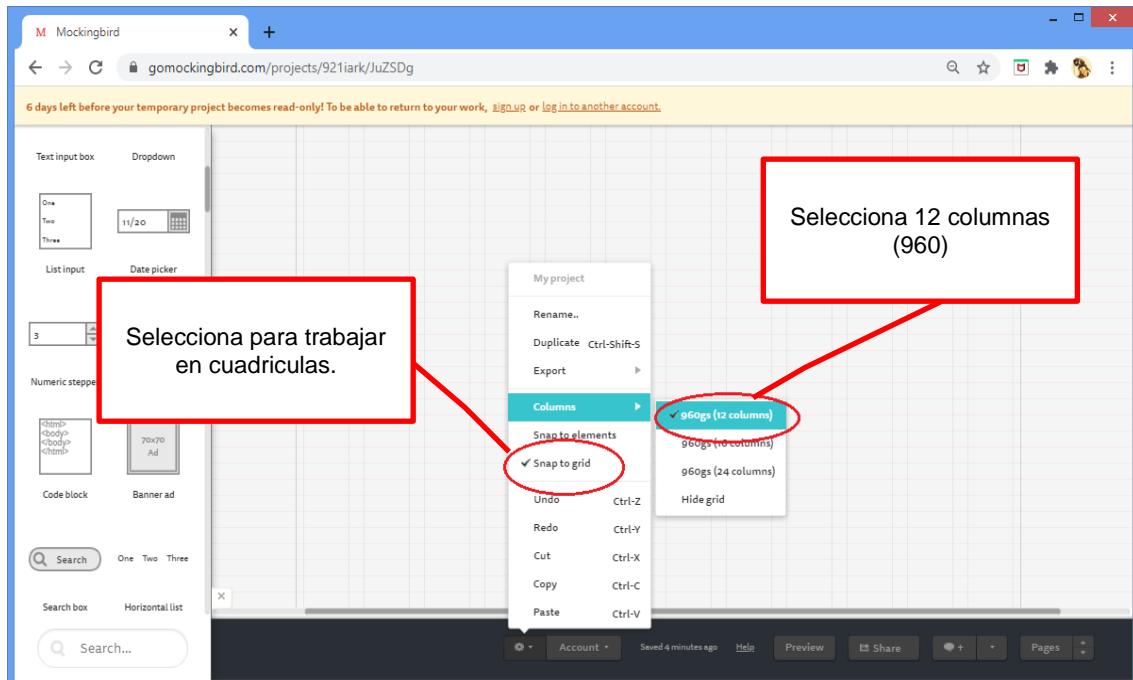


Figura 44: Captura de pantalla
Fuente. - Elaboración propia

Trabajando con el diseño del wireframe

Para la búsqueda de un control, escribe en la opción **Search**, las iniciales del nombre del control, y el buscador mostrará los controles que coincidan con la búsqueda, tal como se muestra en la figura.

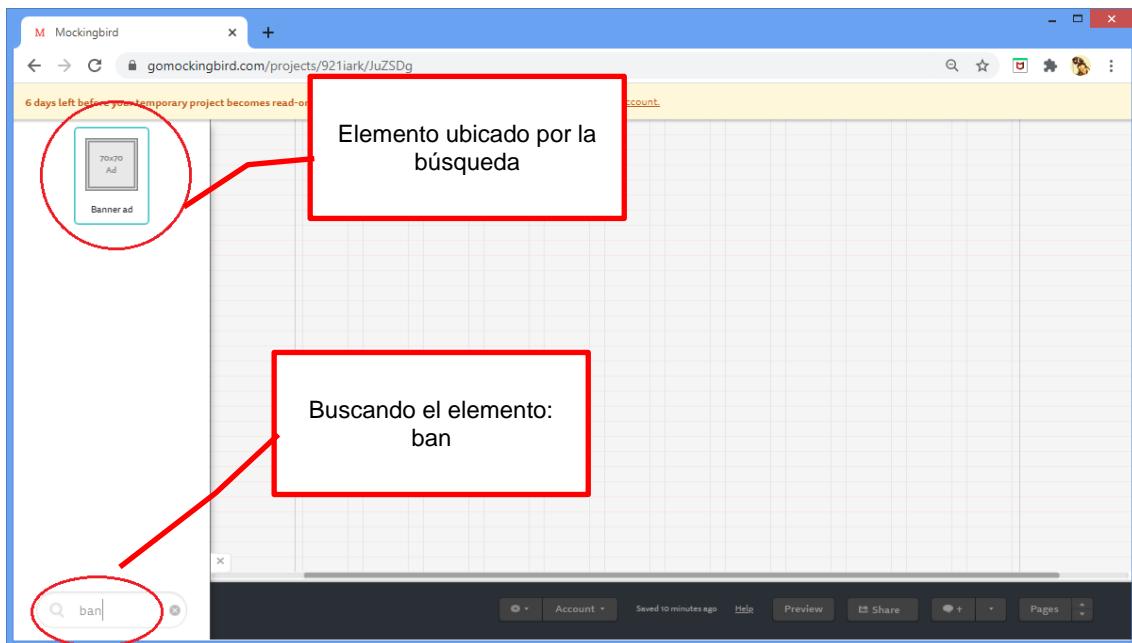


Figura 45: Captura de pantalla

Fuente. - Elaboración propia

Para colocar el elemento a la grilla del wireframe, arrástralolo a la página y luego dimensiónalo dentro de la cuadrícula.

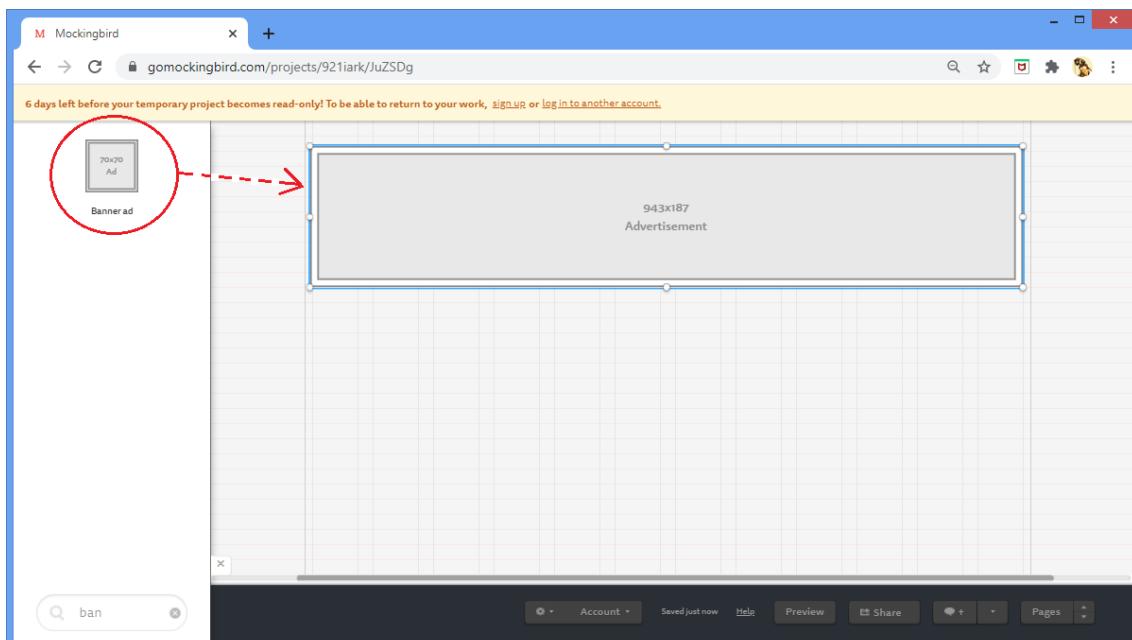


Figura 46: Captura de pantalla

Fuente. - Elaboración propia

Agregando menú de navegación

Escribe en la opción **Search Button Bar**, donde se visualiza dos elementos, tal como se muestra.

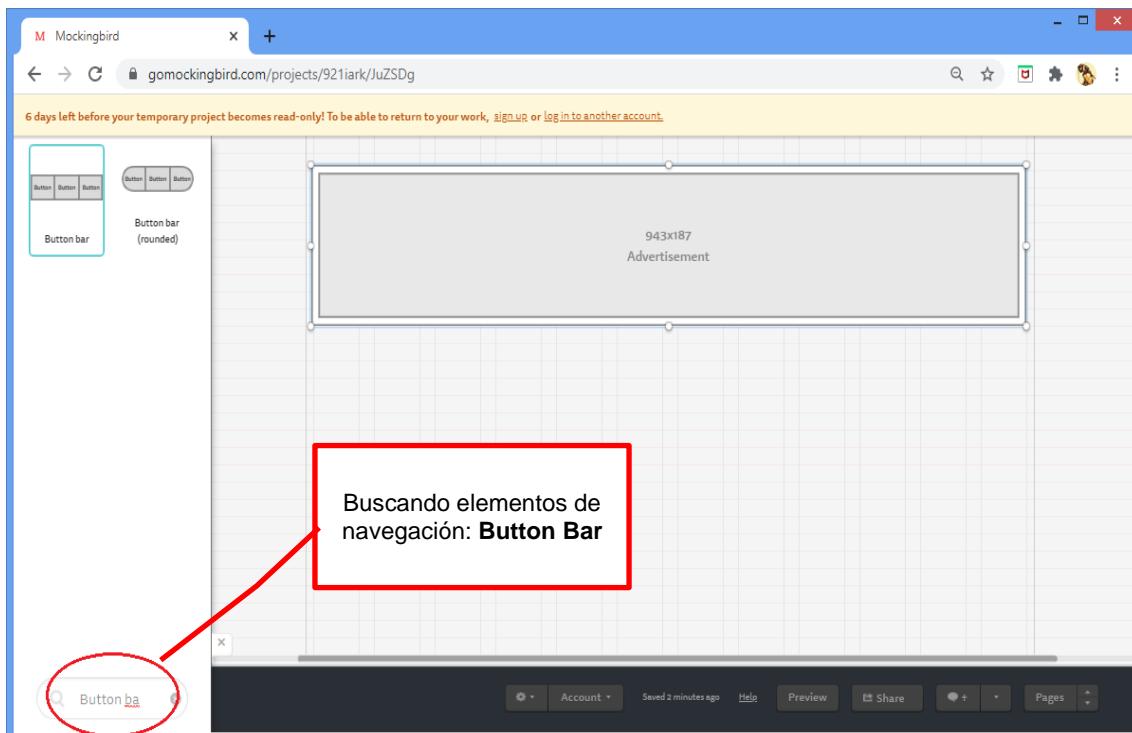


Figura 47: Captura de pantalla
Fuente. - Elaboración propia

Arrastra, desde el cuadro de herramientas, el control (**Button bar**) Barra de menú.

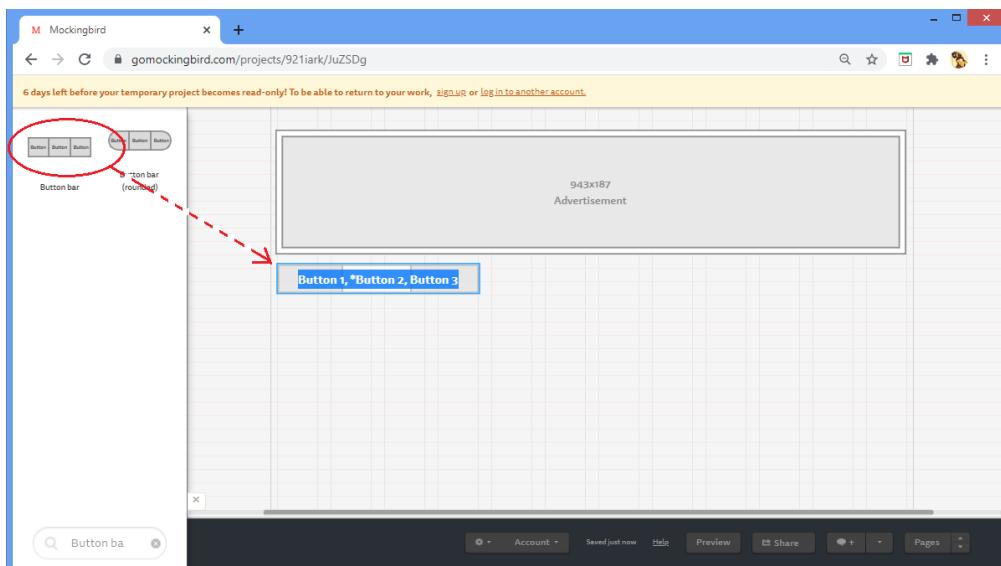


Figura 48: Captura de pantalla
Fuente. - Elaboración propia

Agregado el elemento, edita el menú bar. A continuación, escribe la lista de los elementos, los cuales se separan por una coma, tal como se muestra. Al finalizar presionar **ENTER**.

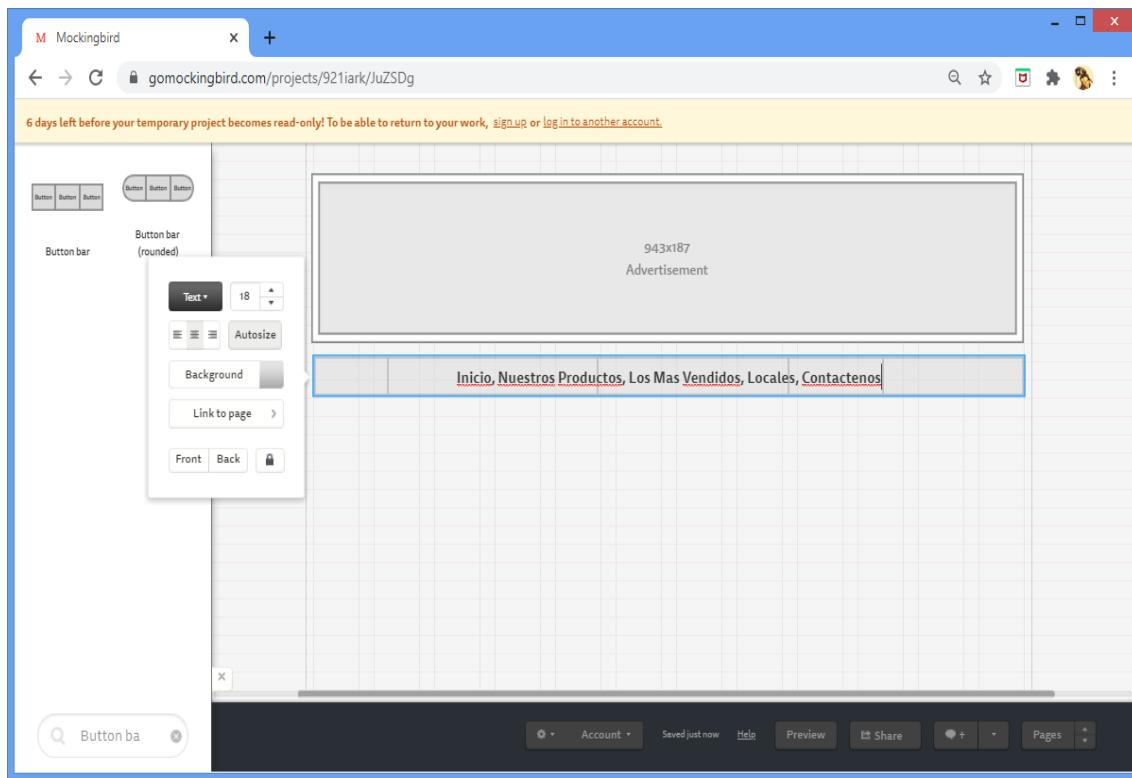


Figura 49: Captura de pantalla
Fuente. - Elaboración propia

Para cambiar el color de fondo o BackGround, desde la lista de **Propiedades**, selecciona la opción **BackGround** y haz clic al color que elijas como fondo. A continuación, extiende el control a lo largo de la página. El menú diseñado se visualiza de esta forma:

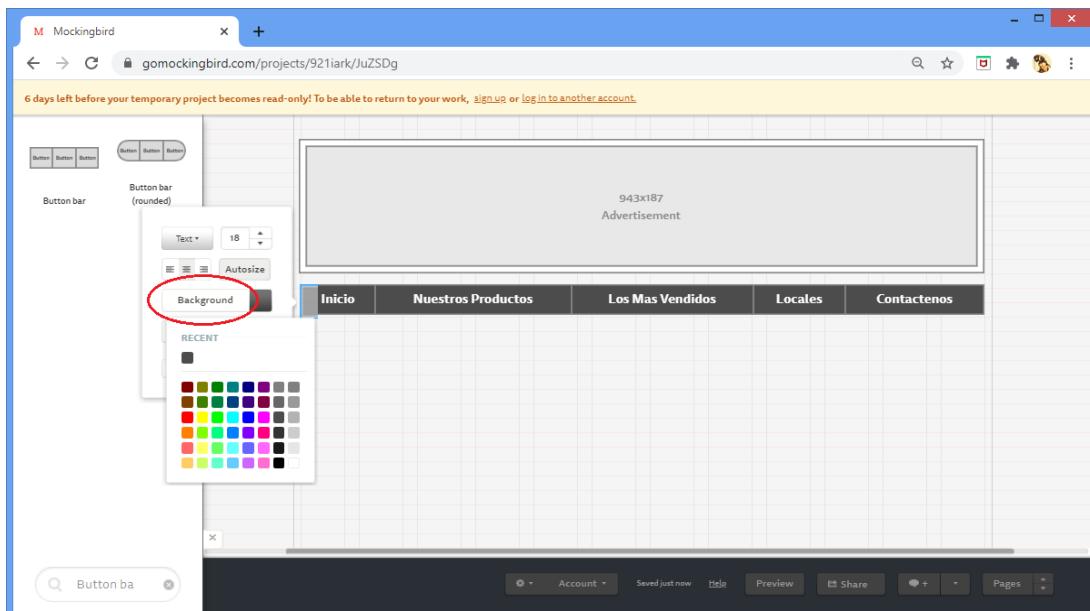


Figura 50: Captura de pantalla
Fuente. - Elaboración propia

Luego, termina el diseño del wireframe tal como se muestra.

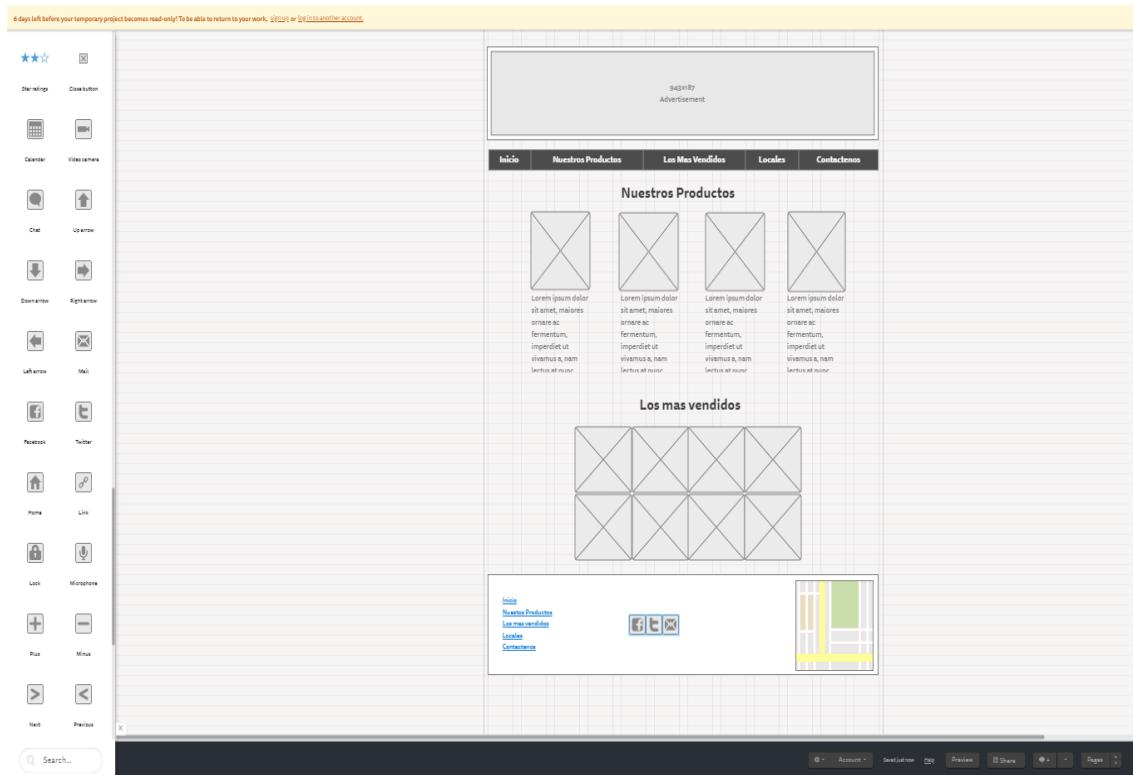


Figura 51: Captura de pantalla
Fuente. - Elaboración propia

Al finalizar de exportar el Wireframes, desde la opción **Configuration**, selecciona la opción **EXPORT** en formato PNG o PDF.

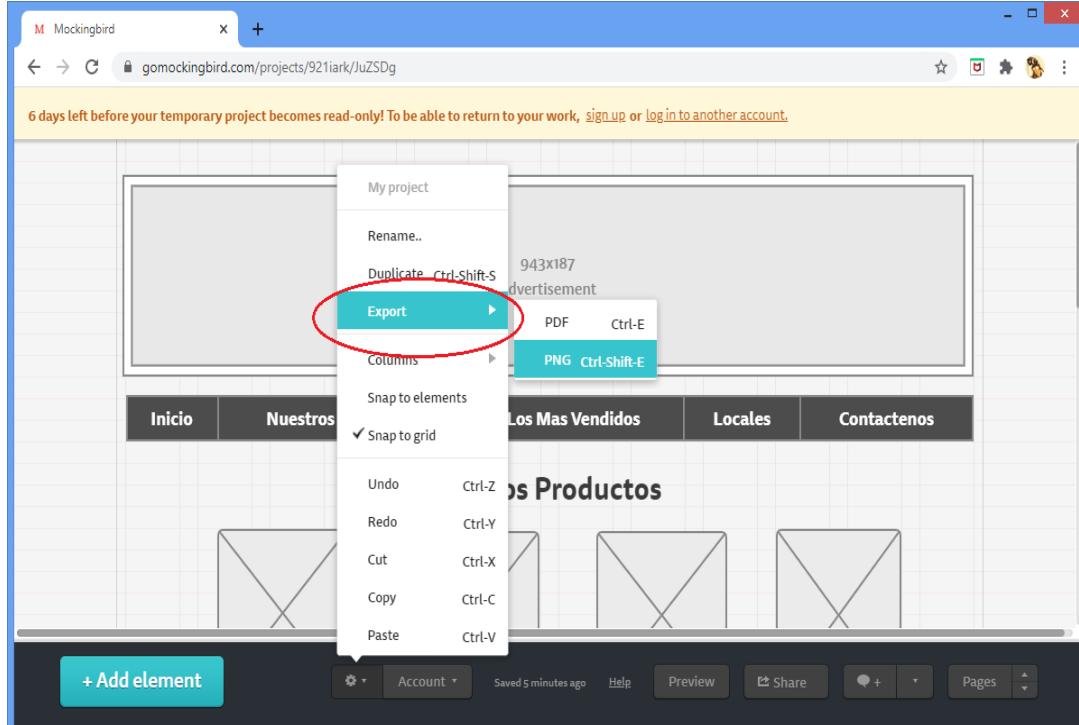


Figura 52: Captura de pantalla
Fuente. - Elaboración propia

Al descargar en formato PDF se guarda en un archivo, tal como se muestra.

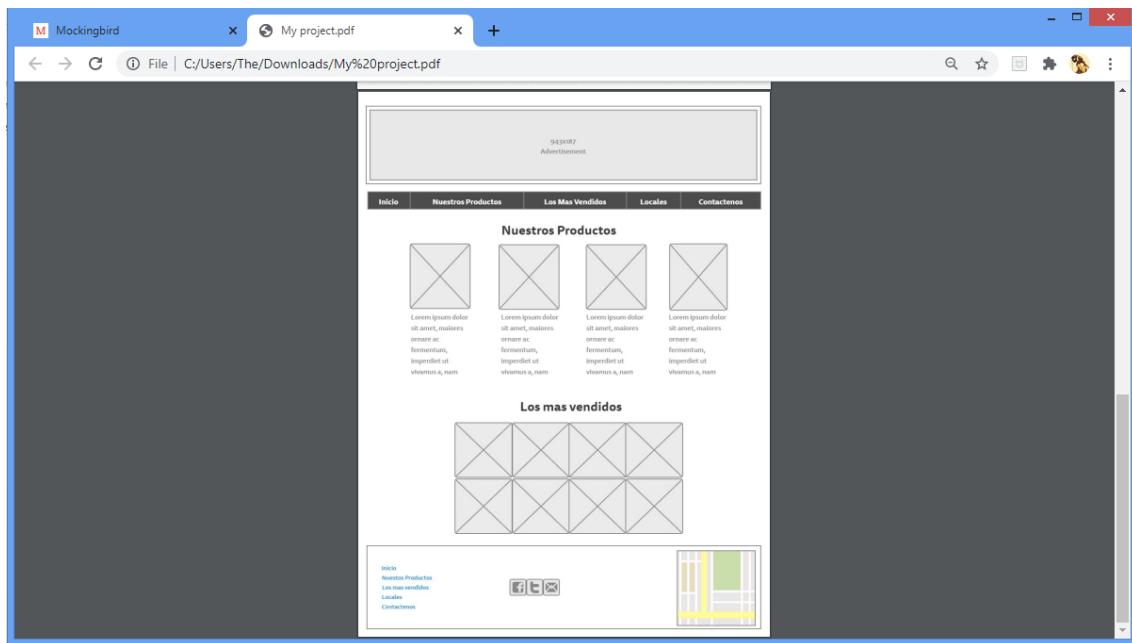


Figura 53: Captura de pantalla
Fuente. - Elaboración propia

Resumen

1. Un wireframe básicamente es un boceto básico y de baja calidad del desarrollo de una página web o el diseño de una interfaz, cuya finalidad es mostrar al cliente un boceto rápido para facilitar la comunicación entre cliente y desarrollador.
2. El wireframe tiene como objetivo proporcionar la visualización del diseño o las funcionalidades de un site de manera netamente referencial y poder definir objetivos en una etapa temprana del proyecto.
3. Entre las ventajas:
 - o Comunicación entre todos los componentes del equipo de desarrollo y los usuarios.
 - o Participación, para integrar activamente a los usuarios en el desarrollo.
 - o Permiten a los diseñadores explorar diversos conceptos del diseño antes de establecer los definitivos.
 - o Permiten evaluar el sistema desde las primeras fases del desarrollo (facilitan la exploración de ideas sobre nuevos conceptos tecnológicos).
 - o Son el primer paso para que ideas abstractas sean concretas, visibles y testeables.
4. Son herramientas de propósito general, pues sirven para comprobar la fiabilidad técnica de una idea, clarificar requisitos que quedaron “indeterminados” o ver cómo responde con el resto de la aplicación.
5. Wireframe de baja calidad, son aquellos prototipos que representan cómo estarán organizados los elementos en las páginas del sistema sin entrar en detalle.
6. Un sketch es un boceto rápido e informal que se realiza en papel para transmitir una idea o concepto con rapidez y claridad.
7. Un wireframe, permite crear una referencia visual de la estructura de una página web, definiendo al detalle el contenido y su distribución visual, organizando así la información a nivel de página.
8. Wireframe de alta fidelidad, son prototipos que simulan o tienen implementadas partes del sistema final a desarrollar. Pueden ser de muy distintos tipos, completo o parcial, horizontal o vertical, etc.

Recursos

Revisa los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- o <http://tednologia.com/que-es-un-mockup-o-wireframe/>
- o <http://olgacarreras.blogspot.com.es/2007/02/wireframes.html>
- o <https://webdesdecero.com/wireframes-que-son-y-como-crearlos/>
- o <https://rockcontent.com/es/blog/wireframe/>
- o <https://neoattack.com/neowiki/wireframe/>



HTML5 Y CSS3

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web aplicando etiquetas HTML5, estilos CSS 3.

TEMARIO

2.1 Tema 3 : Estructura básica HTML

- 2.1.1 : Definición
- 2.1.2 : Características
- 2.1.3 : Estructura de una página HTML
- 2.1.3.1 : Etiquetas: html, head, meta, title, body
- 2.1.4 : Etiquetas básicas
- 2.1.4.1 : Encabezados **<h1>, <h2>, ..., <h6>**
- 2.1.4.2 : Párrafos **<p>**
- 2.1.4.3 : Líneas horizontales **<hr>**
- 2.1.4.4 : Saltos de línea **
**
- 2.1.4.5 : Comentarios **<!-- comentario -->**
- 2.1.4.6 : Etiquetas de lista: Listas ordenadas, desordenadas y de definición
- 2.1.4.7 : Texto enfatizado ****
- 2.1.4.8 : Texto reforzado ****
- 2.1.5. : Etiquetas para hipervínculos **<a>** y atributo **target**
- 2.1.6. : Estilos en línea: uso del atributo **style**
- 2.1.6.1 : Color de fondo
- 2.1.6.2 : Color de texto
- 2.1.6.3 : Alineación del párrafo
- 2.1.6.4 : Uso de fuente: tamaño, decoración, estilo
- 2.1.7. : Etiqueta de imagen **** y atributos **src**, **alt**, **width** y **height**
- 2.1.8. : Etiquetas para figuras **<figure>** y **<figcaption>**
- 2.1.9. : Uso de anclas: Enlace a un lugar de la misma página (**id**)

2.2 Tema 4 : Estilos CSS

- 2.2.1. : Selector de estilo CSS
- 2.2.1.1 : Tipos de estilos
- 2.2.2 : Selectores CSS
- 2.2.2.1 : Selector universal
- 2.2.2.2 : Selector de etiqueta
- 2.2.2.3 : Selector descendente
- 2.2.2.4 : Selector de clase

- 2.2.2.5. : Selector de id
- 2.2.2.6. : Combinación de selectores

ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5 y hojas de estilo CSS3.
- Los alumnos diseñan páginas web utilizando bloques de contenidos.

2.1 ESTRUCTURA BÁSICA HTML

2.1.1 Definición

HTML es un lenguaje de marcado que permite indicar la estructura del documento mediante etiquetas.

Este lenguaje ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas, ya que siguen un esquema de árbol donde existe un elemento raíz y donde el resto de las etiquetas se insertan de manera lógica y estructurada. Además, se trata de un fichero de texto, y solo se necesita un editor como el bloc de notas para generar un documento HTML.

2.1.2 Características

Para diseñar una página Web, los diseñadores de la WWW decidieron que se debían cumplir las siguientes características:

- Puede ser creado y editado con cualquier editor básico de textos.
- Es multiplataforma, o sea, puede ser visualizado por cualquier navegador de cualquier sistema operativo.
- No diferencia entre mayúsculas y minúsculas.
- Utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en su navegador el texto así como también las imágenes y demás elementos en la pantalla del ordenador.
- Cada elemento de un documento HTML consta de una etiqueta de comienzo, un bloque de texto y una etiqueta de fin.
- Lenguaje estático.
- Es utilizado para la creación de páginas web.
- Los documentos HTML son documentos de hipertexto que aparecen enlazando a otros documentos.
- Es un estándar reconocido por todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C.

2.1.3 Estructura de una página HTML

Un documento HTML no es más que un archivo de texto. La extensión de su nombre suele ser .html o .htm.

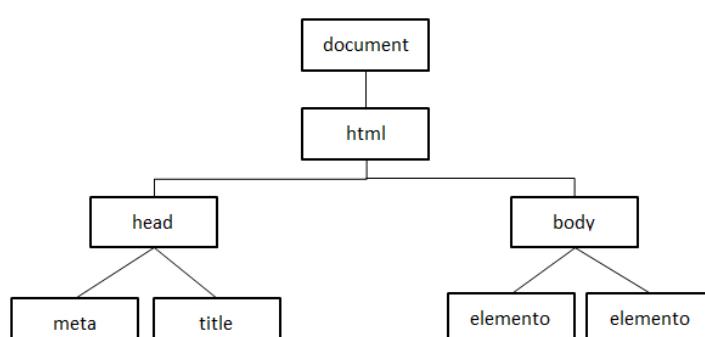


Figura 54. Estructura HTML
Fuente. - Elaboración propia

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página; por ejemplo, título e idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto, imágenes, videos, audio, etc.

Estructura básica



Figura 55: Estructura del HTML

Fuente. - Tomado de <https://disenowebakus.net/domine-html-y-dhtml-primeros-pasos.php>

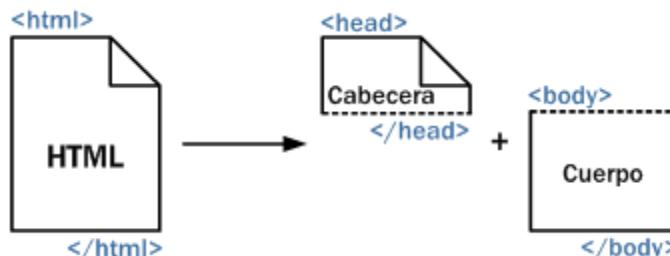


Figura 56: HTML

Fuente. - Tomado de <https://disenowebakus.net/domine-html-y-dhtml-primeros-pasos.php>

2.1.3.1 Etiquetas: html, head, meta, title, body

Para crear un documento, HTML comenzará con estos elementos:

- <DOCTYPE>
- <html>
- <head>
- <body>

Estos elementos se combinan para describir la estructura básica de la página.

<DOCTYPE>

La etiqueta doctype declara el tipo de documento, por lo que está señalando que este es un documento HTML.

```
<!doctype html>
```

<html>

A continuación, se comienza a construir la estructura del documento HTML definiendo las etiquetas **<html></html>**.

Esta etiqueta delimita el documento HTML, indicando al navegador el comienzo y fin de la página html. En esta etiqueta se utiliza el atributo **lang** que servirá para especificar el idioma del documento, en este caso español; por tanto, se asigna el valor **es**.

```
<!doctype html>
<html lang="es">

</html>
```

<head>

La palabra **head** viene del inglés “cabeza” y su función es delimitar cabecera del documento. Sus etiquetas son: **<head>** y **</head>**.

La cabecera es la sección apropiada para incluir información sobre el documento, la mayoría no será mostrada a los lectores. Para incluir esta información existen diversos elementos, pero el más importante es el siguiente:

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Untitled Document</title>
</head>
<body>

</body>
</html>
```

La directiva **<title>** sirve para especificar el título del documento y es visualizada en la barra de título de la ventana correspondiente al programa navegador.

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Pagina Principal</title>
</head>
<body>

</body>
</html>
```

La directiva **<meta>** es utilizada en el documento, se emplea para identificar propiedades del documento; por ejemplo: el autor, el título y la descripción que mostrarán los buscadores, etc.

name	Este atributo identifica un nombre de propiedad.
content	Este atributo especifica el valor de una propiedad.
scheme	Este atributo especifica un esquema que se usará para interpretar el valor de la propiedad.
http-equiv	Este atributo puede utilizarse en lugar del atributo name. Los servidores HTTP utilizan este atributo para obtener información sobre los encabezados del mensaje de respuesta HTTP.
lang	Este atributo se usa para especificar un idioma.
charset	Establece el tipo de codificación del documento.

```
<!doctype html>
<html>
<head>
    <title>Pagina Principal</title>
    <meta charset="utf-8">
    <meta name="description" content="Mi pagina">
    <meta name="keywords" content="HTML,CSS">
    <meta charset="utf-8">
</head>
<body>

</body>
</html>
```

La directiva **<link>** es usada para incorporar estilos, códigos Javascript, imágenes o íconos desde archivos externos. Uno de los usos más comunes para **<link>** es la incorporación de archivos con estilos CSS:

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Pagina Principal</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
</body>
</html>
```

<body>

La etiqueta **<body>** delimita el cuerpo del documento HTML. Este elemento contiene todo el contenido visible de un documento HTML (párrafos de texto, imágenes, tablas). La etiqueta **<body>** es compatible con todos los navegadores.

```
<!DOCTYPE html>
<html>
<head>
    <title>Título de la página</title>
    <meta charset=" utf-8">
    <meta name="title" content="Arquitectura Web">
    <meta name="descripción" content="Diseño de la Web">
    <link href="css/estilos.css" rel="stylesheet" />
</head>
<body>
    Esto es una línea de texto
</body>
</html>
```

2.1.4 Etiquetas básicas

2.1.4.1 Encabezados **<h1>**, **<h2>**, ..., **<h6>**

Definen el tamaño de un título o cabecera.

<h1> brinda el tipo de letra más grande

<h6> brinda el tipo de letra más pequeño.

HTML agrega automáticamente un espacio antes y después de cada título. **<h1>** al ser usado como título de una página web, es de suma importancia, ya que es uno de los parámetros que Google y demás buscadores tienen en cuenta a la hora de indexar un sitio web.

```

pagina1.html* X Sitio sin nombre 30 - C:\Users\The\Desktop\Sesión11-Práctica\pagina1.html
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Documento sin título</title>
6  <meta charset="utf-8">
7  <meta name="Description" content="Cibertec">
8  </head>
9
10 <body>
11   <h1>Cibertec</h1>
12   <h2>Escuela de Tecnología</h2>
13   <h3>Computación e Informática</h3>
14   <h4>Base de Datos</h4>
15   <h5>Modelo Relacional</h5>
16   <h6>Formas Normales</h6>
17 </body>
18 </html>
19

```

Figura 57: Etiquetas de título
Fuente. - Elaboración propia

2.1.4.2 Párrafos <p>

La etiqueta párrafo se define con la etiqueta **<p>**. Esta etiqueta permite definir los párrafos que forma el texto de una página. El texto del párrafo lo puedes alinear con la etiqueta **align** utilizando los parámetros “center”, “right”, “left” y “justify”.

```

pagina1.html* X Sitio sin nombre 30 - C:\Users\The\Desktop\Sesión11-Práctica\pagina1.html
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Documento sin título</title>
6  <meta charset="utf-8">
7  <meta name="Description" content="Cibertec">
8  </head>
9
10 <body>
11   <h1>Cibertec</h1>
12   <h2>Escuela de Tecnología</h2>
13   <p align="center">Computación e Informática</p>
14   <p align="left">Administración y Sistemas</p>
15   <p align="right">Redes y Comunicaciones</p>
16 </body>
17 </html>
18

```

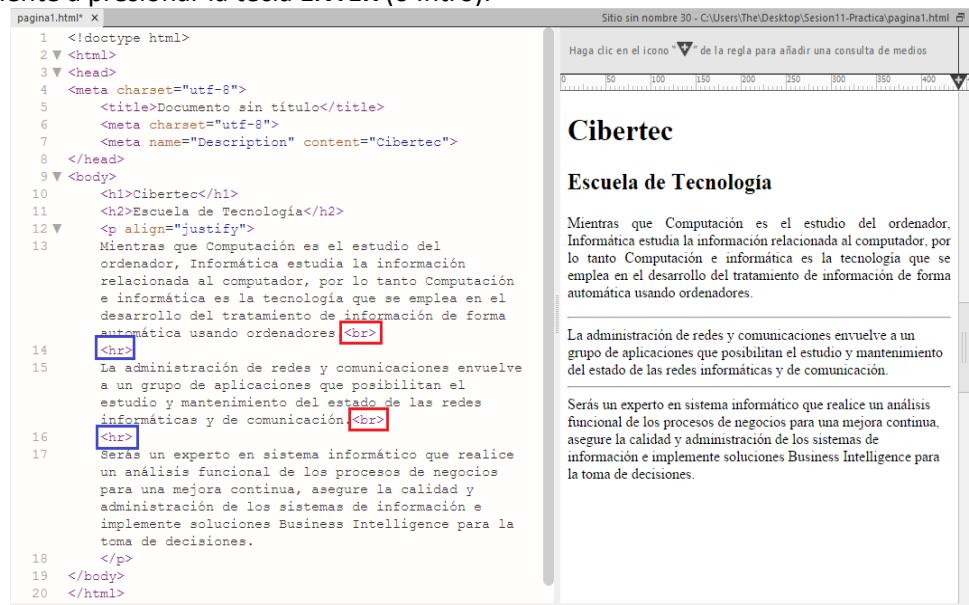
Figura 58: Etiqueta párrafo
Fuente. – Elaboración propia

2.1.4.3 Líneas horizontales <hr>

La etiqueta **<hr>** permite trazar una línea horizontal.

2.1.4.4 Saltos de línea

La etiqueta
 incluye una nueva línea y fuerza a que el texto se muestre en la siguiente línea. Es el equivalente a presionar la tecla **ENTER** (o Intro).



```

pagina1.html x
1  <!doctype html>
2  ▼ <html>
3  ▼ <head>
4    <meta charset="utf-8">
5      <title>Documento sin título</title>
6      <meta charset="utf-8">
7      <meta name="Description" content="Cibertec">
8  </head>
9  ▼ <body>
10   <h1>Cibertec</h1>
11   <h2>Escuela de Tecnología</h2>
12  ▼ <p align="justify">
13    Mientras que Computación es el estudio del ordenador, Informática estudia la información relacionada al computador, por lo tanto Computación e informática es la tecnología que se emplea en el desarrollo del tratamiento de información de forma automática usando ordenadores.<br>
14  <br>
15  La administración de redes y comunicaciones envuelve a un grupo de aplicaciones que posibilitan el estudio y mantenimiento del estado de las redes informáticas y de comunicación.<br>
16  <br>
17  Serás un experto en sistema informático que realice un análisis funcional de los procesos de negocios para una mejora continua, asegure la calidad y administración de los sistemas de información e implemente soluciones Business Intelligence para la toma de decisiones.
18  </p>
19  </body>
20  </html>

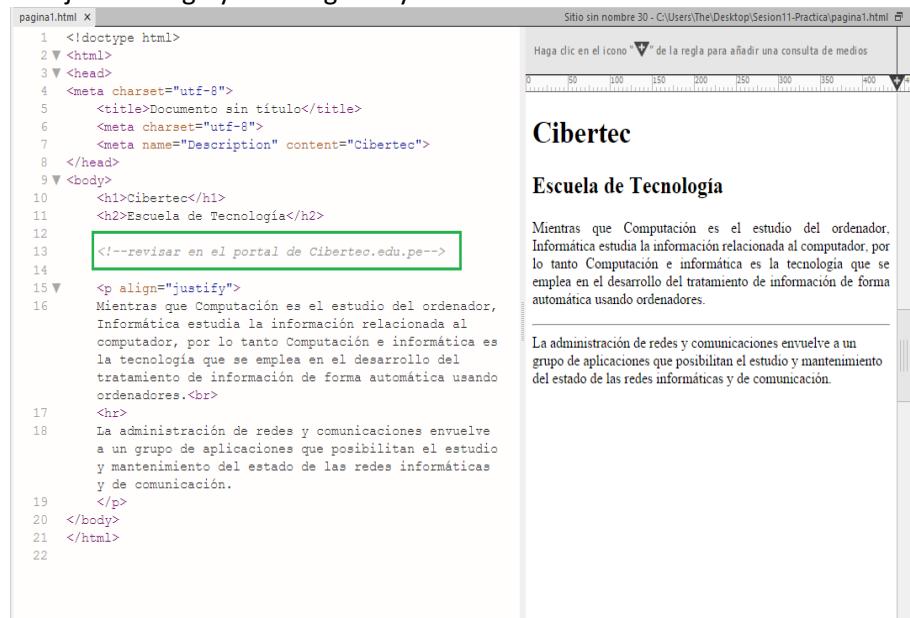
```

Figura 59: Etiqueta salto de línea

Fuente. - Elaboración propia

2.1.4.5 Comentarios <!-- comentario -->

La etiqueta <!-- ... --> se utiliza para insertar un comentario dentro del código que se está escribiendo, el cual es ignorado por el navegador al momento de leerlo. Los comentarios sirven para explicar mejor el código y son de gran ayuda en el momento de editarlo.



```

pagina1.html x
1  <!doctype html>
2  ▼ <html>
3  ▼ <head>
4    <meta charset="utf-8">
5      <title>Documento sin título</title>
6      <meta charset="utf-8">
7      <meta name="Description" content="Cibertec">
8  </head>
9  ▼ <body>
10   <h1>Cibertec</h1>
11   <h2>Escuela de Tecnología</h2>
12  <!--revisar en el portal de Cibertec.edu.pe-->
13
14  <p align="justify">
15    Mientras que Computación es el estudio del ordenador, Informática estudia la información relacionada al computador, por lo tanto Computación e informática es la tecnología que se emplea en el desarrollo del tratamiento de información de forma automática usando ordenadores.<br>
16  <br>
17  La administración de redes y comunicaciones envuelve a un grupo de aplicaciones que posibilitan el estudio y mantenimiento del estado de las redes informáticas y de comunicación.
18  </p>
19  </body>
20  </html>
21
22

```

Figura 60: Etiqueta comentario

Fuente. - Elaboración propia

2.1.4.6 Etiquetas de lista: listas ordenadas, desordenadas y de definición

Las listas permiten ordenar o enumerar datos o elementos para facilitar su lectura. Existen tres tipos de listas:

- Listas ordenadas.
- Listas no ordenadas.
- Listas de definición.

Listas ordenadas.

La etiqueta ``, permite definir listas o viñetas ordenadas (“Ordered List”), bien con numeración o alfabéticamente. Esta etiqueta es a su vez contenedora de otras etiquetas, siendo estas últimas las que componen cada uno de los elementos de los que está compuesta la lista.

Para especificar un elemento en la lista se ha de utilizar la etiqueta ` `, en la que se ha de especificar el texto que compone el elemento de la lista. Sintaxis:

```
<ol reversed start="valor" type="valor">
    <li>Elemento de lista</li>
    <li>Elemento de lista</li>
    <li> ... </li>

</ol>
```

type

Permite especificar el tipo de orden a aplicar en la lista, de esta forma la lista se representa mediante diferentes tipos de numeración.

Los diferentes valores que puede recibir este atributo son los siguientes:

Valor Descripción

- | | |
|---|---|
| 1 | Valor por defecto. Realiza la numeración de las lista (1,2,3...) |
| a | Establece que la numeración se realice minúscula: (a, b, c ...) |
| A | Establece que la numeración se realice mayúscula: (A, B, C ...) |
| i | Establece que la numeración se realice en romanos: (i, ii, iii ...) |
| I | Establece que la numeración se realice en romanos: (I, II, III ...) |

start

Indica el valor de inicio, por el cual ha de comenzar la lista ordenada, siendo este un valor numérico. En caso de ser la lista ordenada de forma alfabética, el valor de inicio que se exprese corresponderá con el orden el abecedario ASCII. Sintaxis:

```
<ol start="valor">
    <li>Elemento de lista</li>
    <li>Elemento de lista</li>
    <li> ... </li>
</ol>
```

reversed

Permite indicar que la numeración u orden que se haya establecido se represente de forma inversa, si se tuviera una lista con valor de inicio 1, 2, 3, sucesivamente, su representación al encontrarse este atributo sería 3,2,1.

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Documento sin título</title>
6  <meta charset="utf-8">
7  <meta name="Description" content="Cibertec">
8  </head>
9  <body>
10 <h1>Cibertec</h1>
11 <h2>Misión</h2>
12 <p align="justify">
13 Formar profesionales integros y competentes brindando
al desarrollo económico y ambiental del país.
</p>
14 <h2>Visión</h2>
15 <p align="justify">
16 Ser la institución líder de educación superior
técnica en el Perú con alcance a nivel nacional.
17 <h2>Carreras Técnicas</h2>
18 <ol type="I" start="1" >
19 <li>Ingeniería</li>
20 <li>Tecnología de Información</li>
21 <li>Comunicación</li>
22 <li>Gestión y Negocios</li>
23 <li>Diseño</li>
24 </ol>
25 </body>
26 </html>
27 |
28 |
29 |

```

Figura 61: Etiquetas de listas

Fuente. - Elaboración propia

Listas desordenadas

Una lista no ordenada es un conjunto de elementos relacionados entre sí, pero no se indica un orden o secuencia determinados. La etiqueta **** encierra todos los elementos de la lista y la etiqueta **** representa cada uno de sus elementos.

Para definir un determinado símbolo, por ejemplo: cuadrados, se define a través del atributo **type**. Los símbolos para las listas son: disc (puntos negros), circle (círculos), square (cuadrado). A continuación, se presenta un ejemplo utilizando listas no ordenadas.

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Documento sin título</title>
6  <meta charset="utf-8">
7  <meta name="Description" content="Cibertec">
8  </head>
9  <body>
10 <h1>Cibertec</h1>
11 <h2>Misión</h2>
12 <p align="justify">
13 Formar profesionales integros y competentes
brindando una educación superior de alta calidad que
contribuya al desarrollo económico y ambiental del
país.
14 </p>
15 <h2>Carreras Técnicas</h2>
16 <ol type="I" start="1" >
17 <li>Ingeniería</li>
18 <li>Tecnología de Información
19 <ul type="circle">
20 <li>Computación e Informática</li>
21 <li>Administración de Redes</li>
22 <li>Administración y Sistemas</li>
23 <li>Industrial y Sistemas</li>
24 </ul>
25 </li>
26 <li>Comunicación</li>
27 <li>Gestión y Negocios</li>
28 <li>Diseño</li>
29 </ol>
30 </body>
31 </html>

```

Figura 62:Etiquetas de listas no ordenadas

Fuente. - Elaboración propia

Listas de definición

Son listas que se utilizan en la mayoría de páginas HTML. Su funcionamiento es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones.

La etiqueta **<dl>** crea la lista de definición y las etiquetas **<dt>** y **<dd>** definen respectivamente el término (título de la definición) y la etiqueta **<dd>** representa la descripción de cada elemento de la lista.

Figura 63: Etiquetas de listas de definición

Fuente. - Elaboración propia

**2.1.4.7 Texto Enfatizado **

Se emplea para enfatizar un texto, habitualmente se representa en cursiva.

**2.1.4.8 Texto Reforzado **

Se emplea para destacar una parte del texto. Normalmente el texto incluido entre esas etiquetas se representa en negrita.

Figura 64: Etiquetas de texto

Fuente. - Elaboración propia

2.1.5 Etiqueta para hipervínculos <a> y atributo target

Los enlaces o *links* (en inglés) permite conectar con otros documentos:

- Una imagen
- Un video
- Un archivo de sonido
- Una página de un determinado sitio web
- Enviar un email

Los enlaces en HTML se crean mediante la etiqueta `<a>` (viene del inglés “anchor” traducido literalmente sería “ancla”).

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar que apunta a ese enlace. Las URL de los enlaces pueden ser: absolutas, relativas, internas y externas.

Un **enlace externo** apunta a páginas de otros sitios web; es decir, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Un **enlace interno** apunta a páginas del propio sitio web. Las **URL absolutas** incluyen todas las partes de la URL (protocolo, servidor y ruta), por lo que no se necesita más información para obtener el recurso enlazado. Las **URL relativas** se construyen a partir de la URL absoluta donde prescindimos de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado.

Para crear un enlace que apunte a la página principal de Cibertec solamente habría que incluir lo siguiente en un documento HTML:

```
<a href="http://www.cibertec.edu.pe">Cibertec</a>
```

Otro atributo de la etiqueta `<a>` es `name`, permite definir enlaces dentro de una misma página web. Es útil cuando la página es muy larga, donde se puede mostrar enlaces como: "Ir hasta la segunda sección", "Volver al principio de la página", etc.

```
<a name="ir_inicio"></a>
```

El atributo `type` se utiliza para notificar al navegador sobre el tipo de contenido que se enlaza. Se indica mediante una cadena de texto cuyos posibles valores también están estandarizados. Los valores de los contenidos más utilizados son los siguientes:

- "text/html" (páginas HTML),
- "image/png" (imágenes con formato PNG),
- "image/gif" (imágenes con formato GIF),
- "text/css" (hojas de estilo CSS),
- "application/rss+xml" (archivos RSS).

```
<a href="foto.jpg" type="image/jpg">Imagen Principal</a>
```

A través del valor `mailto`, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de `mailto`.

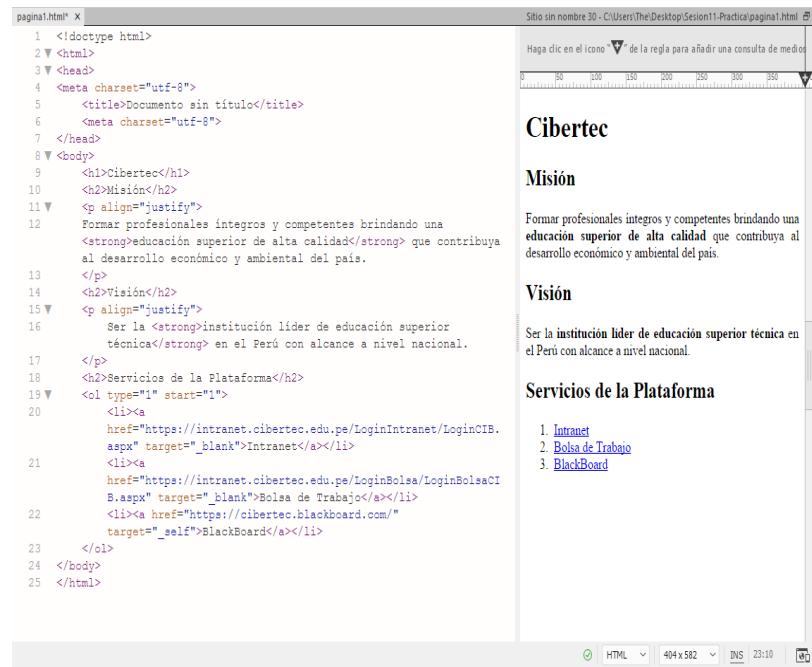
```
<a href="mailto:nombre@direccion.com" title="Email para solicitar más información">Solicita más información</a>
```

Aunque el uso de **mailto** puede parecer una ventaja, su uso está desaconsejado. Si se incluye el email directamente en una página web, en poco tiempo ese email estará lleno de correos electrónicos, basura o "spam", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

El atributo **target** se utiliza para definir dónde se desea que se abra el documento al presionar sobre el enlace, utiliza los siguientes valores:

- **_blank**: indica una nueva ventana.
- **_parent**: especifica el marco matriz que contiene el vínculo de origen.
- **_self**: el significado es muy evidente. El nuevo documento se carga en el mismo marco.
- **_top**: esto indica una nueva ventana de documento.

Ejemplo: utilizando etiquetas de hipervínculos.



```

pagina1.html* X
Síntesis sin nombre 30 - C:\Users\The\Desktop\Sesión11-Práctica\pagina1.html
Haga clic en el ícono "▼" de la regla para añadir una consulta de medios
  1 <!doctype html>
  2 <html>
  3 <head>
  4   <meta charset="utf-8">
  5   <title>Documento sin título</title>
  6   <meta charset="utf-8">
  7 </head>
  8 <body>
  9   <h1>Cibertec</h1>
 10  <h2>Misión</h2>
 11  <p align="justify">
 12    Formar profesionales integros y competentes brindando una
 13    <strong>educación superior de alta calidad</strong> que contribuya
 14    al desarrollo económico y ambiental del país.
 15  </p>
 16  <h2>Visión</h2>
 17  <p align="justify">
 18    Ser la <strong>institución líder de educación superior
 19    técnica</strong> en el Perú con alcance a nivel nacional.
 20  </p>
 21  <h2>Servicios de la Plataforma</h2>
 22  <ol type="1" start="1">
 23    <li><a href="https://intranet.cibertec.edu.pe/LoginIntranet/LoginCIB.aspx" target="_blank">Intranet</a></li>
 24    <li><a href="https://intranet.cibertec.edu.pe/LoginBolsa/LoginBolsaCI.aspx" target="_blank">Bolsa de Trabajo</a></li>
 25    <li><a href="https://cibertec.blackboard.com/" target="_self">BlackBoard</a></li>
 26  </ol>
 27 </body>
 28 </html>
  
```

Figura 65: Hipervínculos
Fuente. - Elaboración propia

2.1.6 Estilos en línea: uso del atributo **style**

Una de las formas más simples e intuitivas de dotar de estilos al código HTML es usando **estilos en línea** que admiten la mayoría de las etiquetas HTML.

Los estilos en línea son declaraciones CSS que se integran en las etiquetas HTML mediante el atributo **style**. Este método tan solo afecta al elemento en el que se integra el código.

Para aplicar un estilo en línea, utiliza la sintaxis:

<Etiqueta style = “propiedadCSS: valor;” > ... </Etiqueta>

2.1.6.1 Color de fondo: para asignar color de fondo se aplica la propiedad background-color, utilizando el valor hexadecimal (valor para todas las versiones del html).

2.1.6.2 Color de texto: para asignar color de texto a un elemento se aplica la propiedad color, utilizando el valor hexadecimal (valor aplicable a todas las versiones de html).

2.1.6.3 Alineación del párrafo: para alinear un elemento se aplica la propiedad text-align, cuyos valores son: left, right, justify, center.

2.1.6.4 Uso de fuente: tamaño, decoración, estilo. Para asignar el tamaño de la letra se aplica la propiedad font-size, dicha propiedad se aplica en pixeles como en expresión: large, larger, etc. Para decorar un elemento se aplica la propiedad text-decoration, cuyos valores es underline, none, tachado, etc. Y para dar estilo a los elementos aplica la propiedad font-style, cuyos valores es italic, cursive, etc.

2.1.7 Etiqueta de imagen y atributos src, alt, width, height

Las imágenes de contenido son las que proporcionan información y complementan la información textual. Las imágenes de adorno son las que se utilizan para hacer bordes redondeados, para mostrar pequeños íconos en las listas de elementos, fondos de página, etc.

Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta **** y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas.

Los dos atributos requeridos son **src** y **alt**. El atributo **src** representa la dirección URL de la imagen; la URL pueden ser absoluta o relativa. El atributo **alt** permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una longitud inferior a 1024 caracteres y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

```

```

Los atributos **width** y **height** se utilizan para indicar la anchura y altura con la que se muestran las imágenes. Como ya se ha comentado, HTML estructura de forma correcta los contenidos de la página y CSS define el aspecto gráfico con el que se muestran los contenidos.

```

```

El atributo **longdesc** proporciona una descripción larga de la imagen, este atributo también es muy importante para la accesibilidad de una página.

El atributo **border** permite asignar un border a la imagen, si su valor es cero la imagen no tendrá borde.

2.1.8 Etiquetas para figuras <figure> y <figcaption>

Las etiquetas <figure> y <figcaption> permiten definir una mejor semántica (significado o sentido) al contenido dentro de un documento HTML5.

Estos elementos en específico, permitirán anotar o resaltar ilustraciones, diagramas o fotos que son referenciados directamente desde el contenido principal. Estas etiquetas podrían ser utilizadas en diagramas, imágenes o fotos que acompañan el texto principal del contenido.

2.1.9 Uso de anclas: enlace a un lugar de la misma página

La sintaxis que se utiliza con estos enlaces es la misma que con los enlaces normales, salvo que se añade el **símbolo #** seguido del nombre de la sección a la que se apunta.

```
<a href="#ir_inicio">Ir al Inicio</a>
```

Los enlaces directos a secciones también funcionan con el **atributo id** de cualquier elemento.

```
<h1 id="inicio">Título de la página</h1>
<a href="#inicio">Volver al inicio de la página</a>
```

Para definir un enlace desde cualquier página web, se vuelve directamente a la página de inicio o página principal del sitio web.

```
<a href="/">Inicio</a>
```

LABORATORIO 1

Trabajando con Etiquetas

Diseña una página web, tal como se muestra, que permita publicar información acerca de “La Tecnología”.



Figura 66: Captura de pantalla
Fuente. - Elaboración propia

1. Creando el Sitio Web

Define una carpeta llamado **SitioWeb_01**, en ella crea las siguientes sub carpetas, tal como se muestra.

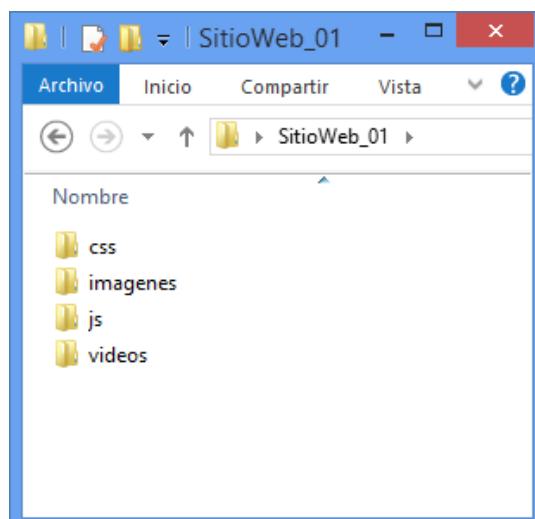


Figura 67: Captura de pantalla
Fuente. - Elaboración propia

En la carpeta **Imágenes**, agrega los archivos de extensión .jpg

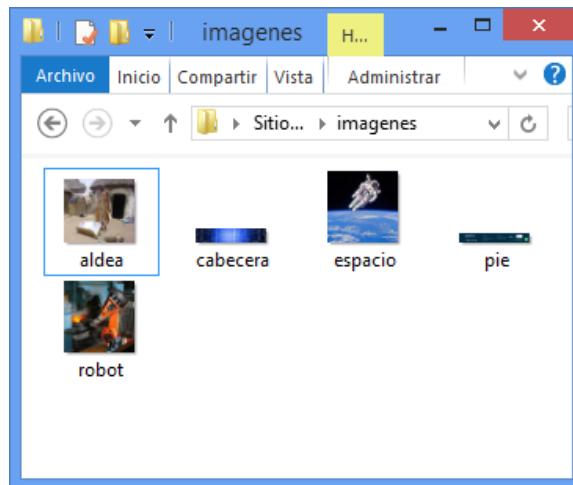


Figura 68: Captura de pantalla
Fuente. - Elaboración propia

2. Trabajando con Visual Studio Code

Para crear una página web, abre el aplicativo **Visual Studio Code**.

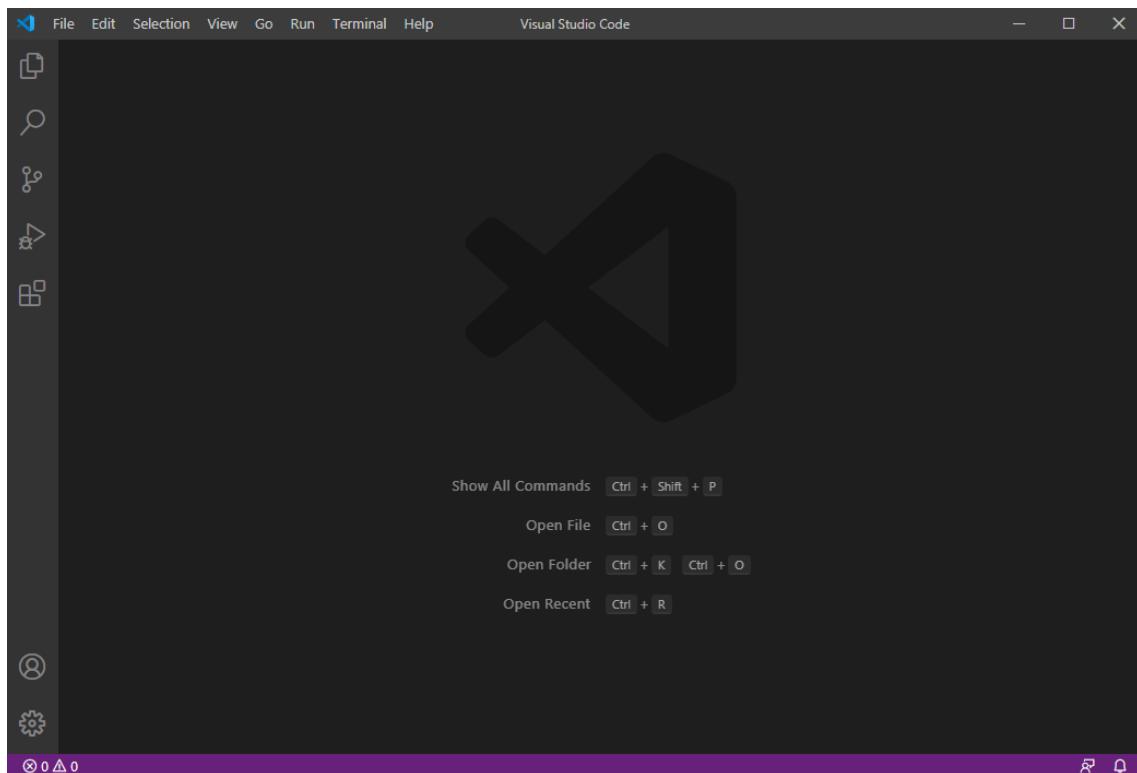


Figura 69: Captura de pantalla
Fuente. - Elaboración propia

Para trabajar con la carpeta llamada **SitioWeb_01**, arrastra la carpeta hacia la ventana del **Visual Studio Code**, tal como se muestra.

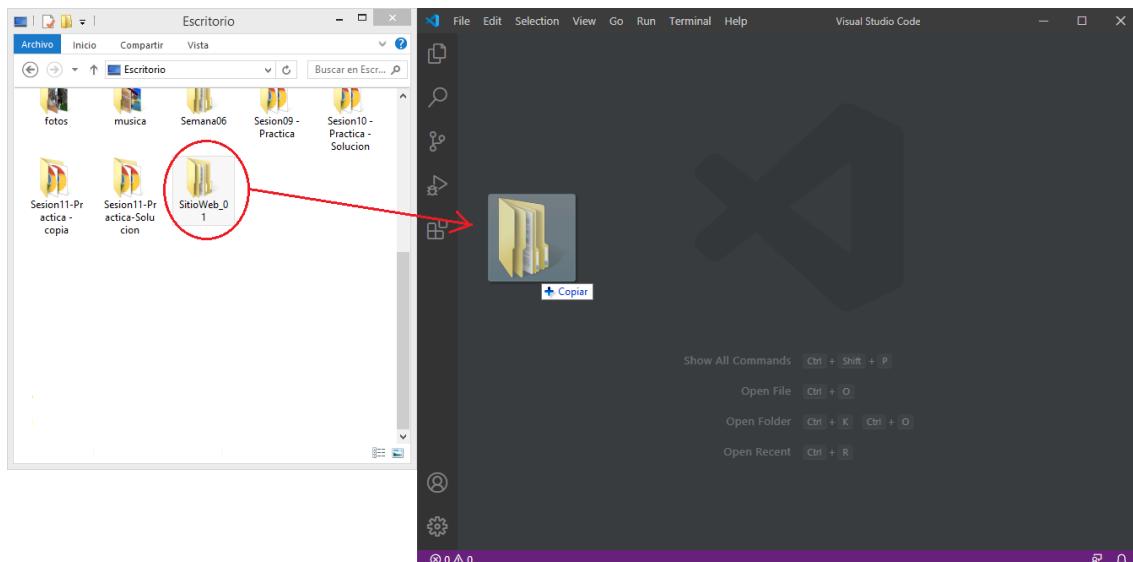


Figura 70: Captura de pantalla
Fuente. - Elaboración propia

Se visualiza la carpeta en la parte izquierda de la ventana del aplicativo.

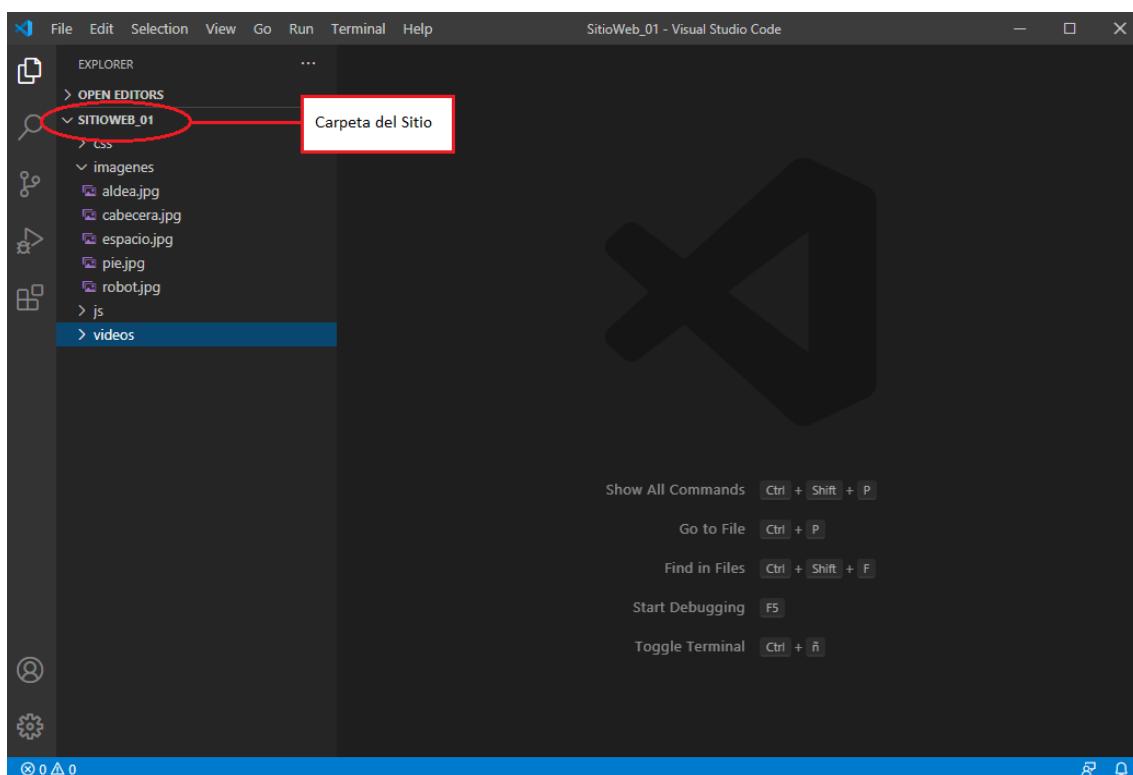


Figura 71: Captura de pantalla
Fuente. - Elaboración propia

3. Creando la página HTML

Para crear una página web, primero crea un nuevo archivo,. Desde la opción **File**, selecciona la opción **New File**, o simplemente utiliza la combinación de teclas **Ctrl+N**.

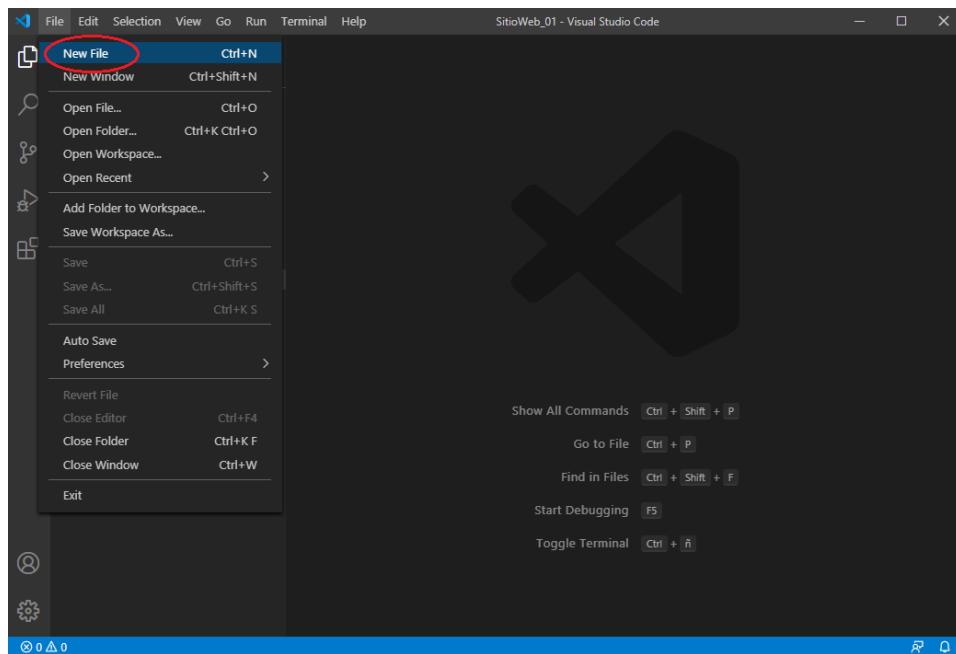


Figura 72: Captura de pantalla
Fuente. - Elaboración propia

Selecciona la opción, aparece un archivo con una etiqueta “**Untitled-1**”. A continuación, guarda el archivo para que sea de tipo html.

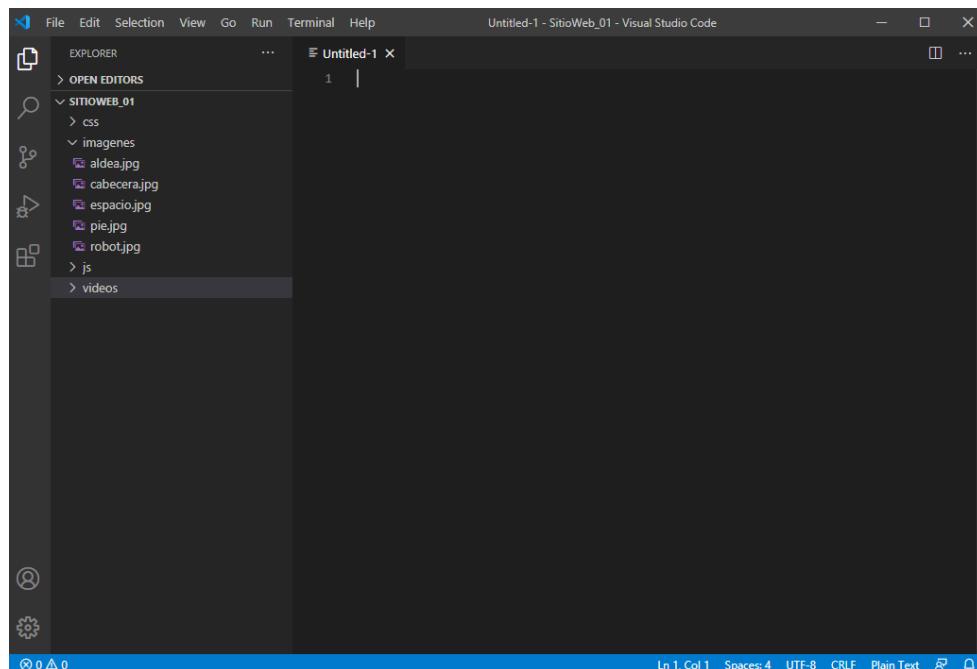


Figura 73: Captura de pantalla
Fuente. - Elaboración propia

Para guardar el archivo, haz clic en la opción **File**, selecciona la opción **Save**, o también presiona la combinación de teclas **Ctrl+S**.

En la ventana Save As, asigna el nombre del archivo: **pagina1**; y selecciona el tipo de archivo el cual será HTML, tal como se muestra. Presiona el botón **GUARDAR**.

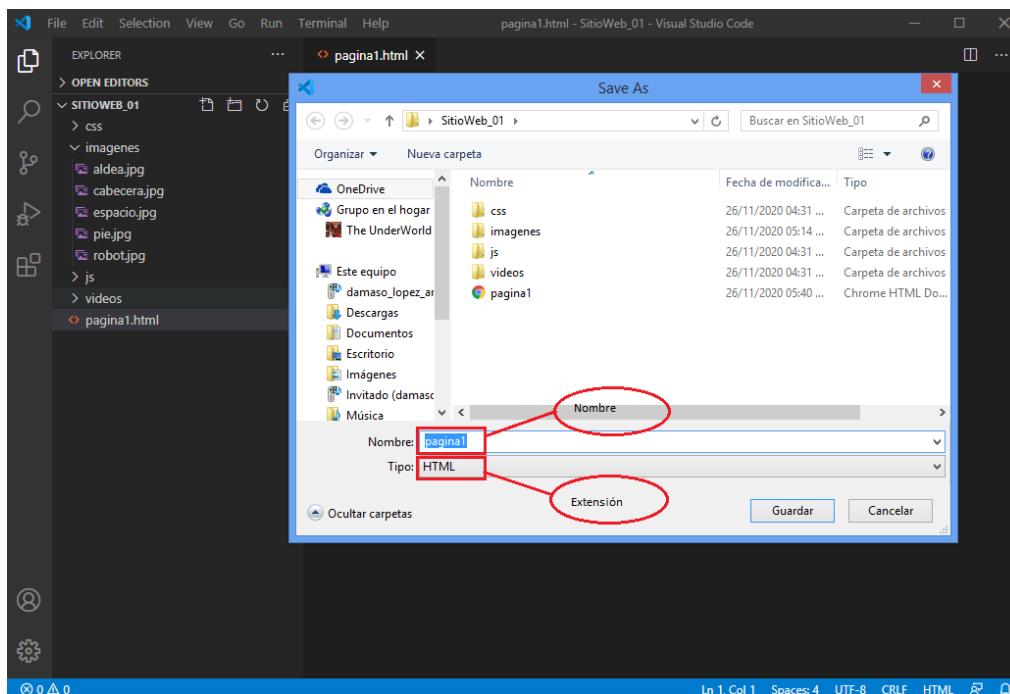


Figura 74: Captura de pantalla
Fuente. - Elaboración propia

En la página, ya creada, agrega la estructura HTML. Para ello, escribe en la primera línea **html**, y selecciona la lista **html:5**, y presiona **ENTER**.

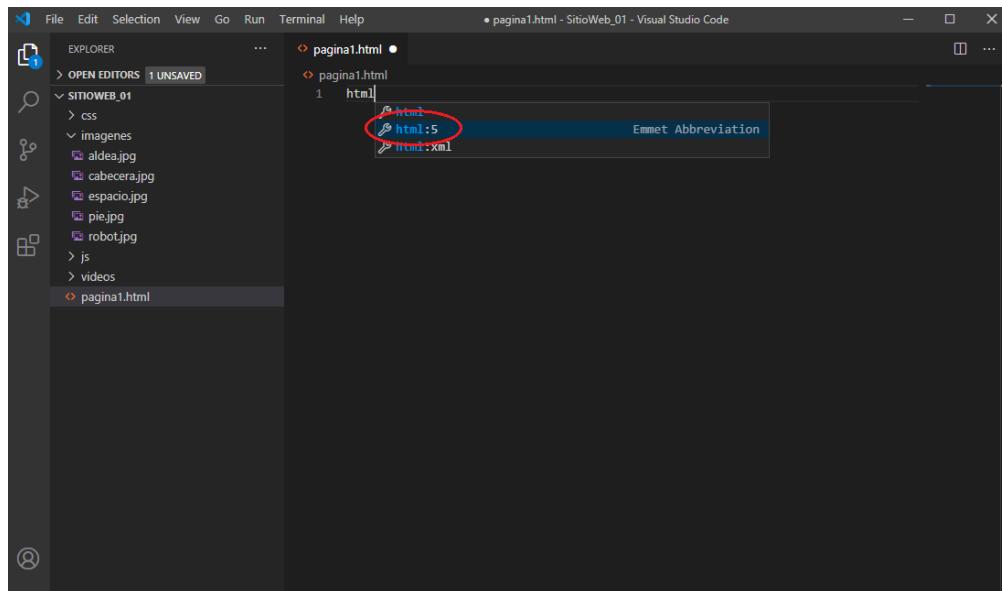
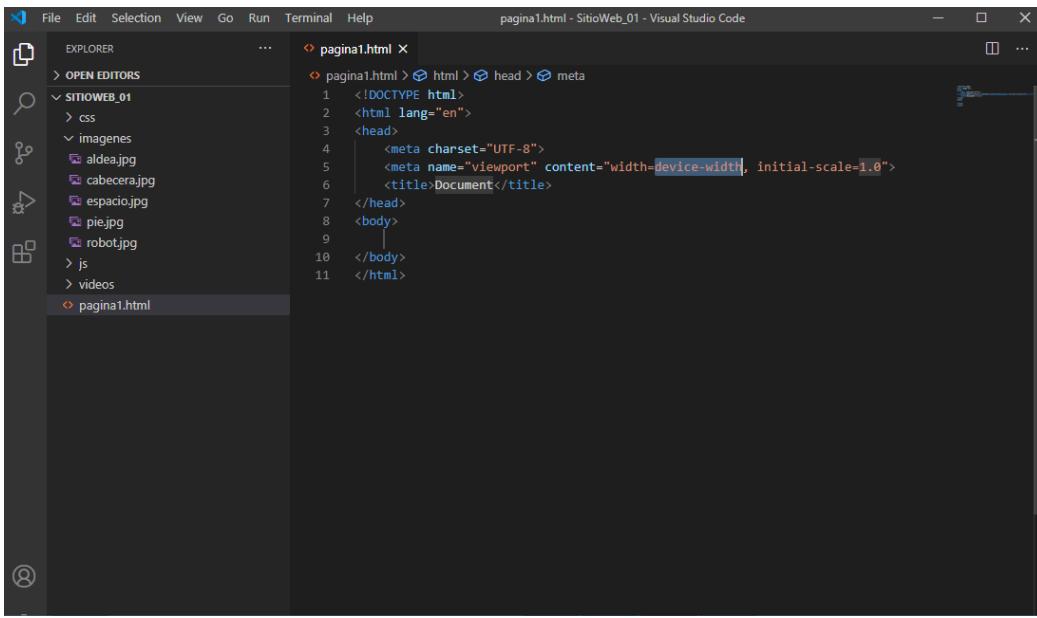


Figura 74: Captura de pantalla
Fuente. - Elaboración propia

Al presionar la tecla **ENTER**, se visualiza la estructura del html versión 5, tal como se muestra. A partir de aquí se puede diseñar la página web.



The screenshot shows the Visual Studio Code interface. The left sidebar (Explorer) lists files in the 'SIMOWEB_01' folder: aldea.jpg, cabecera.jpg, espacio.jpg, pie.jpg, robot.jpg, js, videos, and pagina1.html. The right pane (Code Editor) displays the HTML code for 'pagina1.html':

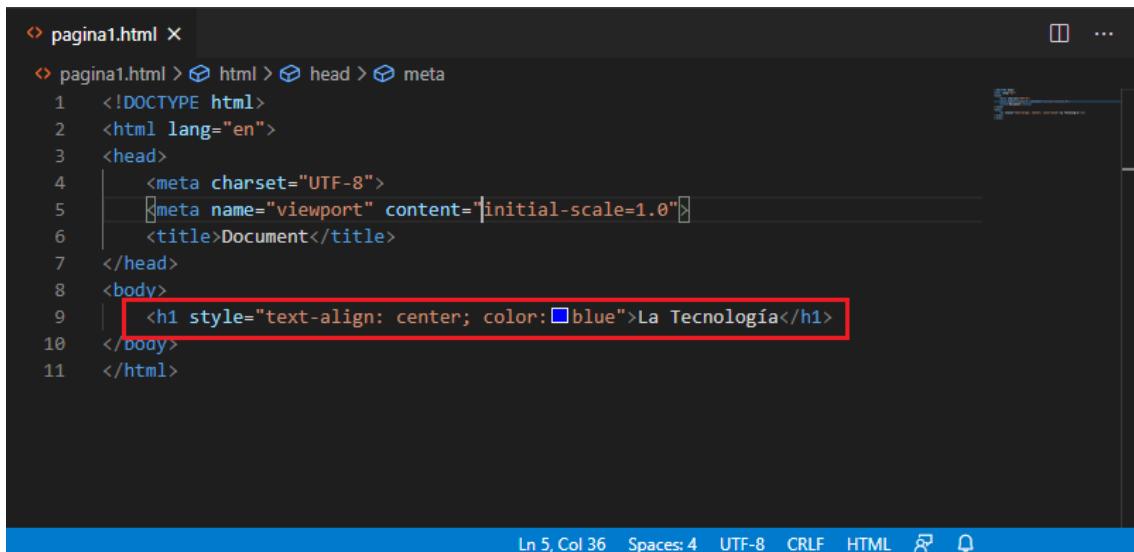
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
  </body>
</html>
```

Figura 75: Captura de pantalla
Fuente. - Elaboración propia

4. Diseño de la página HTML

Añadiendo etiquetas header

A continuación, agrega etiquetas head **<h1>** tal como se muestra, asigna estilo: color de letra y alineamiento.



The screenshot shows the Visual Studio Code interface with the same file structure and code editor as Figure 75. The code editor now includes an additional **<h1>** tag in the body:

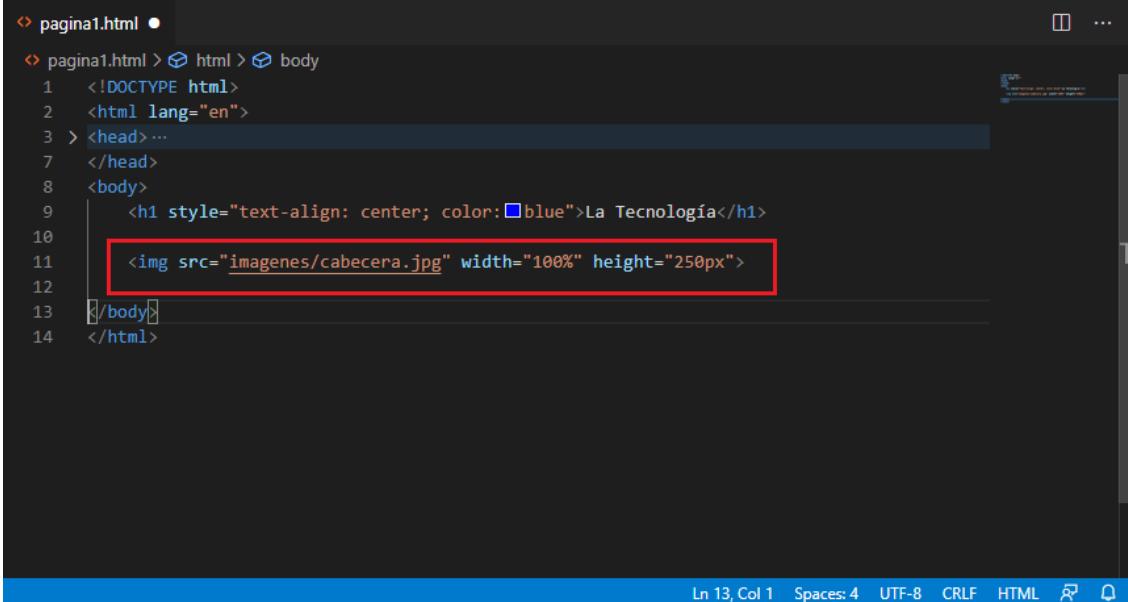
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <h1 style="text-align: center; color: blue">La Tecnología</h1>
  </body>
</html>
```

The line containing the **<h1>** tag is highlighted with a red box.

Figura 76: Captura de pantalla
Fuente. - Elaboración propia

Agregando la imagen de la escuela

Para incrustar una imagen en la página, define la etiqueta **** indicando el origen de la imagen: **src**, y los atributos **width** (ancho expresado en porcentaje) y **height** (altura), tal como se muestra.



```

1 <!DOCTYPE html>
2 <html lang="en">
3 > <head>...
4 </head>
5 <body>
6   <h1 style="text-align: center; color: blue">La Tecnología</h1>
7   
8 </body>
9 </html>

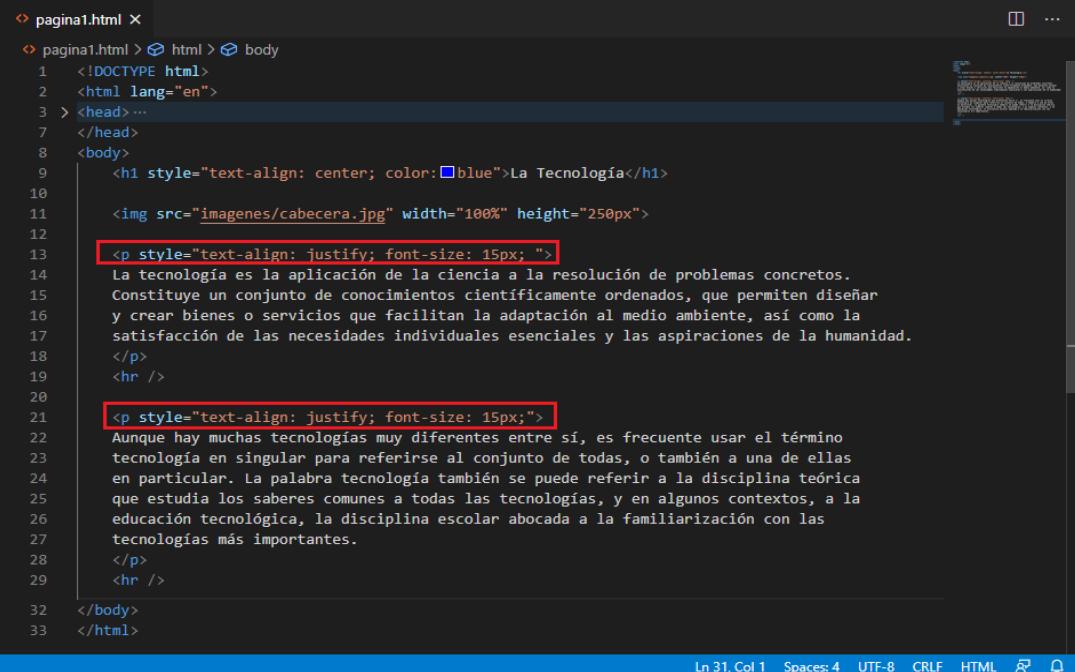
```

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍

Figura 77: Captura de pantalla
Fuente. - Elaboración propia

Agregando texto a la página

A continuación, define la etiqueta **<p>** para agregar los párrafos. Asigna estilos a los párrafos: alineamiento (**text-align**) y tamaño (**font-size**), tal como se muestra.



```

1 <!DOCTYPE html>
2 <html lang="en">
3 > <head>...
4 </head>
5 <body>
6   <h1 style="text-align: center; color: blue">La Tecnología</h1>
7   
8   <p style="text-align: justify; font-size: 15px;">
9     La tecnología es la aplicación de la ciencia a la resolución de problemas concretos.
10    Constituye un conjunto de conocimientos científicamente ordenados, que permiten diseñar
11    y crear bienes o servicios que facilitan la adaptación al medio ambiente, así como la
12    satisfacción de las necesidades individuales esenciales y las aspiraciones de la humanidad.
13  </p>
14  <hr />
15
16  <p style="text-align: justify; font-size: 15px;">
17    Aunque hay muchas tecnologías muy diferentes entre sí, es frecuente usar el término
18    tecnología en singular para referirse al conjunto de todas, o también a una de ellas
19    en particular. La palabra tecnología también se puede referir a la disciplina teórica
20    que estudia los saberes comunes a todas las tecnologías, y en algunos contextos, a la
21    educación tecnológica, la disciplina escolar abocada a la familiarización con las
22    tecnologías más importantes.
23  </p>
24  <hr />
25
26 </body>
27 </html>

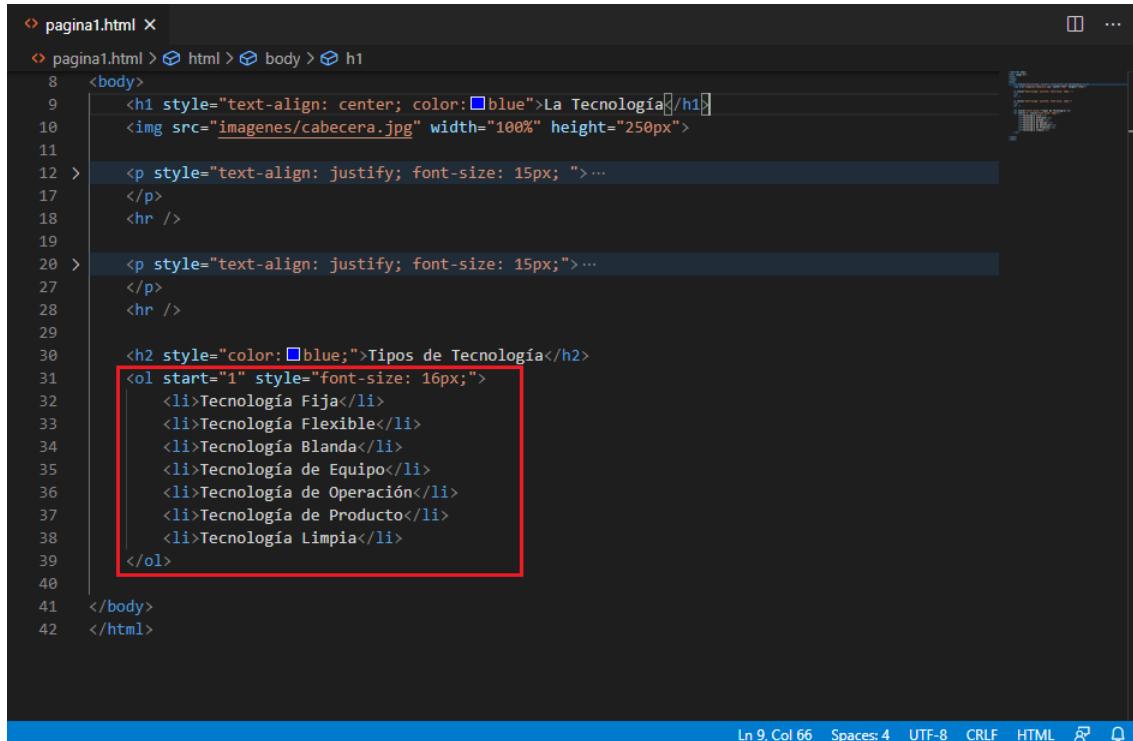
```

Ln 31, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍

Figura 78: Captura de pantalla
Fuente. - Elaboración propia

Agregando listas

Define una lista ordenada ``, el cual inicia (`start=1`) y cuyo tamaño de letra de sus elementos (`font-size`) es 16px; agrega sus elementos ``, para mencionar los tipos de tecnología.



```

<!-- pagina1.html -->
<!-- pagina1.html > html > body > h1
8   <body>
9     <h1 style="text-align: center; color: blue">La Tecnología</h1>
10    
11
12  >   <p style="text-align: justify; font-size: 15px;"> ...
13    </p>
14    <hr />
15
16  >   <p style="text-align: justify; font-size: 15px;"> ...
17    </p>
18    <hr />
19
20  >   <h2 style="color: blue;">Tipos de Tecnología</h2>
21   <ol start="1" style="font-size: 16px;">
22     <li>Tecnología Fija</li>
23     <li>Tecnología Flexible</li>
24     <li>Tecnología Blanda</li>
25     <li>Tecnología de Equipo</li>
26     <li>Tecnología de Operación</li>
27     <li>Tecnología de Producto</li>
28     <li>Tecnología Limpia</li>
29   </ol>
30
31  </body>
32  </html>

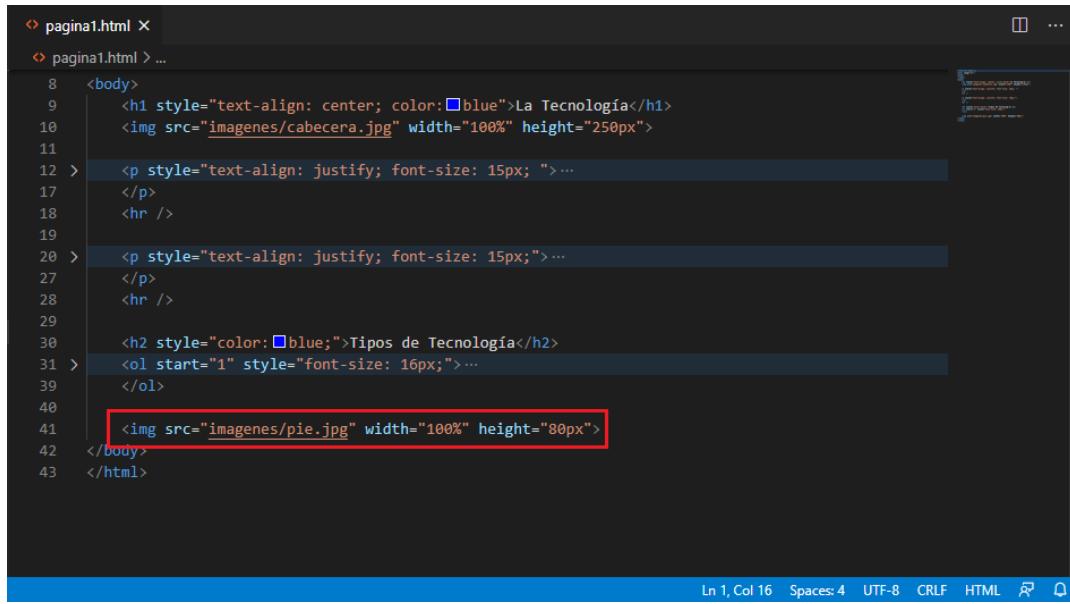
```

The code editor shows an HTML file named 'pagina1.html'. It contains a header section with a centered title 'La Tecnología' and a background image. Below this is a paragraph, a horizontal line, another paragraph, and a horizontal line. A section titled 'Tipos de Tecnología' follows, enclosed in a red box. This section contains an ordered list with the start value set to 1, and each item has a font size of 16px. The list items are: 'Tecnología Fija', 'Tecnología Flexible', 'Tecnología Blanda', 'Tecnología de Equipo', 'Tecnología de Operación', 'Tecnología de Producto', and 'Tecnología Limpia'. The status bar at the bottom of the editor indicates: Ln 9, Col 66, Spaces: 4, UTF-8, CRLF, HTML, and icons for save and search.

Figura 79: Captura de pantalla
Fuente. - Elaboración propia

Agregando la imagen del pie de página

Para incrustar una imagen en el pie de la página, define la etiqueta `` indicando el origen de la imagen: `src`, así como su `width` (ancho al 100%) y su `height` (altura a 80px).



```

<!-- pagina1.html -->
<!-- pagina1.html > ...
8   <body>
9     <h1 style="text-align: center; color: blue">La Tecnología</h1>
10    
11
12  >   <p style="text-align: justify; font-size: 15px;"> ...
13    </p>
14    <hr />
15
16  >   <p style="text-align: justify; font-size: 15px;"> ...
17    </p>
18    <hr />
19
20  >   <h2 style="color: blue;">Tipos de Tecnología</h2>
21   <ol start="1" style="font-size: 16px;"> ...
22     </ol>
23
24  >   
25
26  </body>
27  </html>

```

The code editor shows the same 'pagina1.html' file as Figure 79. The 'Tipos de Tecnología' section is still present. At the bottom of the page, there is a new image element defined in line 24: ``. This image is also enclosed in a red box. The status bar at the bottom of the editor indicates: Ln 1, Col 16, Spaces: 4, UTF-8, CRLF, HTML, and icons for save and search.

Figura 80: Captura de pantalla
Fuente. - Elaboración propia

Para ejecutar la página, presiona la tecla **F5** y selecciona desde **Select Environment, Chrome (preview)**, tal como se muestra.

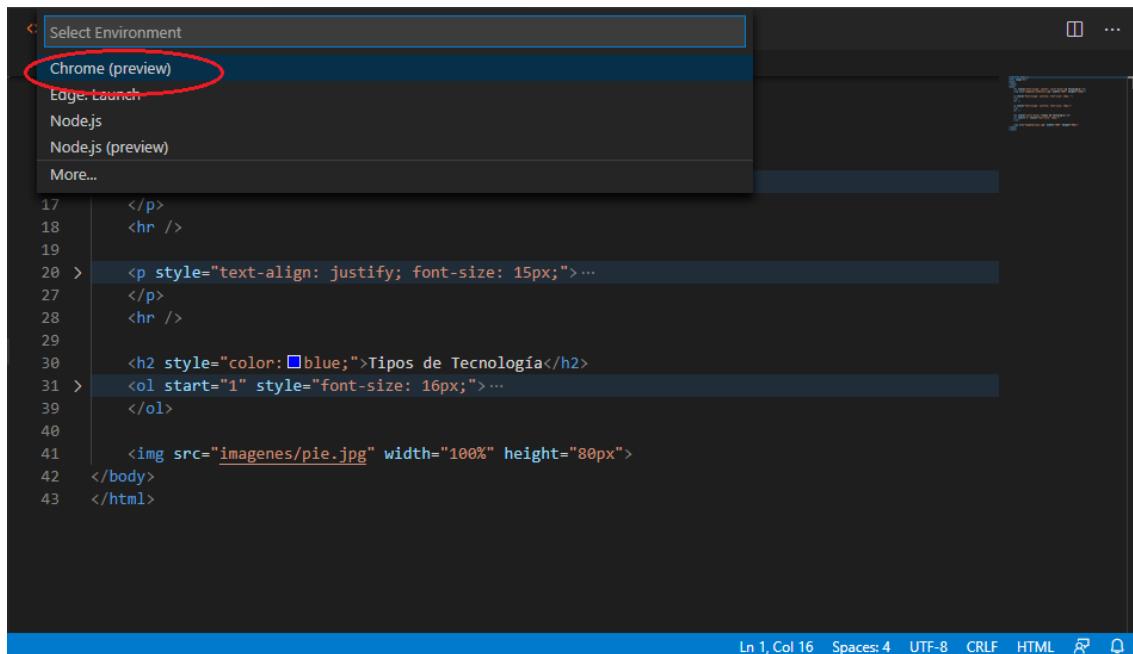


Figura 81: Captura de pantalla
Fuente. - Elaboración propia

Finalmente, visualiza la página en un navegador seleccionado.



Figura 82: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 2

Trabajando con Etiquetas

Diseña una página web que permita hablar acerca del Medio Ambiente y la Tecnología, tal como se muestra en la figura.



Figura 83: Captura de pantalla
Fuente. - Elaboración propia

1. Agregando una página HTML al sitio

Crea un nuevo documento y presiona la combinación de teclas **Ctrl+N**. Para guardar el archivo con extensión html presiona la tecla **Ctrl+S**. Asigna el nombre y selecciona la extensión (**html**), y presiona el botón **Guardar**.

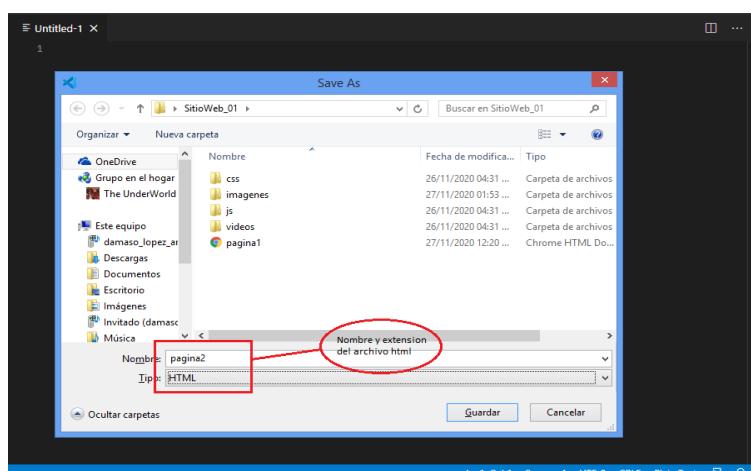
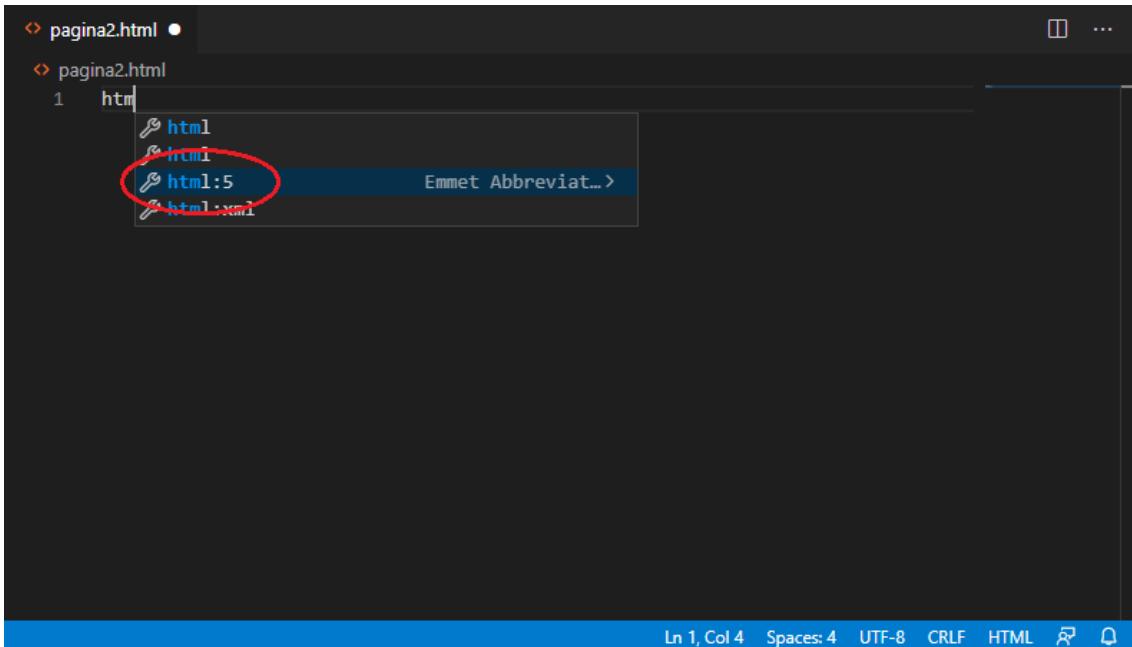


Figura 84: Captura de pantalla
Fuente. - Elaboración propia

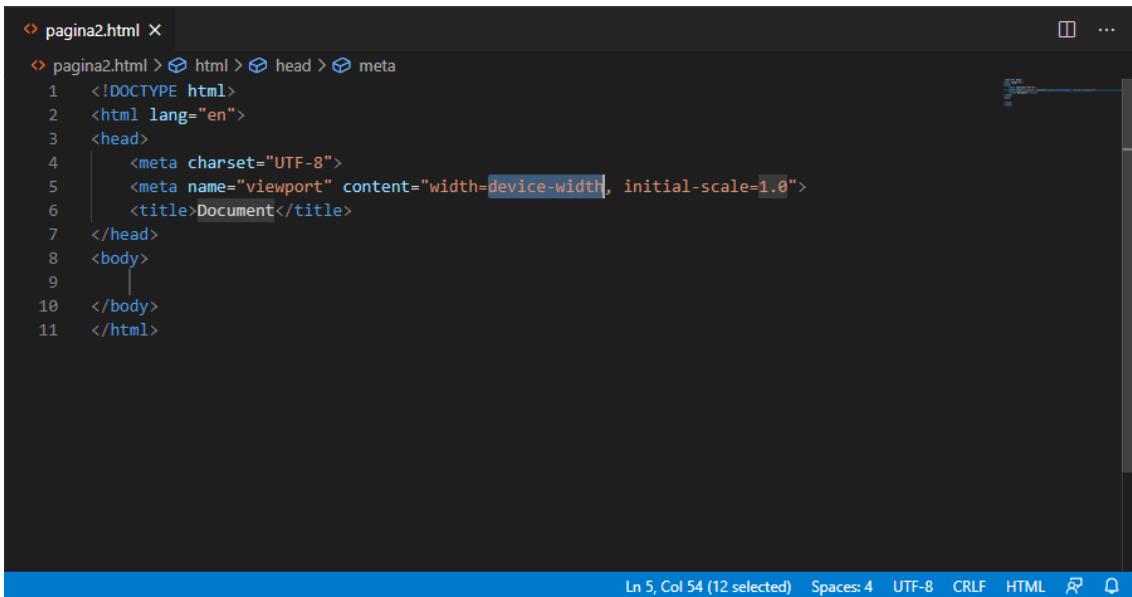
En la página2.html, escribe el texto html, donde se visualiza las definiciones del mismo. Luego, selecciona la opción **html:5** para visualizar la estructura de la página.



The screenshot shows a code editor interface with a dark theme. In the top left, there's a tree view with two items: 'pagina2.html' (selected) and 'pagina2.html'. Below this, a code editor window contains the text '1 htm|'. A dropdown menu titled 'Emmet Abbreviat...>' is open, showing several options: 'html', 'html5' (which has a red circle around it), 'html:5', and 'html:xml'. At the bottom of the editor, there's a status bar with the text 'Ln 1, Col 4 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍'.

Figura 85: Captura de pantalla
Fuente. - Elaboración propia

A continuación, se visualiza la estructura de la página html, tal como se muestra.



The screenshot shows a code editor with a dark theme. The top left displays a tree view with 'pagina2.html' selected. The main editor area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9 
10 </body>
11 </html>
```

At the bottom of the editor, there's a status bar with the text 'Ln 5, Col 54 (12 selected) Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍'.

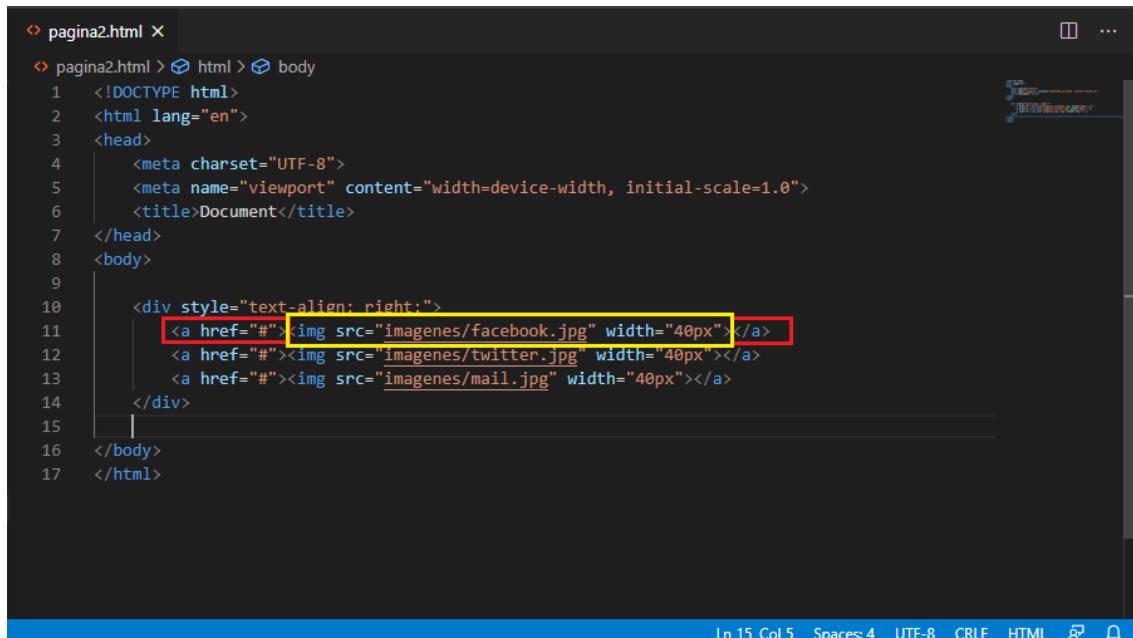
Figura 86: Captura de pantalla
Fuente. - Elaboración propia

2. Diseño de la página HTML

Agrega enlaces o hipervínculos

Para mostrar los enlaces a las redes sociales a través de imágenes:

- Define un elemento `<div>` donde el alineamiento de sus elementos será derecho
- Define la referencia a la página ``
- Dentro de cada etiqueta `<a>`, agrega la etiqueta `` define su origen y width.
- Repite el proceso para twitter y mail.



```

    <div style="text-align: right;">
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
    </div>

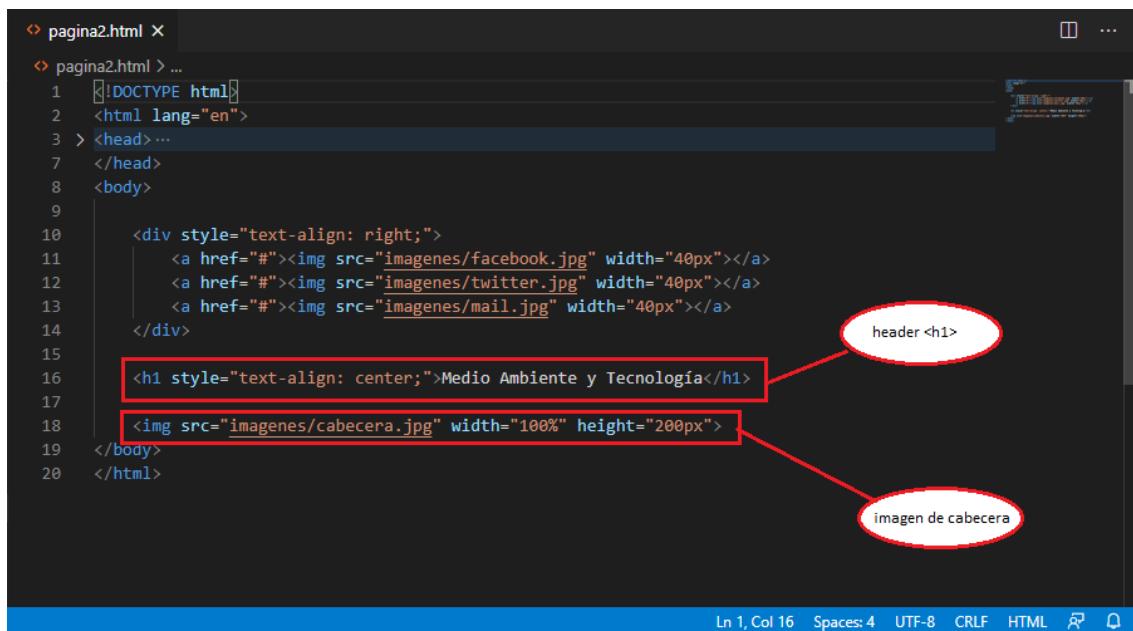
```

Figura 87: Captura de pantalla

Fuente. - Elaboración propia

Agrega header e imagen

A continuación, define una etiqueta headering `<h1>`; luego agrega la etiqueta ``, la cual representa la cabecera de la página.



```

    <div style="text-align: right;">
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
    </div>

    <h1 style="text-align: center;">Medio Ambiente y Tecnología</h1>
    

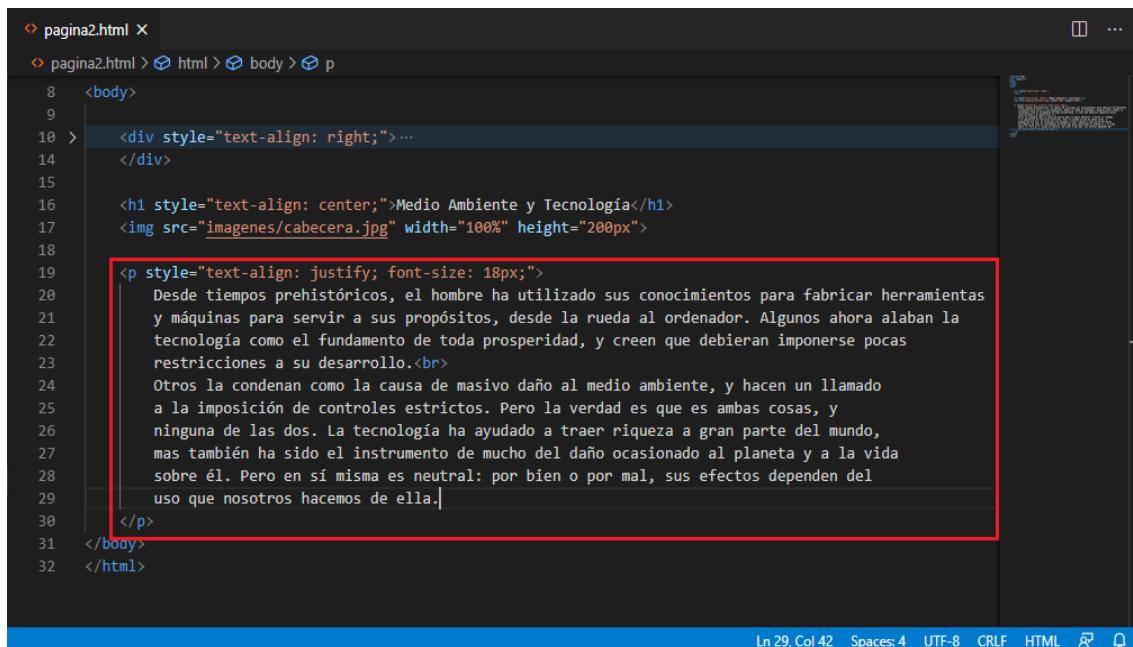
```

Figura 88: Captura de pantalla

Fuente. - Elaboración propia

Agregando texto a la página

Define la etiqueta `<p>` y el estilo: alineamiento justificado y tamaño de letra. Digita el texto siguiente en la página, tal como se muestra.



```

<div style="text-align: right;"></div>
<h1 style="text-align: center;">Medio Ambiente y Tecnología</h1>

<p style="text-align: justify; font-size: 18px;">
    Desde tiempos prehistóricos, el hombre ha utilizado sus conocimientos para fabricar herramientas
    y máquinas para servir a sus propósitos, desde la rueda al ordenador. Algunos ahora alaban la
    tecnología como el fundamento de toda prosperidad, y creen que debieran imponerse pocas
    restricciones a su desarrollo.<br>
    Otros la condenan como la causa de masivo daño al medio ambiente, y hacen un llamado
    a la imposición de controles estrictos. Pero la verdad es que es ambas cosas, y
    ninguna de las dos. La tecnología ha ayudado a traer riqueza a gran parte del mundo,
    mas también ha sido el instrumento de mucho del daño ocasionado al planeta y a la vida
    sobre él. Pero en sí misma es neutral: por bien o por mal, sus efectos dependen del
    uso que nosotros hacemos de ella.</p>
</body>
</html>

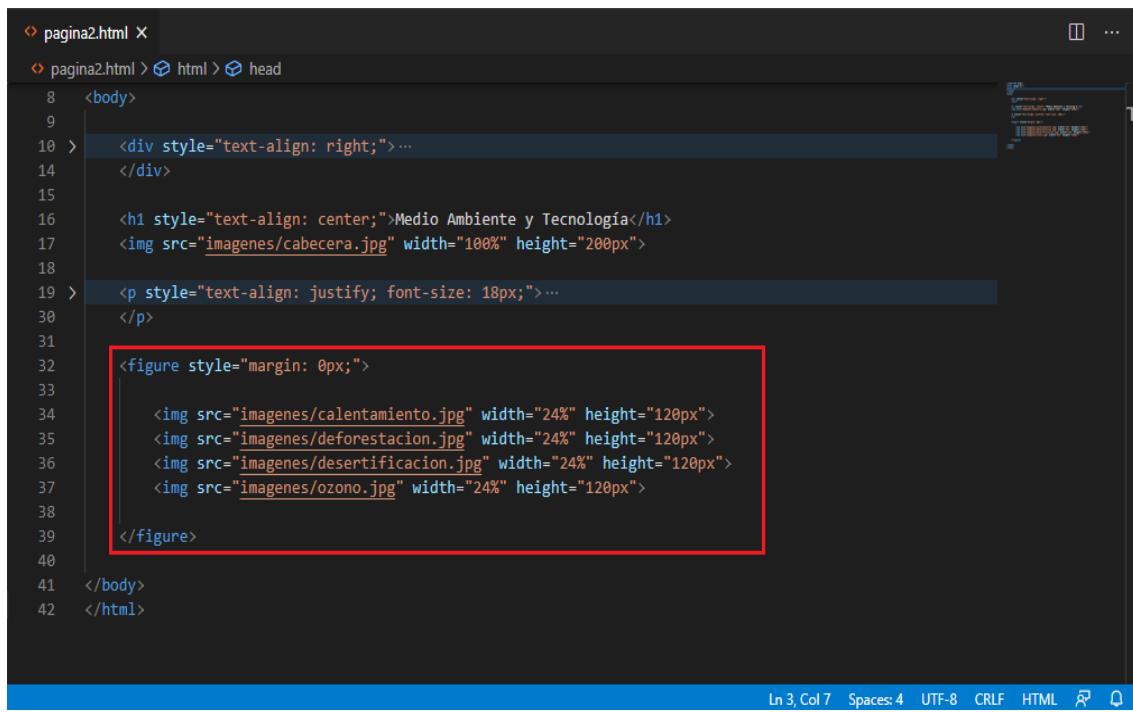
```

Figura 89: Captura de pantalla

Fuente. - Elaboración propia

Agregando imágenes

Para agregar un grupo de imágenes a la página, define el elemento `<figure>` para agrupar 4 imágenes, asigna el estilo `margin: "0px"`, para que no tenga márgenes. Luego, define 4 elementos `` cada uno con su respectivo origen (`src`) y sus atributos `width` y `height`. Dichos atributos deben tener los mismos valores.



```

<div style="text-align: right;"></div>
<h1 style="text-align: center;">Medio Ambiente y Tecnología</h1>

<p style="text-align: justify; font-size: 18px;">
<figure style="margin: 0px;">
    
    
    
    
</figure>
</p>
</body>
</html>

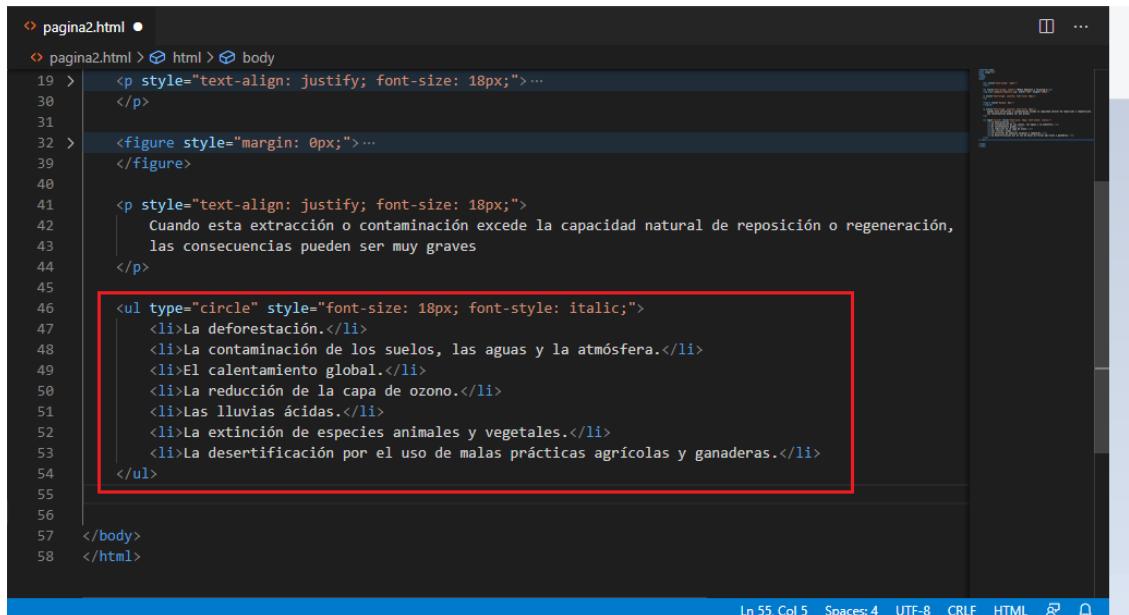
```

Figura 90: Captura de pantalla

Fuente. - Elaboración propia

Agregando una lista no ordenada

Define una lista no ordenada `` cuyo elemento de sangría es el círculo (`type="circle"`), definiendo como estilo: tamaño de letra y estilo: ***italic***. Agrega elementos `` para mencionar las consecuencias de la tecnología.



```

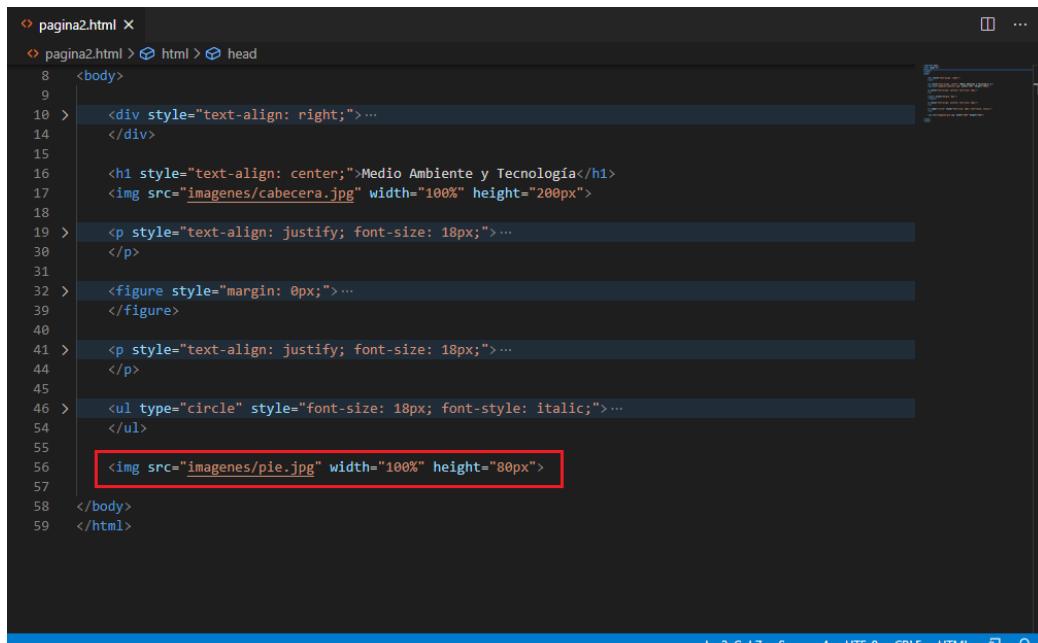
<div data-bbox="138 164 849 438">
    <ul type="circle" style="font-size: 18px; font-style: italic;">
        <li>La deforestación.</li>
        <li>La contaminación de los suelos, las aguas y la atmósfera.</li>
        <li>El calentamiento global.</li>
        <li>La reducción de la capa de ozono.</li>
        <li>Las lluvias ácidas.</li>
        <li>La extinción de especies animales y vegetales.</li>
        <li>La desertificación por el uso de malas prácticas agrícolas y ganaderas.</li>
    </ul>

```

Figura 91: Captura de pantalla
Fuente. - Elaboración propia

Agregando la imagen del pie de página

Finalizando el diseño de la página, agrega la etiqueta `` indicando el origen de la imagen: `src`, definiendo el ancho (width) y alto de la imagen (height) tal como se muestra.



```

<div data-bbox="138 544 792 830">
    

```

Figura 92: Captura de pantalla
Fuente. - Elaboración propia

Para visualizar la página presiona la tecla **F5**, selecciona la opción **Chrome (preview)** ejecutando la página.



Figura 93: Captura de pantalla
Fuente. - Elaboración propia

Resumen

1. Ante las necesidades de los usuarios y productores de contenido se ha lanzado la versión HTML5. Esta nueva versión del lenguaje Web simplifica el trabajo y agrupa las nuevas tecnologías de desarrollo de aplicaciones web: CSS3 y Javascript.
2. **Dreamweaver** es el mejor editor para crear páginas web, permite crear avanzadas web que se pueden visualizar también en dispositivos móviles, sin tener que dominar el código html, combina las últimas tecnologías en desarrollo web como Javascript, CSS, AJAX, XHTM, Adobe AIR, Smart Objects de Photoshop, subversiones (SVN), frameworks Javascript y más.
3. Un sitio Web ofrecerá tanto a clientes como a proveedores y empleados, un canal de comunicación abierto las 24 hrs. del día con información completa y actualizada sobre la organización, para que realicen consultas, pedidos o sugerencias.
4. Un documento HTML no es más que un archivo de texto. La extensión de su nombre suele ser .html o .htm. Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página; por ejemplo, su título e idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto, imágenes, videos, audio, etc.
5. La estructura de un documento en HTML5 inicia con la declaración del tipo de documento que se va a utilizar, esto se realiza definiendo la etiqueta **doctype**. Para construir la estructura del documento HTML define las etiquetas **<html></html>**. La palabra head viene del inglés “cabeza” y su función es delimitar cabecera del documento. Sus etiquetas son: **<head> y </head>**. La etiqueta **<body>** delimita el cuerpo del documento HTML. Este elemento contiene todo el contenido visible de un documento HTML (párrafos de texto, imágenes, tablas).
6. HTML agrega automáticamente un espacio antes y después de cada título. Es de suma importancia cuando **<h1>** es usado como título de una página web, ya que es uno de los parámetros que Google y demás buscadores tienen en cuenta a la hora de indexar un sitio web.
7. La etiqueta párrafo se define con la etiqueta **<p>**. Esta etiqueta permite definir los párrafos que forma el texto de una página. El texto del párrafo se alinea utilizando la etiqueta **align** y empleando los parámetros “center”, “right”, “left” y “justify”.
8. La etiqueta **
** incluye una nueva línea y fuerza a que el texto se muestre en la siguiente línea. Es el equivalente a presionar la tecla ENTER (o Intro).
9. Los enlaces en HTML se crean mediante la etiqueta **<a>** (viene del inglés “anchor” traducido literalmente sería “ancla”). El atributo más importante de la etiqueta **<a>** es **href**, que se utiliza para indicar que apunta a ese enlace. Las URL de los enlaces pueden ser: absolutas, relativas, internas y externas.

10. Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas. Los dos atributos requeridos son src y alt. El atributo src representa la dirección URL de la imagen; la URL pueden ser absoluta o relativa. El atributo alt permite describir el contenido de la imagen mediante un texto breve.
11. Las etiquetas <figure> y <figcaption> permiten definir una mejor semántica (significado o sentido) al contenido dentro de un documento HTML5.
12. Las listas permiten ordenar o enumerar datos o elementos para facilitar su lectura. Existen tres tipos de listas:
 - Ordenadas
 - Desordenadas
 - De definición
13. Las tablas son herramientas muy útiles para presentar datos en una estructura de filas y columnas, para el diseño de páginas y ubicación de textos y gráficos dentro de las mismas. Una tabla en html viene marcada por las etiquetas <table></table>. Entre esas dos etiquetas se define las celdas, columnas y características de cada uno de estos parámetros.

Recursos

Pueden revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- <http://www.virtualnauta.com/html-formato-de-texto>
- http://librosweb.es/xhtml/capitulo_3/marcado_avanzado_de_texto.html
- <http://html.hazunaweb.com/103.php>
- https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos

2.2. ESTILOS CSS

2.2.1 Selector de estilo CSS

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, las cuales están escritas con código HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.



Figura 94: Html y CSS3

Fuente. – Tomado de: <https://rolandocaldas.com/html5/como-incluir-css3-en-html5>

Al crear una página web, se utiliza en primer lugar el lenguaje HTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

Características y ventajas de las CSS

Para definir las CSS, se realiza mediante una sintaxis especial, la cual se aplica a:

- Un sitio web entero, de modo que se puede definir la forma de toda la web de una sola vez.
- Un documento HTML o página, se puede definir la forma, en una pequeña porción de código en la cabecera, a toda la página.
- Una porción del documento, aplicando estilos visibles en una porción de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante, ya que ofrece potencia a la programación.

La potencia de la tecnología salta a la vista. Si antes el HTML se quedaba corto para maquetar las páginas y se tenía que utilizar trucos para conseguir efectos; ahora existen muchas más herramientas que permiten definir esta forma:

- Definir la distancia entre líneas del documento.
- Aplicar identado a las primeras líneas del párrafo.
- Colocar elementos en la página con mayor precisión, y sin lugar a errores.
- Y mucho más, como definir la visibilidad de los elementos, márgenes, subrayados, tachados.

Además, utilizando CSS se puede definir las dimensiones de los elementos de la página utilizando muchas más unidades:

- Pixels (px) y porcentaje (%), como antes.
- Pulgadas (in)
- Puntos (pt)
- Centímetros (cm)

Soporte de CSS en los Navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador. En la siguiente tabla se muestra el soporte de CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios.

Navegador	Motor	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
FireFox	Gecko	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Todos los selectores, pseudo-clases y muchas propiedades

Figura 95: Tabla soporte de CSS

Referencia: http://librosweb.es/css/capitulo_1/soporte_de_css_en_los_navegadores.html

Funcionamiento de las CSS

CSS funciona en base a declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por un selector, una propiedad y el valor que se le asigne.

```

h1{
    color: black;
    font-style: italic
}

```

Donde:

h1 es el selector

{color: black;} es la declaración, **color** es la propiedad y **black** es el valor

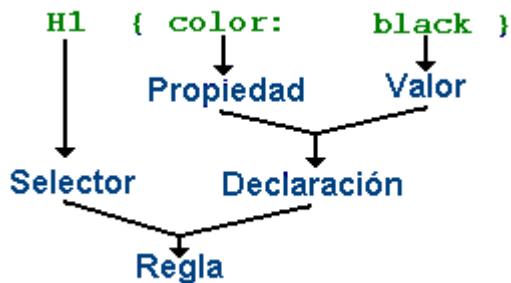


Figura 96: Estructura de CSS
Fuente. - Tomado de <http://tejedoresdelweb.com/w/CSS>

El **selector** funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La **declaración** es la parte de la regla que establece cuál será el efecto.

En el ejemplo anterior, el selector **h1** indica que todos los elementos de tipo **h1** se verán afectados por la declaración donde se establece que la propiedad color va a tener el valor red (rojo) para todos los elementos **h1** del documento o documentos que estén vinculados a esa hoja de estilos.

2.2.1.1 Tipos de Estilos

Estilo en línea

Es el estilo menos utilizado porque incluye la CSS en los mismos elementos de documento HTML.

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Documento sin título</title>
6  </head>
7  <body>
8
9  <p style="font-size:15px; color:blue">
10 Empresa comercial dedicada a la venta de
11 <span style="color:green">accesorios de
12 computadoras</span> y servicio tecnico
13 </p>
14 </body>
15 </html>
  
```

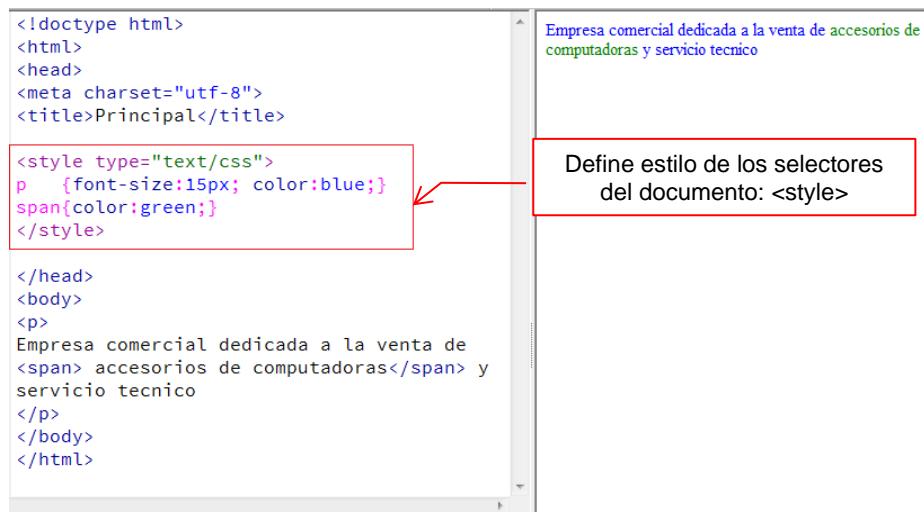
Figura 97: Estilos CSS
Fuente. - Elaboración propia

Asimismo, no es muy utilizado porque la regla solamente se aplica a un selector específico dentro del mismo documento.

Estilo interno

Si desea crear definiciones de estilo solamente para una página del sitio web, crea un estilo interno.

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta **<style>** de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección **<head>**).



The screenshot shows a web page with the following HTML code:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Principal</title>

<style type="text/css">
p {font-size:15px; color:blue;}
span{color:green;}
</style>

</head>
<body>
<p>
    Empresa comercial dedicada a la venta de
    <span> accesorios de computadoras</span> y
    servicio tecnico
</p>
</body>
</html>

```

A red box highlights the **<style>** block. A callout bubble points to it with the text: "Define estilo de los selectores del documento: <style>".

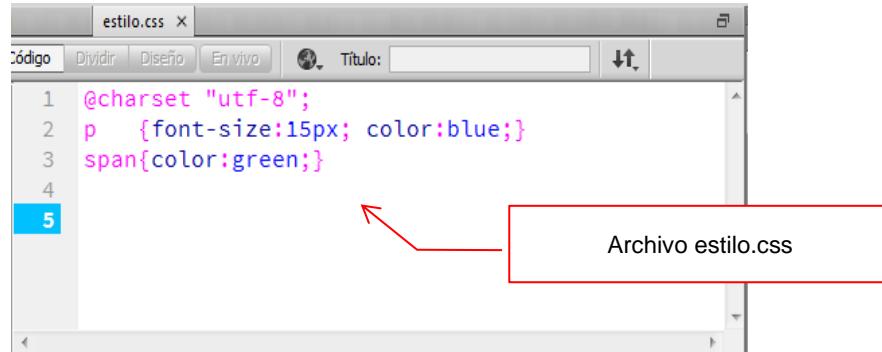
Figura 98: Estilo interno
Fuente. - Elaboración propia

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML, que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es que, si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

Estilo externo

Si deseas que el sitio Web tenga un aspecto uniforme, utiliza una hoja de estilos externa y vinculada a cada documento. Las hojas de estilo se pueden almacenar en **archivos externos** y vincularlas a uno o más documentos.



The screenshot shows a CSS editor window titled "estilo.css". The code is:

```

@charset "utf-8";
p {font-size:15px; color:blue;}
span{color:green;}

```

A red box highlights the entire code area. A callout bubble points to it with the text: "Archivo estilo.css".

Figura 99: Estilo externo
Fuente. - Elaboración propia

Una hoja de estilo externa es un archivo que contiene únicamente especificaciones CSS. Puedes utilizar hojas de estilo externas para asegurar una homogeneidad de diseño entre las páginas.

Un archivo de tipo CSS se podrá enlazar a una página HTML mediante la etiqueta <link>. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es .css Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

The screenshot shows a code editor with an HTML document. The code includes a <link> tag highlighted with a red box and a callout arrow pointing to a text box on the right. The text box contains the explanatory text: "La página HTML se enlaza al archivo CSS a través de la etiqueta <link>".

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Principal</title>

<link rel="stylesheet" type="text/css"
      href="../css/estilo.css" />

</head>
<body>
<p>
    Empresa comercial dedicada a la venta de
    de <span> accesorios de computadoras</span>
    y servicio tecnico
</p>
</body>
</html>

```

Figura 100: Etiqueta link
Fuente. - Elaboración propia

La etiqueta <link> incluye los siguientes atributos cuando enlaza un archivo CSS:

- **rel**: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor stylesheet.
- **type**: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es text/css.
- **href**: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.

Para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo **@import**. Las reglas de tipo **@import** siempre preceden a cualquier otra regla CSS (con la única excepción de la regla **@charset**).

La URL del archivo CSS externo se indica mediante una cadena de texto encerrada con comillas simples o dobles o mediante la palabra reservada url(). De esta forma, las siguientes reglas @import son equivalentes:

```

@import '/css/estilos.css';
@import "/css/estilos.css";
@import url('/css/estilos.css');
@import url("/css/estilos.css");

```

2.2.2 Selectores CSS

2.2.2.1 Selector universal

Se utiliza para seleccionar todos los elementos de la página. El selector universal se indica mediante un asterisco (*).

A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

El siguiente ejemplo elimina el margen, el borde y el relleno de todos los elementos HTML.

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

2.2.2.2 Selector de etiqueta

Este tipo de selector identifica a un tipo de elemento que conforma el código HTML modificando el aspecto predeterminado de la etiqueta.

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p{  
    color:red;  
    text-align:justify;  
    font-style:italic;  
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores. En el siguiente ejemplo, los títulos de sección h1, h2 y h3 comparten los mismos estilos:

```
h1, h2, h3{  
    color:blue;  
    text-align:left;  
    text-decoration:underline;  
    font-style:italic;  
}
```

En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

2.2.2.3 Selector descendente

Este tipo de selector, selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

```
p > span{
    color:green;
    font-size:18px;
}
```

Se utiliza para seleccionar un elemento que es hijo de otro elemento y se indica mediante el "signo de mayor que" (>).

Mientras que, en el selector descendente sólo importa que un elemento esté dentro de otro, independientemente de lo profundo que se encuentre, en el selector de hijos el elemento debe ser hijo directo de otro elemento.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Documento sin título</title>
<style type="text/css">
p{
    color:blue;
    font-style:italic;
    font-size:14px;
}
p > span{
    color:green;
    font-size:18px;
}
</style>
</head>

<body>
<p>
    Empresa comercial dedicada a la venta de
    accesorios para computadores y servicio
    tecnico especializado.
</p>
</body>
</html>
```

Figura 101: Selector descendente
Fuente. - Elaboración propia

2.2.2.4 Selector de clase

Cuando se declara una regla en la hoja de estilo, esta regla se aplica a todos los elementos HTML descritos por su selector.

¿Qué sucede si en ciertos casos se quiere usar un párrafo con características diferenciadas? Este planteamiento se resuelve en utilizar el atributo **class** de HTML, para definir un nuevo selector. El selector de clases consta de un punto (.) seguido por el nombre de la clase que se haya creado.

A continuación, define una nueva regla llamada .título con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo **class** con un punto (.).

```
.título{
    color:black;
    font-size:24px;
    font-style:normal;
    text-align:center;
}
```

Para aplicar el estilo de la clase CSS sobre un elemento se utiliza el atributo **class** junto con el nombre de la regla:

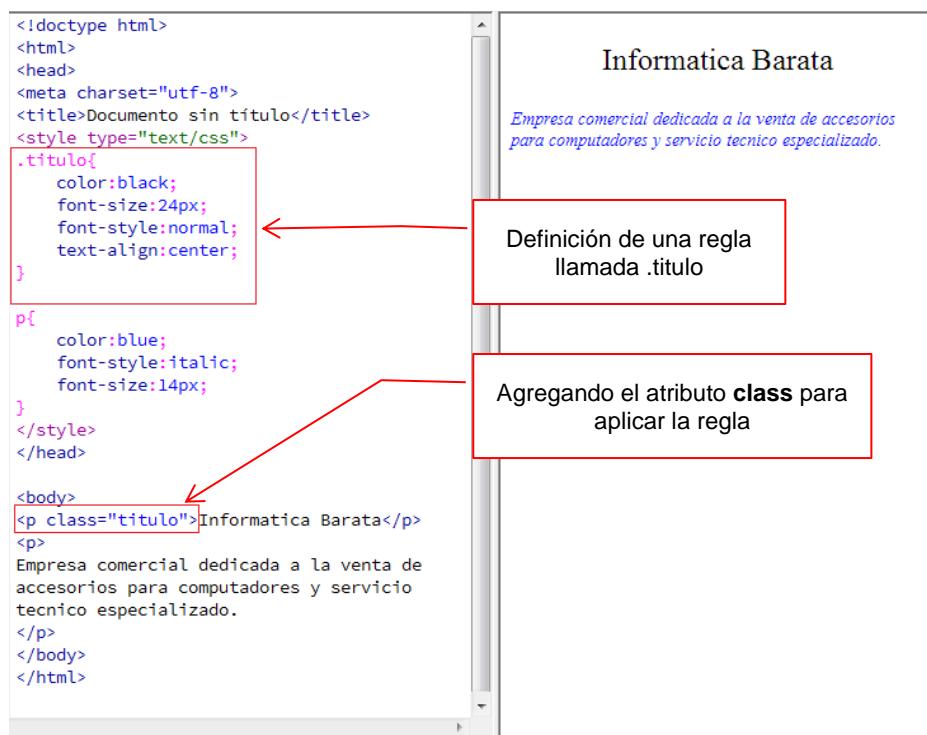


Figura 102: Selector de class
Fuente. - Elaboración propia

2.2.2.5 Selector de id

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo **id**. Este tipo de selectores solo seleccionan un elemento de la página porque el valor del atributo **id** no se puede repetir en dos elementos diferentes de una misma página.

```
#titulo{
    color:black;
    font-size:24px;
    font-style:normal;
    text-align:center;
}
```

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.), como prefijo del nombre de la regla CSS:

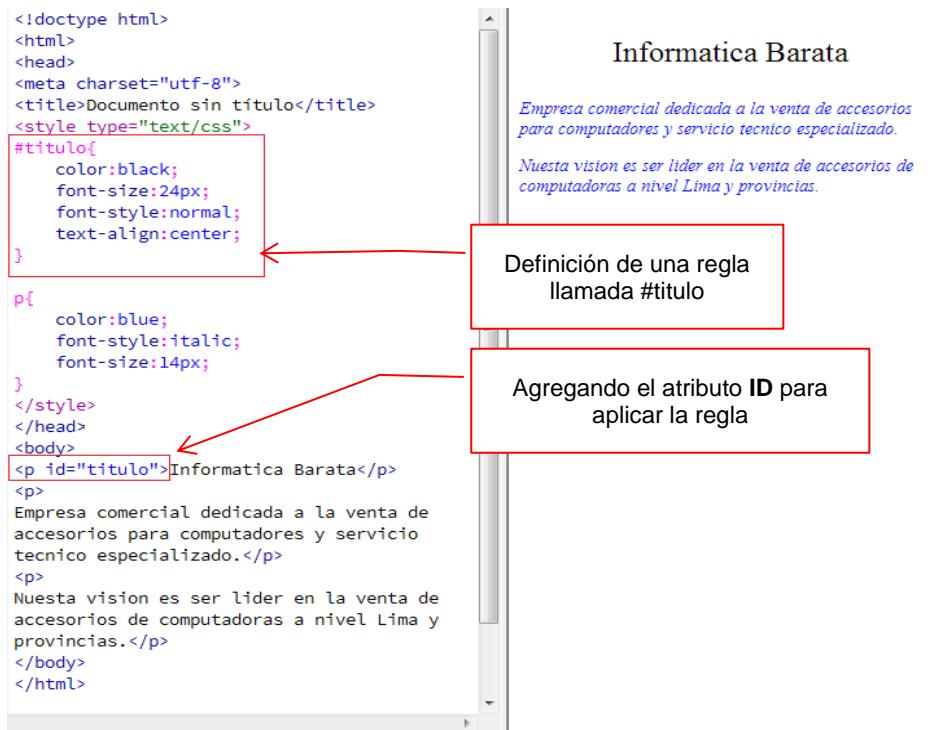


Figura 103: Selector de ID
Fuente. - Elaboración propia

2.2.2.6 Combinación de selectores

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación, se muestran algunos ejemplos habituales de combinación de selectores.

.aviso .especial { ... }

El selector selecciona aquellos elementos con un class="especial" que se encuentren dentro de cualquier elemento con un class="aviso".

Si se modifica el anterior selector:

div.aviso span.especial { ... }

Ahora, el selector solamente selecciona aquellos elementos de tipo con un atributo class="especial" que estén dentro de cualquier elemento de tipo <div>, que tenga un atributo class="aviso".

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:

```
ul#menuPrincipal li.destacado a#inicio { ... }
```

El anterior selector hace referencia al enlace con un atributo id igual a inicio que se encuentra dentro de un elemento de tipo con un atributo class igual a destacado, que forma parte de una lista con un atributo id igual a menuPrincipal.

LABORATORIO 1

Trabajando con Estilos CSS

Se nos pide implementar una página para promocionar las tecnologías y sus aportaciones al mundo, tal como se visualiza.

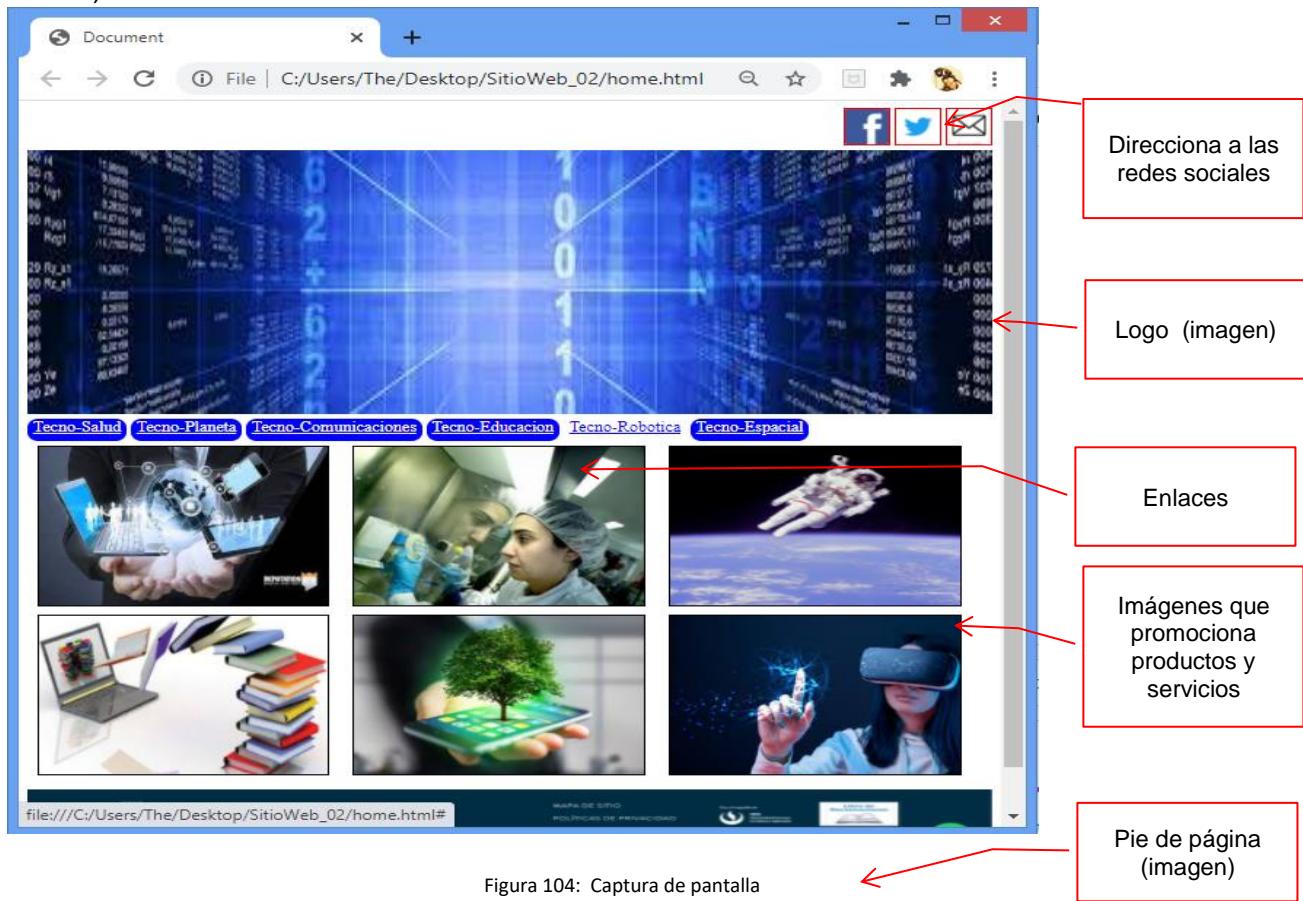


Figura 104: Captura de pantalla
Fuente. - Elaboración propia

Estructura del sitio Web

Define la estructura de carpetas del sitio web (**SitioWeb_02**), tal como se muestra. En la carpeta imágenes, almacene archivos .jpg para el diseño de la página HTML.

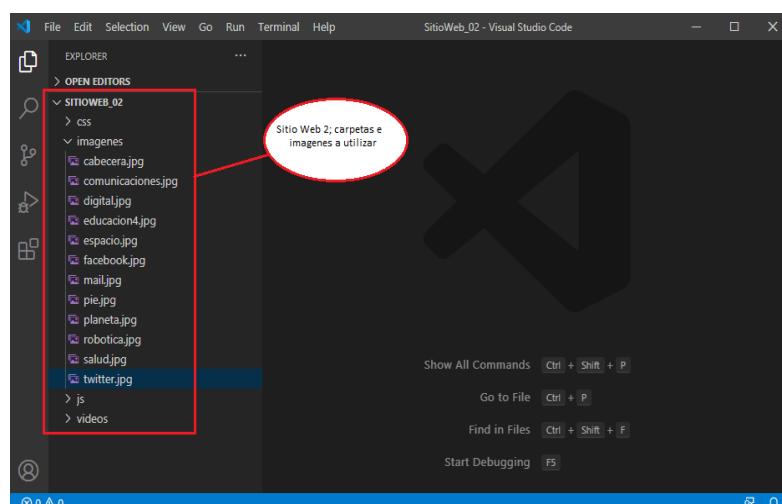


Figura 105: Captura de pantalla
Fuente. - Elaboración propia

Agregando un archivo CSS

Antes de iniciar con el diseño de la página HTML, primero crea un archivo CSS, para definir las reglas de estilos que utilizará la página:

1. Presiona la tecla **Ctrl+N** para crear un nuevo archivo.
2. A continuación, presiona las teclas **Ctrl+S** para guardar, selecciona la carpeta **css** y **Abrir**; luego, escribe el nombre del archivo, selecciona el tipo: **CSS** y presiona **Guardar**.

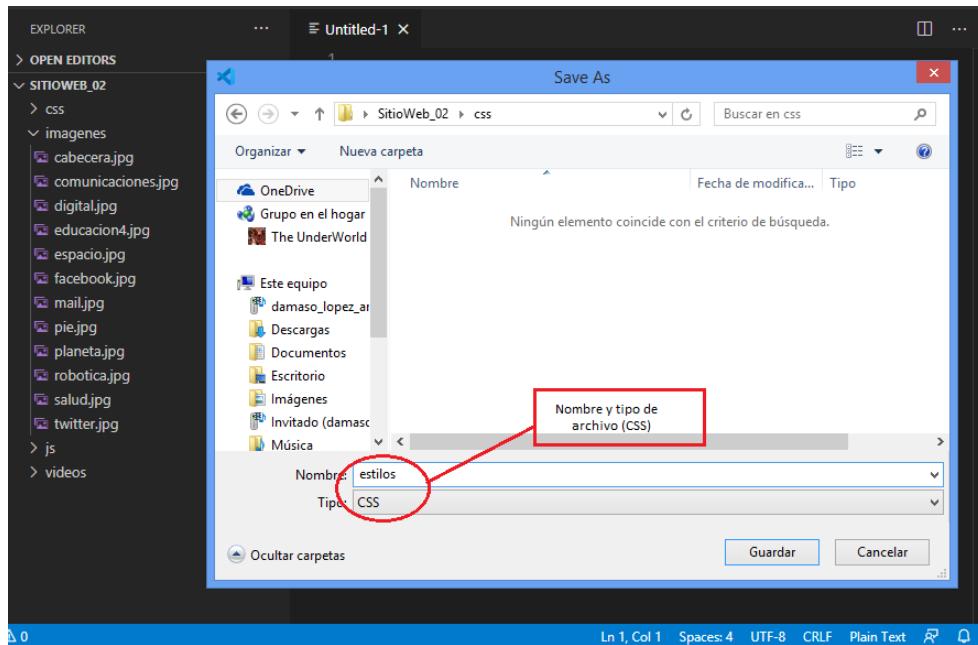


Figura 106: Captura de pantalla

Fuente. - Elaboración propia

Agregando la página HTML

A continuación, crea un archivo HTML:

1. Presiona la tecla **Ctrl+N** para crear un nuevo archivo.
2. Presiona la tecla **Ctrl+S** para guardar, selecciona la carpeta raíz y abrir; escribe el nombre del archivo "**home**" y selecciona el tipo **HTML**.

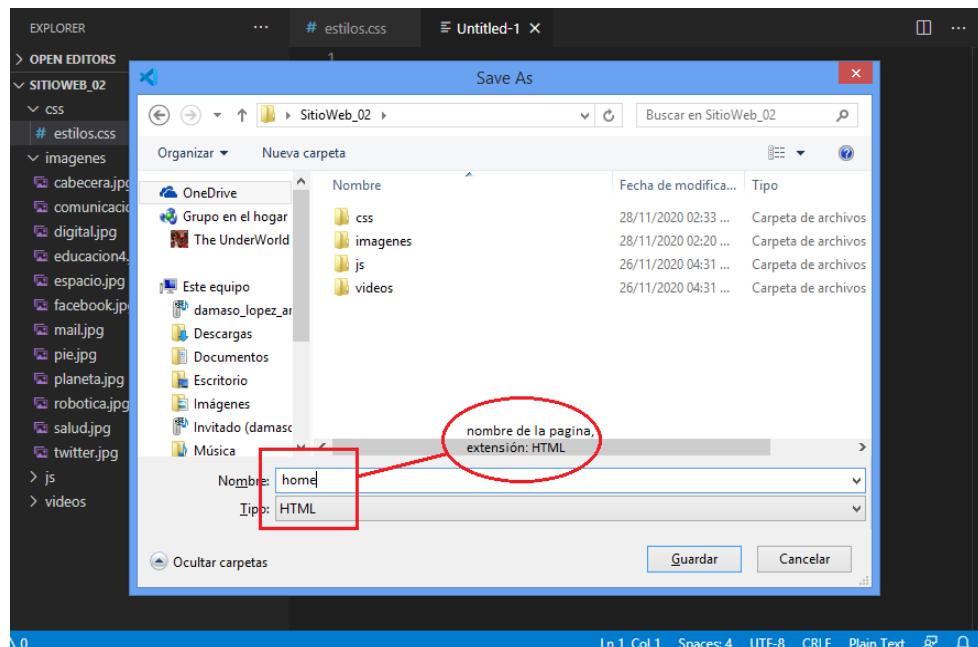


Figura 107: Captura de pantalla

Fuente. - Elaboración propia

En la página, ya creada, agrega la estructura HTML. Para ello, escribe en la primera línea **html**, selecciona la lista **html:5**, y presiona **ENTER**.

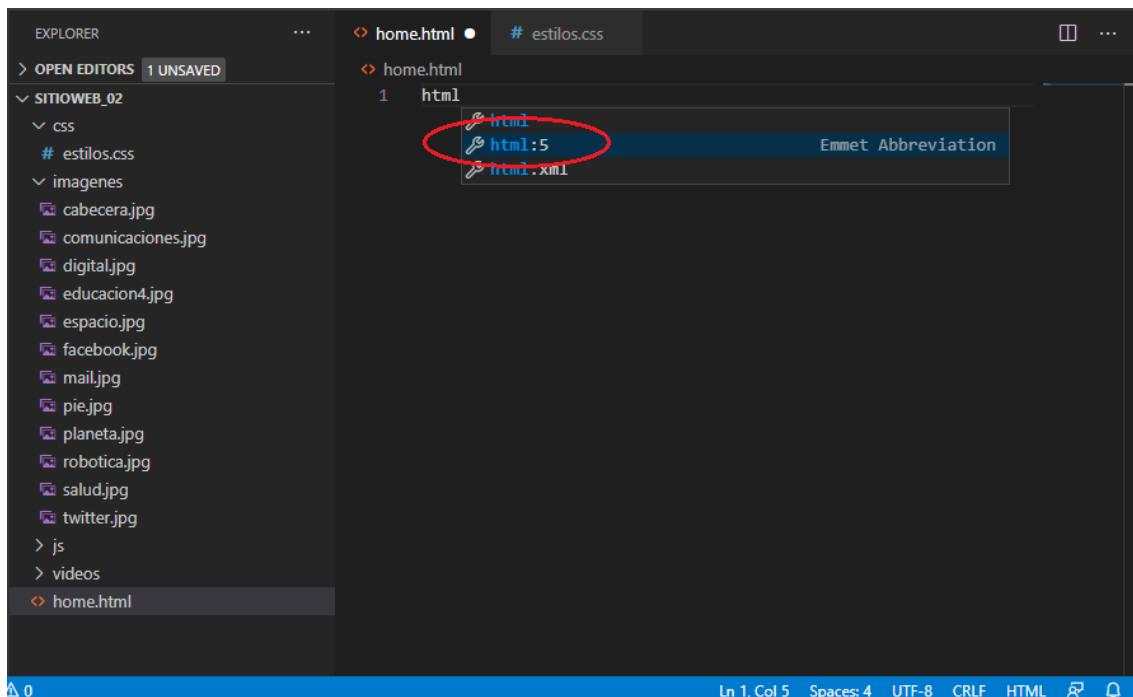


Figura 108: Captura de pantalla
Fuente. - Elaboración propia

Enlazando la hoja de estilo a home.html

Creada la página (home.html), define, dentro del **<head>** la etiqueta **<link>**, el cual hace una referencia al archivo **CSS estilos.css**, tal como se muestra.

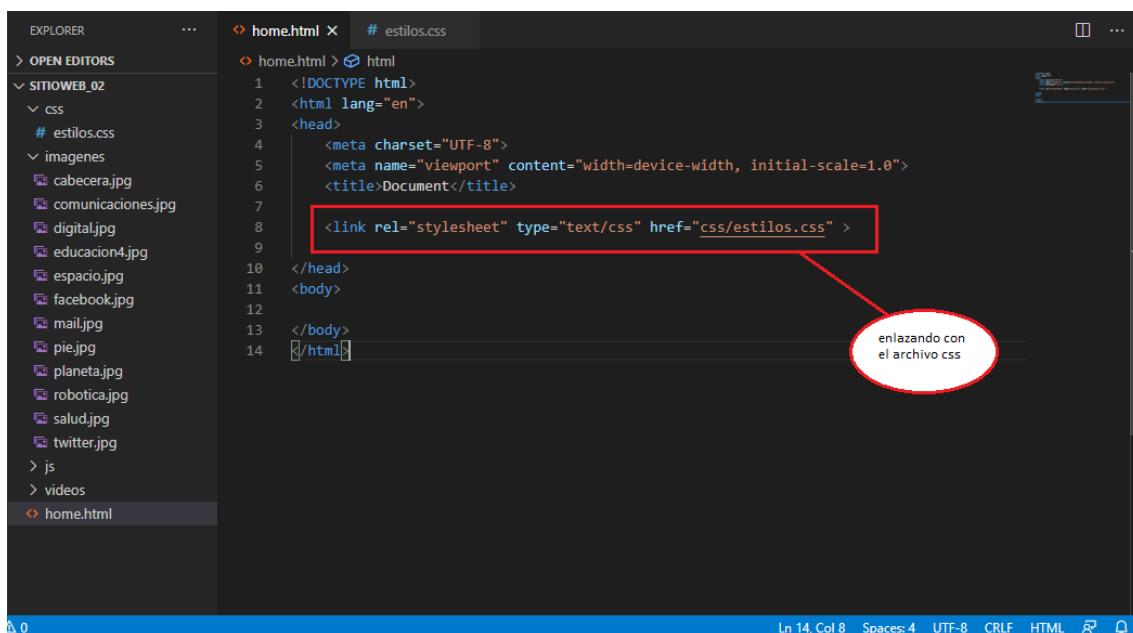


Figura 109: Captura de pantalla
Fuente. - Elaboración propia

Definiendo reglas de estilos en el archivo estilos.css

En el archivo CSS, estilo.css define las reglas: #bloque-rs, .rs y #logo

- Selector id bloque-rs, agrupa a las imágenes alineándolas a la derecha
- Selector de clase rs, es una regla para las imágenes de redes sociales.
- Selector id logo, es una regla para la imagen de logo de la página.

A continuación, guarda el contenido del archivo estilos.css: **CTRL+S**

```

1  #bloque-rs{
2      width: 100%;
3      height: auto;
4      text-align: right;
5  }
6
7
8  .rs{
9      width: 40px;
10     height: 40px;
11     border: 1px solid red;
12 }
13
14 #logo{
15     width: 100%;
16     height: 300px;
17 }
18

```

Ln 1, Col 1 Spaces:4 UTF-8 CRLF CSS ⌂ ⌂

Figura 110: Captura de pantalla

Fuente. - Elaboración propia

Agregando las etiquetas y reglas en la página

Definida las reglas en estilo.css, dentro del body define las etiquetas **** agregando a cada una su selector: de clase para las redes sociales y de identificador para el logo de la página.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7
8      <link rel="stylesheet" type="text/css" href="css/estilos.css" >
9
10 </head>
11 <body>
12     <div id="bloque-rs">
13         
14         
15         
16     </div>
17
18     
19 </body>
20 </html>

```

Ln 1, Col 16 Spaces:4 UTF-8 CRLF HTML ⌂ ⌂

Figura 111: Captura de pantalla

Fuente. - Elaboración propia

Definiendo las reglas de los enlaces (Menú) en el archivo estilo.css

En el archivo CSS, estilo.css define las reglas para la etiqueta `<a>` y `hover`.

- Define un selector id “bloque-menu”, donde agrupe los elementos `<a>`
- Define la clase **menu**, es una regla para los enlaces.
- Define la regla para `a: hover` (donde al pasar el mouse cambia su apariencia).

A continuación, guarda el archivo estilos.css: **CTRL+S**

```

css > # estilos.css > #bloque-...
12 > #logo{ ...
15 }
16
17 #bloque-menu{
18   width: 100%;
19   height: auto;
20 }
21
22 .menu{
23   display: inline-block;
24   background-color: #blue;
25   border-radius: 10px;
26   color: #white;
27   height: 25px;
28   padding-left: 5px;
29   padding-right: 5px;
30 }
31
32 .menu:hover{
33   background-color: #white;
34   color: #blue;
35 }

```

Figura 112: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta `<a>` y sus reglas en la página

Definida las reglas en estilos.css, en el body define el elemento de bloque `<div>` asignando su selector id (agrupa los elementos `<a>`) y agrega las etiquetas `<a>` añadiendo a cada una su selector de clase: **menú**, tal como se muestra.

```

home.html > # estilos.css
home.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head> ...
10 </head>
11 <body>
12 >   <div id="bloque-...
13   </div>
14
15   
16
17   <div id="bloque-menu">
18     <a href="#" class="menu">Tecno-Salud</a>
19     <a href="#" class="menu">Tecno-Planeta</a>
20     <a href="#" class="menu">Tecno-Comunicaciones</a>
21     <a href="#" class="menu">Tecno-Educacion</a>
22     <a href="#" class="menu">Tecno-Robotica</a>
23     <a href="#" class="menu">Tecno-Espacial</a>
24   </div>
25 </body>
26 </html>

```

Figura 113: Captura de pantalla

Fuente. - Elaboración propia

Definiendo las reglas de las imágenes en el archivo estilo.css

En el archivo CSS, estilo.css define las reglas para la etiqueta `` y el pie de página ``.

- Define la clase **tecnología**, es una regla para las imágenes de tecnología.
- Define el identificador **pie**, para la imagen del pie.

```

home.html      # estilos.css X
css > # estilos.css > #pie
19   }
20 > .menu{ ...
28   }
29 > .menu:hover{ ...
32   }
33
34 .tecnologia{
35   width:30%;
36   margin-left: 1%;
37   margin-right: 1%;
38   margin-top: 5px;
39   border: 1px solid;
40 }
41
42 #pie{
43   width: 100%;
44   height: 80px;
45   margin-top: 10px;
46 }

```

The screenshot shows a code editor with two red boxes highlighting specific CSS rules. A callout bubble points to the first box with the text "selector de class para las imágenes de tecnología". Another callout bubble points to the second box with the text "selector id para la imagen de pie de página".

Figura 114: Captura de pantalla

Fuente. - Elaboración propia

Agregando las etiquetas `` para el cuerpo y pie en la página

Definida las reglas en estilos.css, dentro del body, define las etiquetas `` para tecnologías agregando la clase **tecnología**; y la etiqueta `` para el pie de página, agregando su identificador **pie**, tal como se muestra.

```

home.html      # estilos.css
home.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head> ...
10 </head>
11 <body>
12 >   <div id="bloque-rs"> ...
16 </div>
17
18   
19
20 >   <div id="bloque-menu"> ...
27 </div>
28
29   
30   
31   
32   
33   
34   
35
36   
37 </body>
38 </html>

```

The screenshot shows a code editor with two red boxes highlighting `` tags. A callout bubble points to the first box with the text "Elementos con selector tecnologia". Another callout bubble points to the second box with the text "Elemento con selector pie".

Figura 115: Captura de pantalla

Fuente. - Elaboración propia

Para ejecutar a través del navegador, primero guarda la página: **CTRL+S** y presiona la tecla **F5** para ejecutar la página con el navegador.

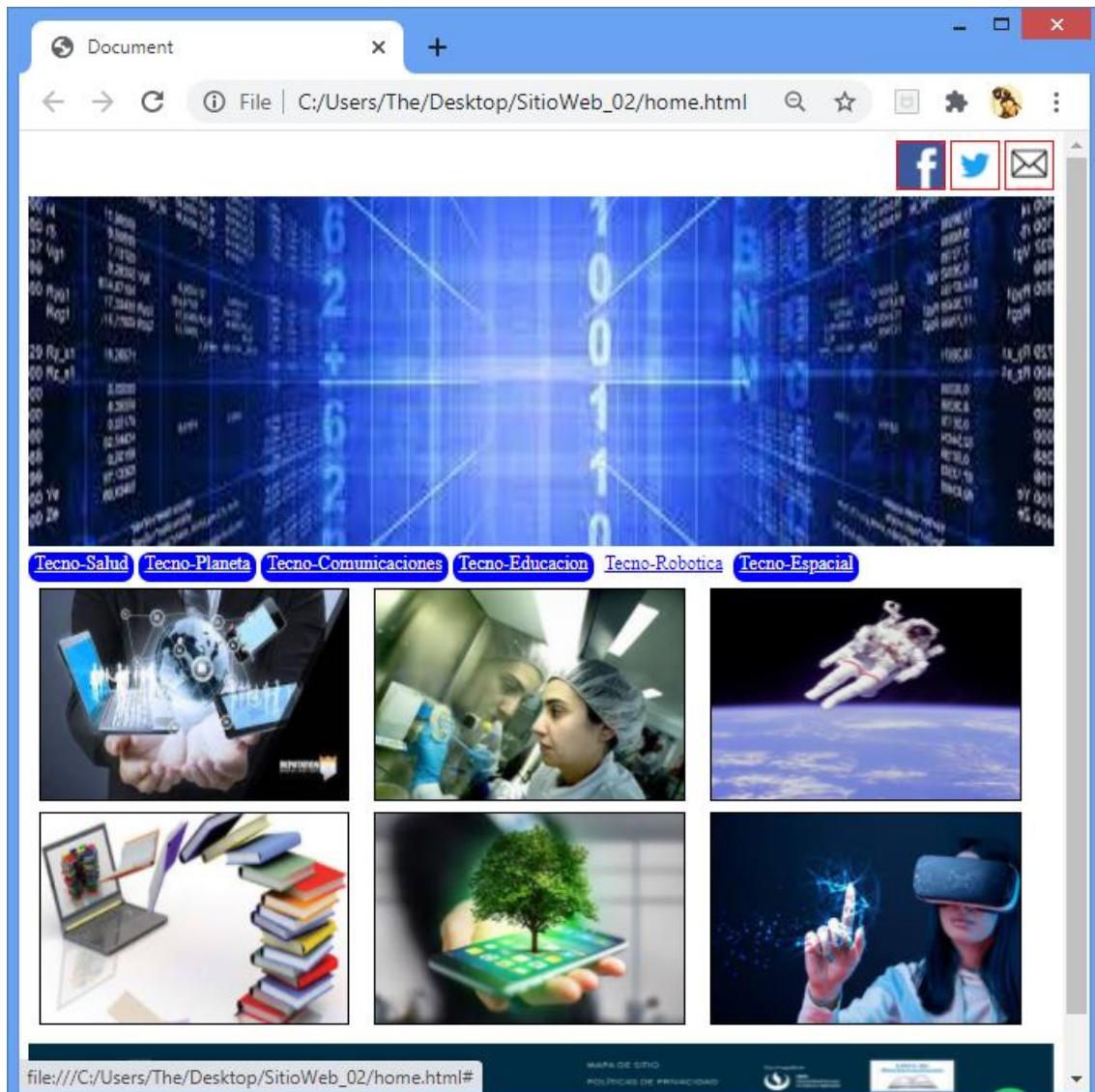


Figura 116: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 2

Trabajando con Estilos CSS

Se solicita implementar una página para hablar acerca de la tecnología e Informática, tal como se visualiza.

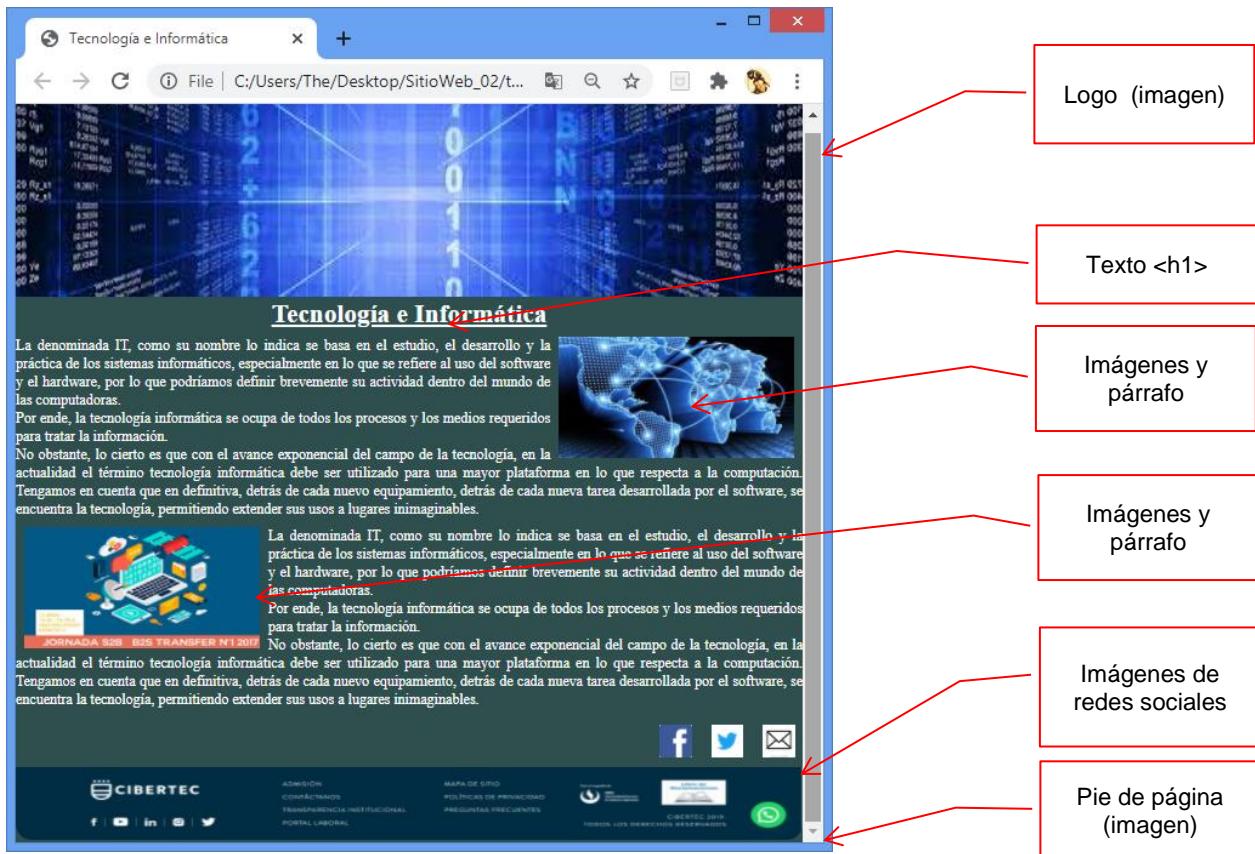


Figura 117: Captura de pantalla
Fuente. - Elaboración propia

Creando la pagina HTML

A continuación, crea una página HTML y nómbrala **tecnología-informática.html**, tal como se muestra. Presiona el botón **Guardar**.

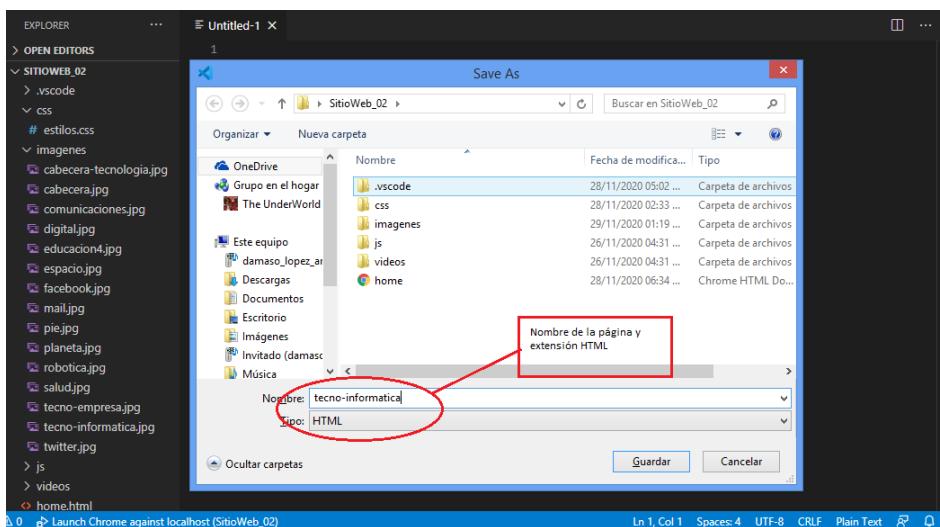


Figura 118: Captura de pantalla
Fuente. - Elaboración propia

Creando la hoja de estilo CSS

Crea la hoja de estilo, dentro de la carpeta CSS, llamada **estilo-informática.css**, tal como se muestra. Luego, presiona el botón **Guardar**.

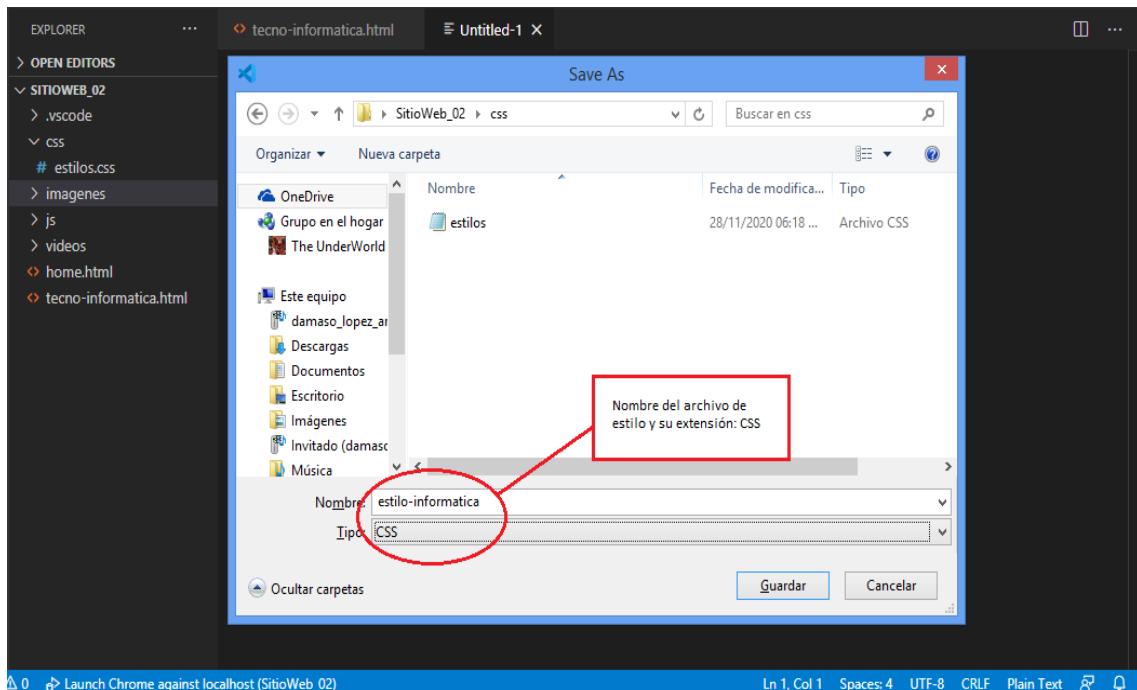


Figura 119: Captura de pantalla
Fuente. - Elaboración propia

Enlazando a la hoja de estilo

Creada la página, define dentro de la etiqueta **<head>**, la etiqueta **<link>**, la cual hace una referencia al archivo CSS **estilo-informatica.css**, tal como se muestra.

```

<!-- tecno-informatica.html X # estilos.css -->
<!-- tecno-informatica.html > ... -->
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <title>Tecnología e Informática</title> Título de la página
8
9      <link rel="stylesheet" href="css/estilo-informatica.css"> Enlazando con el archivo css (hoja de estilo)
10
11 </head>
12 <body>
13
14 </body>
15 </html>

```

Figura 120: Captura de pantalla
Fuente. - Elaboración propia

Definiendo reglas al body, cabecera y título de la página

En el archivo CSS, **estilo.css** define las primeras reglas:

- Define el selector universal (*), donde ningún elemento tendrá márgenes.
- Define la regla para el **body**, define el color de fondo de la página.
- Define el id **img-título**, regla para la imagen de la cabecera de la página.
- Define el id **título**, regla para el título de la página.

A continuación, guarda el archivo: **CTRL+S**

```

css > # estilo-informatica.css > #titulo
css > # estilo-informatica.css > *
1   *{
2     margin:0;
3     padding:0;
4   }
5
6 body{
7   background-color: darkslategray;
8 }
9
10 #img-titulo{
11   width: 100%;
12   height: 250px;
13   border-radius: 20px 20px 0 0;
14 }
15
16 #titulo{
17   text-align: center;
18   color: white;
19   text-decoration: underline;
20 }

```

Figura 121: Captura de pantalla
Fuente. - Elaboración propia

Agregando las etiquetas y reglas en la página

Definida las reglas en estilo-informatica.css, asigna las reglas a la etiqueta de la página.

```

tecnoinformatica.html > # estilo-informatica.css
tecnoinformatica.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>...
11 </head>
12 <body>
13   
14
15   <h1 id="titulo">Tecnología e Informática</h1>
16 </body>
17 </html>

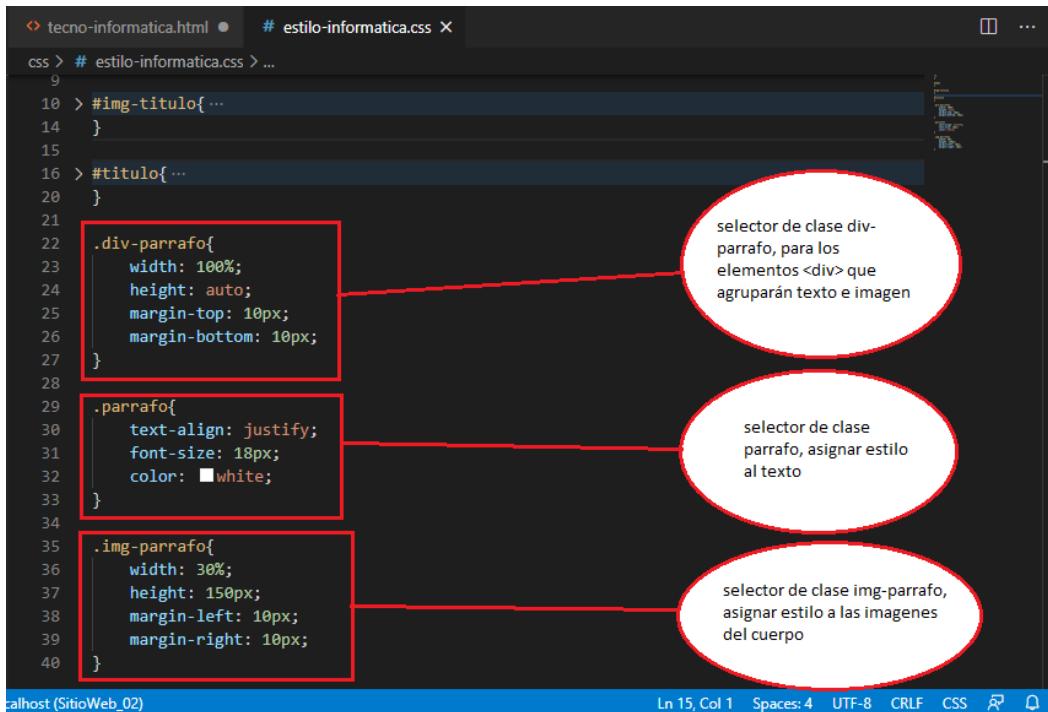
```

Figura 122: Captura de pantalla
Fuente. - Elaboración propia

Definir reglas de los párrafos y de las imágenes en la hoja de estilo

En el archivo CSS, define reglas para el párrafo y las imágenes del cuerpo.

- Define la clase **div-párrafo**, regla que da formato a los elementos <div>.
- Define la clase **párrafo**, regla que da formato a los párrafos de la página.
- Define la clase **img-párrafo**, es una regla para las imágenes.
- A continuación, guarda el archivo estilo.css: **CTRL+S**.



The screenshot shows a code editor with three highlighted CSS rules:

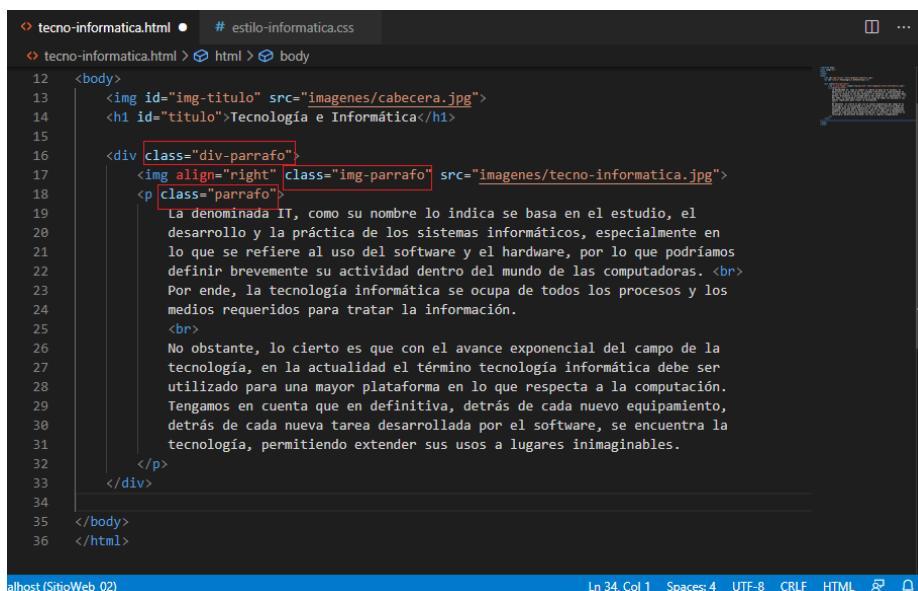
- .div-parrafo{ width: 100%; height: auto; margin-top: 10px; margin-bottom: 10px; }**: A red box highlights this rule. A callout bubble points to it with the text: "selector de clase div-parrafo, para los elementos <div> que agruparán texto e imagen".
- .parrafo{ text-align: justify; font-size: 18px; color: white; }**: A red box highlights this rule. A callout bubble points to it with the text: "selector de clase parrafo, asignar estilo al texto".
- .img-parrafo{ width: 30%; height: 150px; margin-left: 10px; margin-right: 10px; }**: A red box highlights this rule. A callout bubble points to it with the text: "selector de clase img-parrafo, asignar estilo a las imágenes del cuerpo".

Figura 123: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <p> e para el cuerpo de la página

Definida las reglas en estilo.css, define las etiquetas <p> e para promocionar los productos. Utiliza la clase **párrafo** para la etiqueta <p> y la clase **img-párrafo** en la etiqueta del cuerpo de la página.



The screenshot shows a code editor displaying both HTML and CSS. The HTML includes:

```

<body>
  
  <h1 id="titulo">Tecnología e Informática</h1>

  <div class="div-parrafo">
    
    <p class="parrafo">
      La denominada IT, como su nombre lo indica se basa en el estudio, el desarrollo y la práctica de los sistemas informáticos, especialmente en lo que se refiere al uso del software y el hardware, por lo que podríamos definir brevemente su actividad dentro del mundo de las computadoras. <br>
      Por ende, la tecnología informática se ocupa de todos los procesos y los medios requeridos para tratar la información.
      <br>
      No obstante, lo cierto es que con el avance exponencial del campo de la tecnología, en la actualidad el término tecnología informática debe ser utilizado para una mayor plataforma en lo que respecta a la computación.
      Tengamos en cuenta que en definitiva, detrás de cada nuevo equipamiento, detrás de cada nueva tarea desarrollada por el software, se encuentra la tecnología, permitiendo extender sus usos a lugares inimaginables.
    </p>
  </div>
</body>
</html>

```

The CSS file (estilo-informatica.css) defines the classes used in the HTML:

```

.div-parrafo{
  width: 100%;
  height: auto;
  margin-top: 10px;
  margin-bottom: 10px;
}

.parrafo{
  text-align: justify;
  font-size: 18px;
  color: white;
}

.img-parrafo{
  width: 30%;
  height: 150px;
  margin-left: 10px;
  margin-right: 10px;
}

```

Figura 124: Captura de pantalla

Fuente. - Elaboración propia

De igual modo, para el segundo párrafo, utiliza los selectores para los elementos de la página; y de esta manera los dos párrafos de la página utilizan las mismas reglas.

```

12 <body>
13   
14   <h1 id="titulo">Tecnología e Informática</h1>
15
16 >   <div class="div-parrafo"> ...
17   </div>
18
19   <div class="div-parrafo">
20     
21     <p class="parrafo">
22       La denominada IT, como su nombre lo indica se basa en el estudio, el
23       desarrollo y la práctica de los sistemas informáticos, especialmente en
24       lo que se refiere al uso del software y el hardware, por lo que podriamos
25       definir brevemente su actividad dentro del mundo de las computadoras. <br>
26       Por ende, la tecnología informática se ocupa de todos los procesos y los
27       medios requeridos para tratar la información.
28     <br>
29     No obstante, lo cierto es que con el avance exponencial del campo de la
30     tecnología, en la actualidad el término tecnología informática debe ser
31     utilizado para una mayor plataforma en lo que respecta a la computación.
32     Tengamos en cuenta que en definitiva, detrás de cada nuevo equipamiento,
33     detrás de cada nueva tarea desarrollada por el software, se encuentra la
34     tecnología, permitiendo extender sus usos a lugares inimaginables.
35   </p>
36 </div>
37 </body>
38 </html>

```

localhost (SitoWeb_02) Ln 52, Col 11 Spaces: 4 UTF-8 CRLF HTML

Figura 125: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas de las imágenes: pie, en el archivo css

En el archivo CSS, define las reglas siguientes:

- Selector de clase **div-rs** para el contenedor de las imágenes de redes sociales.
- Selector de clase **rs** para las imágenes de las redes sociales.
- Selector id **pie** para dar estilo a la imagen del pie de página.

A continuación, guarda el archivo css: **CTRL+S**.

```

27 }
28
29 > .parrafo{ ...
30 }
31
32 > .img-parrafo{ ...
33 }
34
35 > .div-rs{
36   width: 100%;
37   height: auto;
38   text-align: right;
39 }
40
41 .rs{
42   width: 40px;
43   height: 40px;
44   margin: 1% 0;
45 }
46
47 #pie{
48   width: 100px;
49   height: 90px;
50   border-radius: 0 20px 20px;
51 }
52
53
54
55
56
57
58

```

localhost (SitoWeb_02) Ln 10, Col 13 Spaces: 4 UTF-8 CRLF CSS

Figura 126: Captura de pantalla
Fuente. - Elaboración propia

Agregando las etiquetas para las redes sociales y el pie de página

Agrega los elementos para las imágenes de las redes sociales y la imagen del pie de página, tal como se muestra. Presiona la tecla **F5** para visualizar el diseño.

```

tecnologia.html # estilo-informatica.css
tecnologia.html > html
12   <body>
13     Tecnología e Informática</h1>
15   >   <div class="div-parrafo">...
16   >   <div class="div-parrafo">...
17   >     <div class="div-rs">
18       
19       
20       
21     </div>
22     
23   </body>
24 </html>

```

alhost (SitioWeb_02) Ln 2, Col 17 Spaces: 4 UTF-8 CRLF HTML

Figura 127: Captura de pantalla

Fuente. - Elaboración propia

Para ejecutar a través del navegador, primero guarda la página: **CTRL+S** y presiona la tecla **F5** para ejecutar la página con el navegador.



Figura 128: Captura de pantalla

Fuente. - Elaboración propia

Resumen

1. CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, las cuales están escritas con código HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.

2. El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.
3. CSS funciona en base a declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. La regla tiene dos partes: un selector y la declaración.
4. Estilo interno, si se desea crear definiciones de estilo solamente para una página del sitio web. Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta **<style>** de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección **<head>**).
5. Estilo externo, si se desea que el sitio Web tenga un aspecto uniforme se debe utilizar una hoja de estilos externa y vinculada a cada documento. Las hojas de estilo se pueden almacenar en **archivos externos** y vincularlas a uno o más documentos.
6. Para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo **@import**. Las reglas de tipo **@import** siempre preceden a cualquier otra regla CSS (con la única excepción de la regla **@charset**).
7. El selector de tipo o etiqueta es un tipo de selector que identifica a un tipo de elemento que conforma el código HTML modificando el aspecto predeterminado de la etiqueta. Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.
8. El selecto de clase es cuando se declara una regla en la hoja de estilo, esta regla se aplica a todos los elementos HTML descritos por su selector.
9. EL selector de ID, en ocasiones, es necesario aplicar estilos CSS a un único elemento de la página HTML. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

Recursos

Pueden revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- http://librosweb.es/css_avanzado/capitulo_3.html
- http://www.mclibre.org/consultar/htmlcss/css/css_pseudoclasses.html
- <http://www.sidar.org/recur/desdi/traduc/es/css/selector.html>
- <http://html.conclase.net/w3c/css1-es.html>
- <https://kechy.files.wordpress.com/2012/11/manual-css-hojas-estilo.pdf>
- http://librosweb.es/css/capitulo_2/selectores_basicos.html



ESTRUCTURAS Y BLOQUES

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y estructuras de bloques.

TEMARIO

3.1 Tema 5 : **Modelo de cajas y posicionamiento CSS**

- 3.1.1. : Modelo de cajas
- 3.1.2. : Posicionamiento de cajas: estático, relativo, absoluto, flotante; visualización: display, visibility, overflow, z-index
- 3.1.3. : Etiquetas estructurales: header, nav, aside, main, section, article, footer
- 3.1.4. : Etiquetas flexibles: box-sizing, flexible box layout, grid

3.2 Tema 6 : **Creación de menú**

- 3.2.1. : Diseño de menú
- 3.2.2. : Creando un menú utilizando estilo a los elementos

ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3 y bloques de contenidos.
- Los alumnos diseñan un menú en una página web.

3.1 MODELO DE CAJAS Y POSICIONAMIENTO CSS

3.1.1 Modelo de Cajas

El modelo de cajas es el comportamiento de estilo CSS que hace que todos los elementos de una página se representen mediante cajas rectangulares. Este modelo es la característica más importante del lenguaje de estilos CSS, ya que condiciona el diseño de todas las páginas web.

Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:

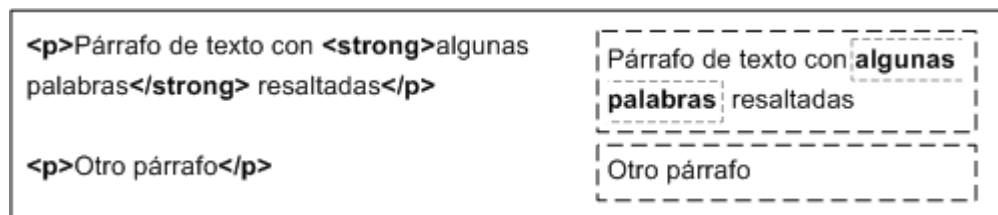


Figura 129: Modelamiento de cajas

Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-4>

Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen.

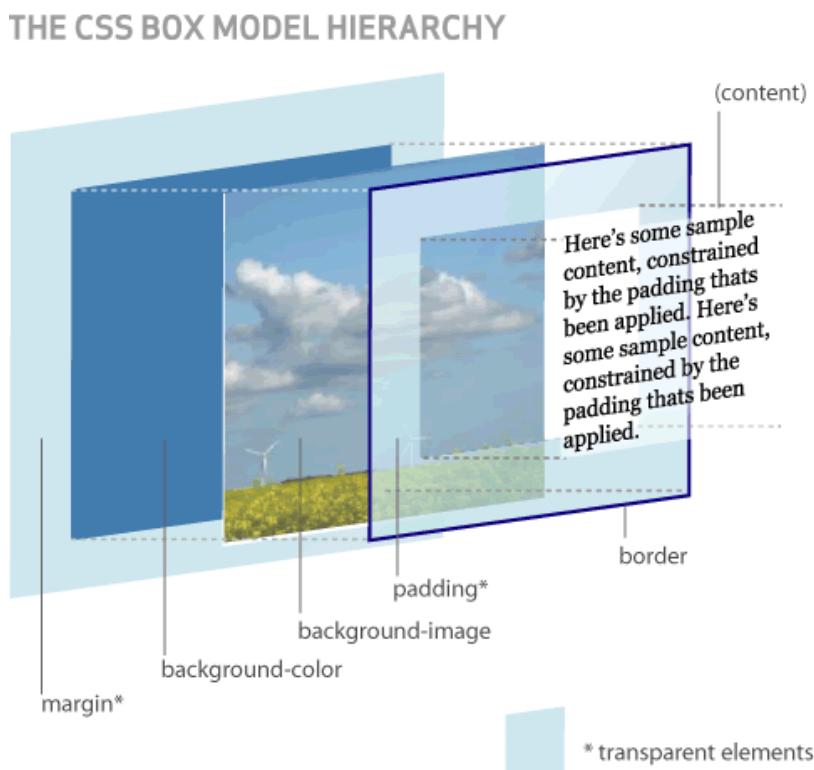


Figura 130: Modelo de cajas

Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-4>

Las partes que componen una caja, desde el punto de vista del usuario, son las siguientes:

Componente	Definición
Contenido (content)	Se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
Relleno (padding)	Espacio libre opcional existente entre el contenido y el borde
Borde (border)	Línea que encierra completamente el contenido y su relleno
Imagen de fondo (background image)	Imagen que se muestra por detrás del contenido y el espacio de relleno
Color de fondo (background color)	Color que se muestra por detrás del contenido y el espacio de relleno
Margen (margin)	Separación opcional existente entre la caja y el resto de cajas adyacentes

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

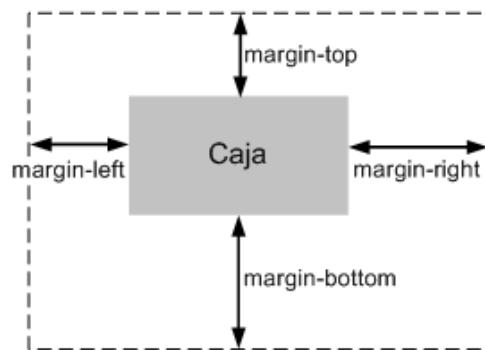


Figura 131: Márgenes externos
Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-4/margen-y-relleno>

CSS define cuatro propiedades para controlar cada uno de los espacios de relleno: horizontales y verticales de un elemento.

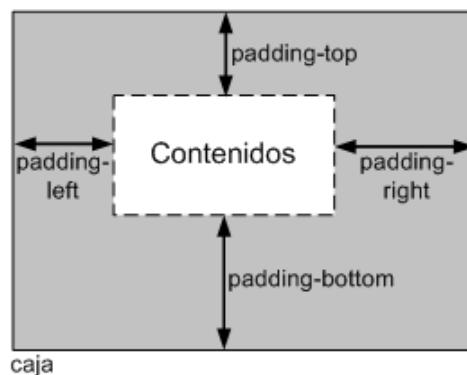


Figura 132: Márgenes internos
Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-4/margen-y-relleno>

Referente al background-image o imagen de fondo, CSS define cinco propiedades para establecer el fondo de cada elemento (background-color, background-image, background-repeat, background-attachment, background-position) y otra propiedad de tipo "shorthand" (background).

Propiedad	Descripción
background-color	Permite mostrar un color de fondo sólido en la caja de un elemento.
background-image	Permite mostrar una imagen como fondo de la caja de cualquier elemento.
background-repeat	Permite controlar la forma de repetición de las imágenes de fondo. El valor no-repeat muestra una sola vez la imagen y no se repite en ninguna dirección. El valor repeat-x repite la imagen solo horizontalmente y el valor repeat-y repite la imagen solamente de forma vertical.
background-position	Permite controlar la posición de la imagen dentro del fondo del elemento mediante la propiedad.
background-attachment	Opcionalmente, se puede indicar que el fondo permanezca fijo cuando la ventana del navegador se desplaza mediante las barras de scroll. Se trata de un comportamiento que en general no es deseable y que algunos navegadores no soportan correctamente.

La propiedad **border** permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su anchura o grosor, su color y su estilo, por lo que en total CSS define 20 propiedades relacionadas con los bordes.

La anchura se controla con las cuatro propiedades siguientes: **border-top-width**, **border-right-width**, **border-bottom-width** y **border-left-width**. Si se quiere establecer de forma simultánea el ancho de todos los bordes de una caja, se utiliza la propiedad **border-width**.

El color de los bordes se controla con las cuatro propiedades siguientes: **border-top-color**, **border-right-color**, **border-bottom-color** y **border-left-color**. Si se quiere establecer de forma simultánea el color de todos los bordes de una caja, se utiliza la propiedad **border-color**.

El estilo de los bordes se controla con las cuatro propiedades siguientes: **border-top-style**, **border-right-style**, **border-bottom-style** y **border-left-style**. Los bordes más utilizados son solid y dashed, seguidos de double y dotted. Si se quiere establecer de forma simultánea el color de todos los bordes de una caja, se utiliza la propiedad **border-style**.

La propiedad **border-radius** permite que las esquinas sean bordes redondeados. Esta propiedad en este momento es experimental por lo que se debe usar los prefijos –moz- y –webkit para que funcionen en navegadores basados en motores Gecko y WebKit, como Firefox, Safari y Google Chrome. Si todas las esquinas tienen la misma curvatura se puede utilizar un solo valor.

Para aplicarla efectivamente a sus documentos, se debe declarar con los correspondientes prefijos, para los navegadores más comunes son los siguientes:

- -moz- para Firefox.
- -webkit- para Safari y Chrome.
- -o- para Opera.
- -khtml- para Konqueror.

- -ms- para Internet Explorer.
- -chrome- específico para Google Chrome.

```
#principal {  
    display: block;  
    width: 500px;  
    margin: 50px auto;  
    padding: 15px;  
    text-align: center;  
    border: 1px solid #999999;  
    background: #DDDDDD;  
    -moz-border-radius: 20px 10px 30px 50px;  
    -webkit-border-radius: 20px 10px 30px 50px;  
    border-radius: 20px 10px 30px 50px;  
}
```

3.1.2 Posicionamiento de cajas: estático, relativo, absoluto y flotante; visualización: display, visibility, overflow, z-index

Cuando los navegadores descargan el contenido HTML y CSS de las páginas web, aplican un procesamiento muy complejo antes de mostrar las páginas en la pantalla del usuario.

Para cumplir con el modelo de cajas, los navegadores crean una caja para representar a cada elemento de la página HTML. Los factores que se tienen en cuenta para generar cada caja son los siguientes:

- Las propiedades width y height de la caja (si están establecidas).
- El tipo de cada elemento HTML (elemento de bloque o elemento en línea).
- Posicionamiento de la caja (normal, relativo, absoluto, fijo o flotante).
- Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- **Posicionamiento estático:** se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- **Posicionamiento relativo:** variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- **Posicionamiento absoluto:** la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.

- **Posicionamiento fijo:** variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- **Posicionamiento flotante:** se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

Posicionamiento relativo

Desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.

El valor de la propiedad top se interpreta como el desplazamiento entre el borde superior de la caja en su posición final y el borde superior de la misma caja en su posición original.

De la misma forma, el valor de las propiedades left, right y bottom indica respectivamente el desplazamiento entre el borde izquierdo/derecho/inferior de la caja en su posición final y el borde izquierdo/derecho/inferior de la caja original.

Por tanto, la propiedad top se emplea para mover las cajas de forma descendente, la propiedad bottom mueve las cajas de forma ascendente, la propiedad left se utiliza para desplazar las cajas hacia la derecha y la propiedad right mueve las cajas hacia la izquierda. Este comportamiento parece poco intuitivo y es causa de errores cuando se empiezan a diseñar páginas con CSS. Si se utilizan valores negativos en las propiedades top, right, bottom y left, su efecto es justamente el inverso.

El desplazamiento relativo de una caja no afecta al resto de cajas adyacentes, que se muestran en la misma posición que si la caja desplazada no se hubiera movido de su posición original.

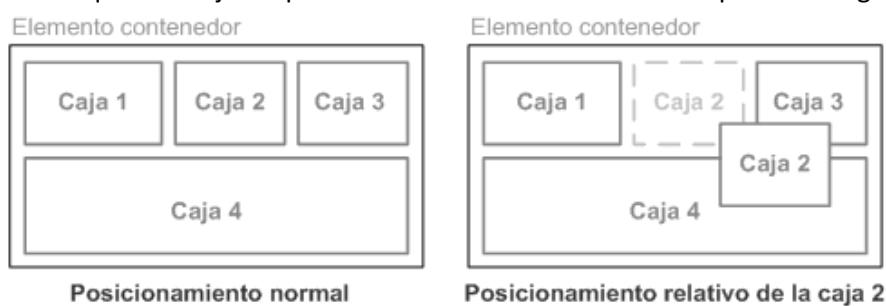


Figura 133: Posicionamiento relativo

Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-relativo>

Posicionamiento absoluto

Se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left. La interpretación de los valores de estas propiedades es mucho más compleja que en el posicionamiento relativo, ya que en este caso dependen del posicionamiento del elemento contenedor.

Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas.



Figura 134: Posicionamiento Absoluto

Fuente. - Tomado de <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-absoluto>

El desplazamiento de una caja posicionada de forma absoluta se controla mediante las propiedades top, right, bottom y left. A diferencia del posicionamiento relativo, la interpretación de los valores de estas propiedades depende del elemento contenedor de la caja posicionada.

Posicionamiento flotante

La mayoría de estructuras de las páginas web complejas están diseñadas con el posicionamiento flotante. Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba. La propiedad CSS que permite posicionar de forma flotante una caja se denomina **float**.

Si se indica un valor **left**, la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea. El resto de elementos adyacentes se adaptan y fluyen alrededor de la caja flotante.

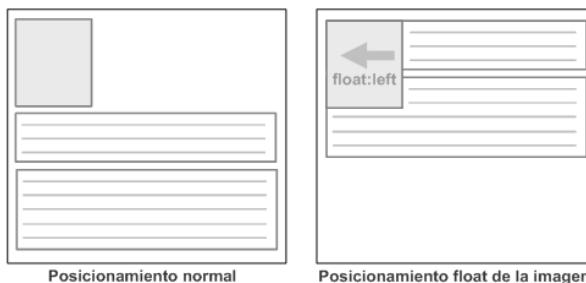


Figura 135: posicionamiento Float

Fuente.- Tomado de <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-flotante>

```
img {  
    float:left;  
}
```

El valor **right** tiene un funcionamiento idéntico, salvo que, en este caso la caja se desplaza hacia la derecha. El valor **none** permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original. La propiedad **clear** permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

La regla CSS que se aplica al segundo párrafo del ejemplo anterior es la siguiente:

```
<p style="clear: left;">...</p>
```

Visualización de las cajas

Las propiedades **display** y **visibility** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

La propiedad **display** permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.

Los valores más utilizados son **inline**, **block** y **none**. El valor **block** muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. El valor **inline** visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate. El valor **none** oculta un elemento y hace que desaparezca de la página.

La propiedad **visibility** permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento, pero no la muestra. En este caso, el resto de los elementos de la página no modifican su posición, ya que, aunque la caja no se ve, sigue ocupando sitio.

Valor	Descripción
hidden	Convierte una caja en invisible para que no muestre sus contenidos.
collapse	Se utiliza en las filas, grupos de filas, columnas y grupos de columnas de una tabla.
display	Muestra otros contenidos en ese lugar, su efecto es idéntico al valor hidden.

La propiedad **overflow** permite controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

Valor	Descripción
visible	El contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Valor por defecto.
hidden	El contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada del elemento.
scroll	Muestra una barra scroll para visualizar el contenido oculto.
auto	El comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

La propiedad **z-index**, además de posicionar una caja de forma horizontal y vertical, CSS permite controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.

La posición tridimensional de un elemento se establece sobre un tercer eje llamado Z y se controla mediante la propiedad **z-index**. Utilizando esta propiedad es posible crear páginas complejas con varios niveles o capas.

3.1.3 Etiquetas estructurales: header, nav, aside, main, section, article, footer

Para distribuir mejor el contenido de una página se han definido etiquetas estructurales. La razón para trabajar con etiquetas estructurales es dividir una página Web en partes lógicas que describan el tipo de contenido que incluyen.

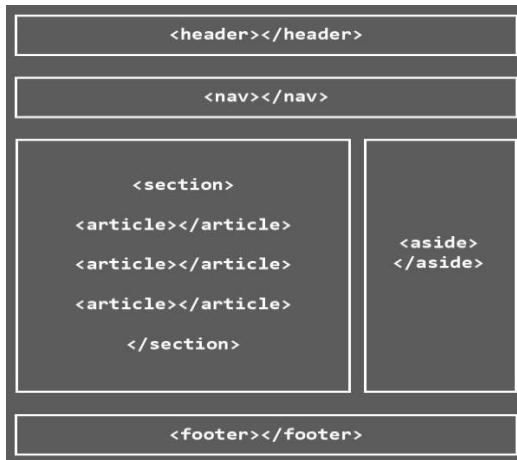


Figura 136: Estructura HTML5

Fuente .- Tomado de <https://www.anerbarrena.com/nuevas-etiquetas-semanticas-estructura-html5-4853/>

HTML5 aporta una serie de etiquetas nuevas que permiten mejorar la semántica de la página. Los elementos de HTML5 nuevos que se va a cubrir en este artículo son:

<header>

Es un contenedor que forma la cabecera de la página, generalmente con un logotipo o gráfico de la compañía, el título de página principal, etc.

<header> puede ser utilizado varias veces en un documento, por lo general, contiene al menos un elemento <h1>-<h6>.

<hgroup>

Notarás que, en el código anterior, el único contenido de la cabecera es un elemento <hgroup>, que rodea dos encabezados. Lo que se pretende especificar es el encabezado de nivel superior del documento, más un subtítulo o ‘tagline’. Si deseas que solo cuente el nivel superior de encabezado en la jerarquía de encabezados del documento, es exactamente lo que hace <hgroup>: que un grupo de encabezados cuente como un único encabezado a efectos de la estructura del documento.

<nav>

Es utilizado para demarcar una sección que contiene los enlaces a otros contenidos o secciones. Los enlaces se pueden anidar dentro de una lista, como una lista desordenada .

<section>

Se usa para representar un bloque de la página que tiene valor semántico, es decir, que aporta un significado a la página. Realmente si se tiene que clasificar por la importancia del significado, el ranking sería: article sería la etiqueta que contiene la información más relevante, section la que contiene información menos relevante, y div que contiene información que no aporta significado ninguno.

<article>

Se utiliza para contenidos independientes que tendrían sentido fuera del contexto de la página actual, y podrían sindicarse perfectamente. Eso incluiría las entradas de un blog, un vídeo con su transcripción, una noticia, o una parte de una historia por capítulos.

<main>

Sirve para definir el contenido principal o más destacado del DOM. Sólo puede haber una etiqueta <main>.

Dentro de esta etiqueta se debe tener elementos, tales como: cabecera de la web, menú de navegación, información de la web, pie de página, barras laterales, etc.

Nunca debe estar dentro de las etiquetas: header, footer, article, aside y nav.

<footer>

Se utiliza para contener el contenido de pie de página del sitio. El pie de página se utiliza para colocar avisos de copyright, datos de contacto a las declaraciones de accesibilidad, información de licencias y varios otros enlaces secundarios.

3.1.4 Etiquetas flexibles: box-sizing, flexible box layout, grid

box-sizing

Permite cambiar el espacio total ocupado por un elemento en pantalla, esto será calculado forzando a los navegadores a incluir en el ancho original los valores de las propiedades padding y border.

Cada vez que el área total ocupada por un elemento es calculada, el navegador obtiene el valor final por medio de la siguiente fórmula:

$$\text{tamaño} + \text{márgenes} + \text{márgenes internos} + \text{bordes}.$$

Por este motivo, si se declara la propiedad width igual a 100 pixeles, margin en 20 pixeles, padding en 10 pixeles y border en 1 pixel, el área horizontal total ocupada por el elemento será: $100+40+20+2= 162$ pixeles.

Esto significa que cada vez que declare el ancho de un elemento con la propiedad width, se deberá recordar que el área real para ubicar el elemento en pantalla será seguramente más grande.

Dependiendo de sus costumbres, a veces podría resultar útil forzar al navegador a incluir los valores de padding y border en el tamaño del elemento.

En este caso la nueva fórmula sería simplemente: tamaño + márgenes.

```
div {  
    width: 100px;  
    margin: 20px;  
    padding: 10px;  
    border: 1px solid;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

La propiedad **box-sizing** puede tomar dos valores. Por defecto es configurada como **content-box**, lo que significa que los navegadores agregarán los valores de padding y border al tamaño especificado por **width** y **height**. Usando el valor **border-box** en su lugar, este comportamiento es cambiado de modo que padding y border son incluidos dentro del elemento.

flexible box layout

Flexbox es un sistema de elementos flexibles en la que los elementos HTML se adaptan y colocan automáticamente, y es más fácil personalizar los diseños. Está especialmente diseñado para crear, mediante CSS, estructuras de una sola dimensión.

En este sistema existe un elemento padre que es el contenedor y que tendrá en su interior cada uno de los ítems flexibles y adaptables. Para activar el modo flexbox hay que utilizar sobre el elemento contenedor la propiedad **display** y especificar el valor **flex** o **inline-flex** dependiendo de cómo se quiere que se comporte el contenedor: si como un elemento en línea, o como un elemento en bloque.

Los contenedores flexibles tendrán una orientación principal específica que por defecto es horizontal (fila). Mediante la propiedad **flex-direction** se puede modificar esa dirección para que sea horizontal o en vertical. También se puede incluir el sufijo **-reverse** para indicar que coloque los ítems en orden inverso.

De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical, y viceversa.

Existe otra propiedad llamada **flex-wrap** con la que se puede especificar el comportamiento del contenedor cuando los ítems no caben en una línea y se desbordan. El valor **wrap** autoriza el salto de fila o columna cuando los elementos no caben, **nowrap** (valor por defecto) lo impide y **wrap-reverse** autoriza, pero en orden inverso.

flex-flow es la propiedad abreviada para **flex-direction** y **flex-wrap**.

grid layout

contiene funciones de diseño dirigidas a los desarrolladores de aplicaciones web. El CSS grid se puede utilizar para lograr muchos diseños diferentes. También se destaca por permitir dividir una página en áreas o regiones principales, por definir la relación en términos de tamaño, posición y capas entre partes de un control construido a partir de primitivas HTML.

Para activar la cuadrícula **grid** hay que utilizar sobre el elemento contenedor la propiedad **display** y especificar el valor **grid** o **inline-grid**. Este valor influye en cómo se comportará la cuadrícula con el contenido exterior. El primero de ellos permite que la cuadrícula aparezca encima/debajo del contenido exterior (en bloque) y el segundo de ellos permite que la cuadrícula aparezca a la izquierda/derecha (en línea) del contenido exterior.

Es posible crear cuadrículas con un tamaño explícito. Para ello, sólo se utiliza las propiedades CSS **grid-template-columns** y **grid-template-rows**, que sirven para indicar las dimensiones de cada celda de la cuadrícula, diferenciando entre columnas y filas.

Se utiliza la unidad especial de Grid CSS **fr** (fraction), que simboliza una fracción de espacio restante en el grid.

```
.grid {  
    display: grid;  
    grid-template-columns: 1fr;  
    grid-template-rows: 2fr 1fr;  
}
```

En este ejemplo, se crea una cuadrícula de 2x2, donde el tamaño de ancho de la cuadrícula se divide en dos columnas (mismo tamaño de ancho para cada una), y el tamaño de alto de la cuadrícula se divide en dos filas, donde la primera ocupará el doble (2 fr) que la segunda (1 fr):

En algunos casos, en las propiedades **grid-template-columns** y **grid-template-rows** se puede necesitar indicar las mismas cantidades un número alto de veces, resultando repetitivo y molesto. Se puede utilizar la expresión **repeat()** para indicar repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

La expresión a utilizar sería la siguiente: `repeat([número de veces], [valor o valores])`:

```
.grid {  
    display: grid;  
    grid-template-columns: 100px repeat(2, 50px) 200px;  
    grid-template-rows: repeat(2, 50px 100px);  
}
```

LABORATORIO 1

Manejando Bloques y estilos CSS en una página HTML

Diseña una página HTML donde se visualice la estructura de bloques utilizando las etiquetas <header>, <nav>, <section>, <aside>, footer> y estilos CSS, tal como se muestra en la gráfica.

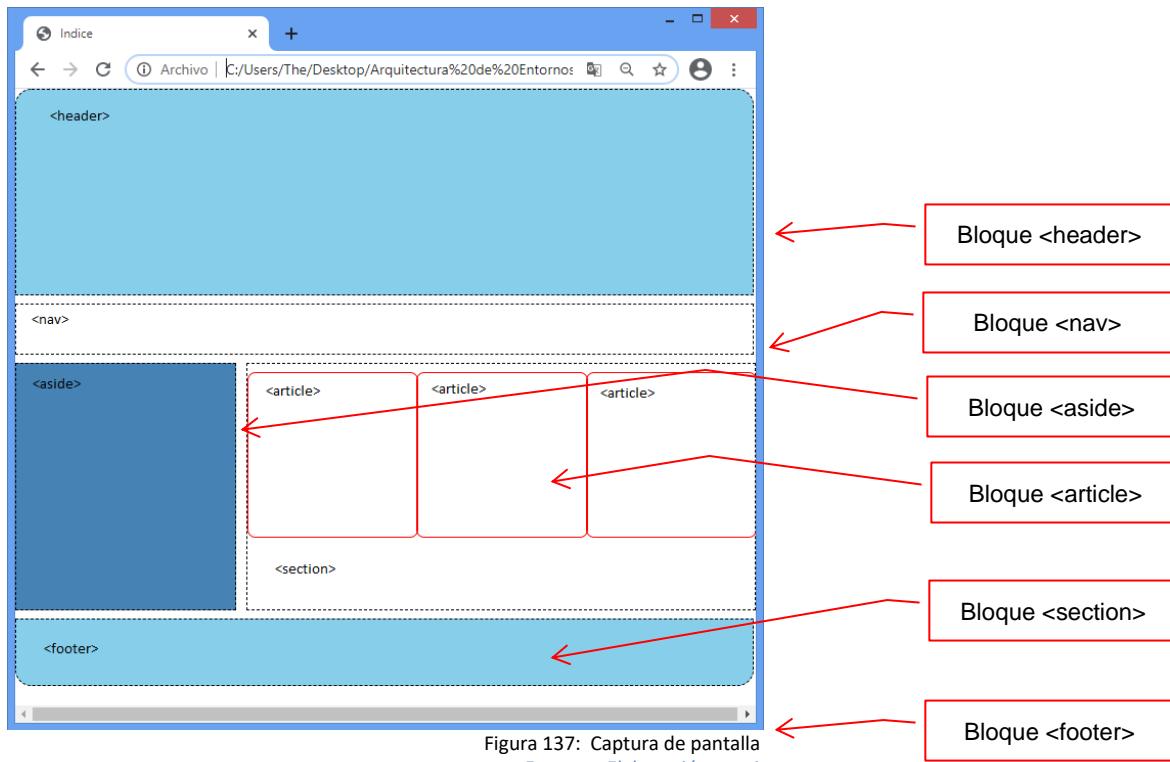


Figura 137: Captura de pantalla
Fuente. - Elaboración propia

Creando los archivos del proyecto

En la carpeta del sitio Web:

1. Agrega en la carpeta css, el archivo **estilos.css**
2. Agrega en la carpeta raíz, el archivo **índice.html**

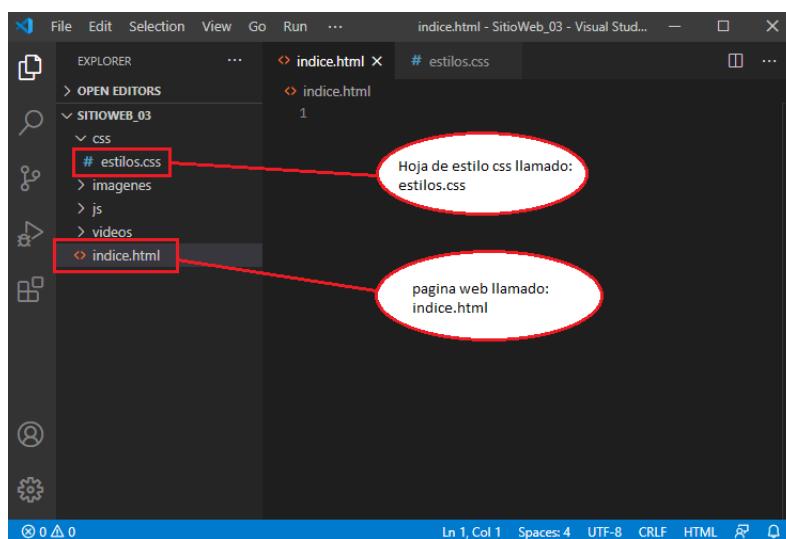


Figura 138: Captura de pantalla
Fuente. - Elaboración propia

Enlazando a la hoja de estilo

Creada la página, define, dentro del `<head>`, la etiqueta `<link>`, el cual hace una referencia al archivo CSS estilos.css, tal como se muestra.

```

indice.html x # estilos.css
indice.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Indice</title>
7
8      <link rel="stylesheet" href="css/estilos.css">
9
10 </head>
11 <body>
12
13 </body>
14 </html>

```

Figura 139: Captura de pantalla
Fuente. - Elaboración propia

Agregando reglas de estilo a las etiquetas `<main>`, `<header>` y `<nav>`

En el archivo estilos.css define las reglas de estilo para `<main>`, `<header>` y `<nav>`:

- width: ancho del bloque
- height: altura del bloque
- float: bloque flotante, left, al lado izquierdo de la página
- border: asigna borde a la caja
- background-color: color del fondo del bloque
- border-radius: radio de borde del bloque

A continuación, guarda el archivo estilos.css: **CTRL+S**.

```

indice.html # estilos.css ...
css > # estilos.css > ...
1  *{
2      margin: 0px;
3      padding: 0px;
4  }
5  main{
6      width: 100%;
7      height: auto;
8      border-radius: 20px;
9  }
10 header{
11     width: 99.5%;
12     height: 250px;
13     margin-bottom: 10px;
14     border: 1px dashed;
15     border-radius: 20px 20px 0 0;
16     background-color: skyblue;
17     float: left;
18 }
19 nav{
20     width: 99.5%;
21     height: 60px;
22     margin-bottom: 10px;
23     border: 1px dashed;
24     float: left;
25 }

```

Figura 140: Captura de pantalla
Fuente. - Elaboración propia

Agregando las etiquetas de bloque en la página

Definida las reglas en estilos.css, dentro del body define las etiquetas <main>, <header> y <nav>, tal como se muestra.

```

<html lang="en">
  <head>...</head>
  <body>
    <main>
      <header>Cabecera</header>
      <nav>Menu de Navegacion</nav>
    </main>
  </body>
</html>

```

The screenshot shows the HTML code for 'indice.html'. A red arrow points from the text 'Etiqueta de bloque <header>' to the opening tag of the <header> element at line 14. Another red arrow points from the text 'Etiqueta de bloque <nav>' to the opening tag of the <nav> element at line 15. The code editor interface includes tabs for 'indice.html' and '# estilos.css', and status bar information like 'Ln 13, Col 9' and 'UTF-8'.

Figura 141: Captura de pantalla

Fuente. - Elaboración propia

Agregando reglas de estilo a las etiquetas <section> y <aside>

En el archivo estilos.css define las reglas de estilo: <section>, <article> y <aside>:

- width y weight: ancho y altura del bloque
- float: bloque flotante, left, al lado izquierdo y right al lado derecho de la página
- border: asigna borde a la caja
- background-color: color del fondo del bloque
- display: grid definiendo 3 columnas de la misma dimensión

A continuación, guarda el archivo estilos.css: **CTRL+S**

```

aside{
  width: 29.5%;
  height: 300px;
  border: 1px dashed steelblue;
  background-color: steelblue;
  float: left;
}
section{
  width: 68.5%;
  height: 300px;
  border: 1px dashed black;
  float: right;
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
article{
  width: 100%;
  height: 200px;
  border: 1px solid red;
  border-radius: 10px;
  margin-top: 10px;
}

```

The screenshot shows the CSS code in 'estilos.css'. Three red arrows point from callout boxes to specific lines of code: one to the 'border' rule of the 'aside' selector at line 30, one to the 'display: grid' and 'grid-template-columns' rules of the 'section' selector at line 40, and one to the 'border' rule of the 'article' selector at line 45. The code editor interface includes tabs for 'indice.html' and '# estilos.css', and status bar information like 'Ln 50, Col 1' and 'UTF-8'.

Figura 142: Captura de pantalla

Fuente. - Elaboración propia

Agregando reglas de estilo a las etiquetas <footer>

En el archivo CSS, estilos.css define las reglas de estilo para <footer>.

- width y height: ancho y altura del bloque
- float: bloque flotante, left, al lado izquierdo de la página
- border: asigna borde a la caja
- background-color: color del fondo del bloque
- border-radius: radio de borde del bloque

A continuación, guarda el archivo estilos.css: **CTRL+S**

```

indice.html # estilos.css
css > # estilos.css > header
42
43 > article{...
44 }
45
46 footer{
47     width: 99.5%;
48     height: 80px;
49     margin-top: 10px;
50     border: 1px dashed;
51     border-radius: 0 20px 20px;
52     background-color: #skyblue;
53     float: left;
54 }
55
56
57
58
59

```

Figura 143: Captura de pantalla
Fuente. - Elaboración propia

Agregando las etiquetas de bloque en la página

Definida las reglas en estilos.css, dentro del body define las etiquetas, <section>, <article> y <footer>, tal como se muestra. Luego, presiona **Ctrl+S** para guardar la página.

```

indice.html # estilos.css
indice.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 > <head> ...
10 </head>
11 <body>
12   <main>
13     <header>Cabecera</header>
14     <nav>Menu de Navegacion</nav>
15
16     <aside>Barra Lateral</aside> Etiqueta <aside>: barra lateral
17
18     <section> Etiqueta <section>: sección
19       <article>articulo 1</article>
20       <article>articulo 2</article>
21       <article>articulo 3</article>
22     </section> Etiqueta <article>: articulo
23
24     <footer>pie de pagina</footer> Etiqueta <footer>: pie
25   </main>
26 </body>
27 </html>

```

Figura 144: Captura de pantalla
Fuente. - Elaboración propia

Presiona la tecla **F5** para ejecutar la página por el navegador.

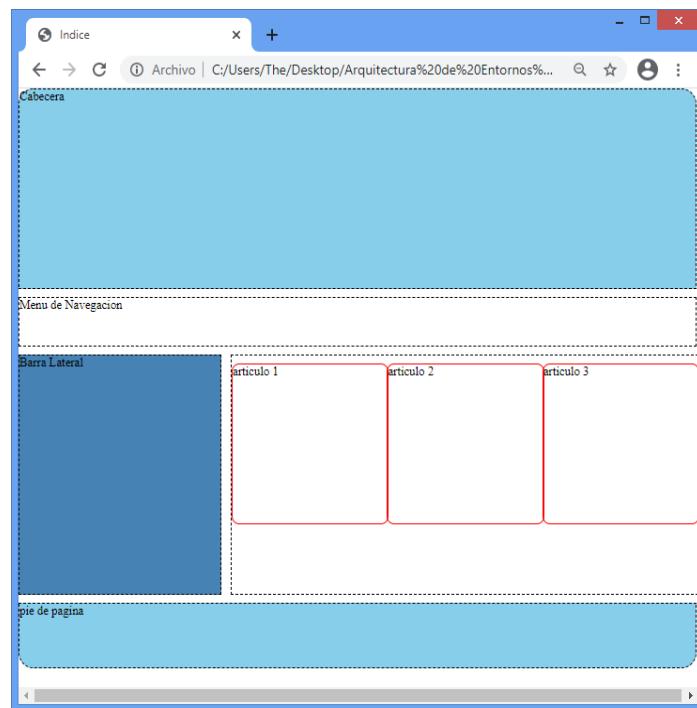


Figura 145: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 2

Se solicita diseñar una página de inicio del sitio web utilizando bloques y estilos CSS

Diseña una página de inicio HTML y utiliza las etiquetas `<header>`, `<nav>`, `<section>`, `<aside>`, `<footer>` y estilos CSS, tal como se muestra en la gráfica.

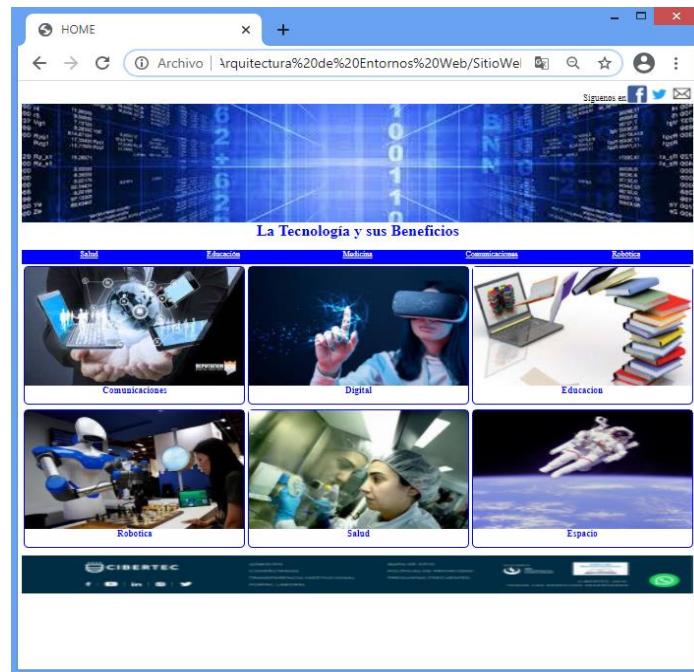


Figura 146: Captura de pantalla

Fuente. - Elaboración propia

Agregando la pagina HTML y la hoja de estilo CSS

Crea un archivo HTML llamado **home.html**. Luego, crea el archivo CSS, llamado **estilo-home.css**. Para guardar estos archivos, presiona la combinación **CTRL+S**.

```

EXPLORER ...  home.html # estilo-home.css
OPEN EDITORS
SITIOWEB_03
  css
    # estilo-home.css (resaltado)
    # estilos.css
  imagenes
  js
  videos
  home.html (resaltado)
  indice.html

```

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>

```

Ln 11, Col 8 Spaces: 4 UTF-8 CRLF HTML ⚙ ⚙

Figura 147: Captura de pantalla

Fuente. - Elaboración propia

Enlazando a la hoja de estilo

Creada la página, define, dentro del **<head>** la etiqueta **<link>**, el cual hace una referencia al archivo CSS **estilo-home.css**, tal como se muestra.

```

EXPLORER      ...
SITIOWEB_03
  <OPEN EDITORS>
    <home.html X> # estilos.css
    <css>
      # estilos.css
      # estilos.css
      <imagenes>
      <javascript>
      <videos>
    <home.html>
    <indice.html>
  
```

```

home.html X  # estilos.css
home.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>HOME</title>
7
8    <link rel="stylesheet" href="css/estilo-home.css">
9
10 </head>
11 <body>
12
13 </body>
14 </html>

```

Ln 9, Col 5 Spaces: 4 UTF-8 CRLF HTML ⚙️ ⚙️

Figura 147: Captura de pantalla
Fuente. - Elaboración propia

Agregando reglas de estilo a la cabecera de la página

En el archivo CSS, define las reglas de estilo para la cabecera:

- **#header-rs**, identificador que define las reglas para el bloque de redes sociales
- **.p-rs**, class que define las reglas para el párrafo de redes sociales
- **.rs**, clase que define el sector para las opciones de redes sociales
- **.#img-header**, id que define el sector para la imagen de la página

A continuación, guarda el archivo estilo.css: **CTRL+S**.

```

File Edit Selection View Go Run Terminal Help estilos.css - SitioWeb_03 - Visual Studio Code
EXPLORER      ...
SITIOWEB_03
  <OPEN EDITORS>
    <home.html X> # estilos.css
    <css>
      # estilos.css
      # estilos.css
      <imagenes>
      <javascript>
      <videos>
    <home.html>
    <indice.html>
  
```

```

css > # estilos.css > main
1 main{ width: 100%; height: auto; border-radius: 20px; float: left; }
2
3 header{ width: 100%; height: auto; float: left; }
4
5 #header-rs{ width: 100%; height: 40px; display: grid; grid-template-columns: 90% 1fr 1fr 1fr; float: left; }
6
7 .p-rs{ text-align: right; font-size: 18px; }
8
9 .rs{ width: 100%; height: 40px; border-radius: 10px; }
10
11 #img-header{ width: 100%; height: 250px; float: left; }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

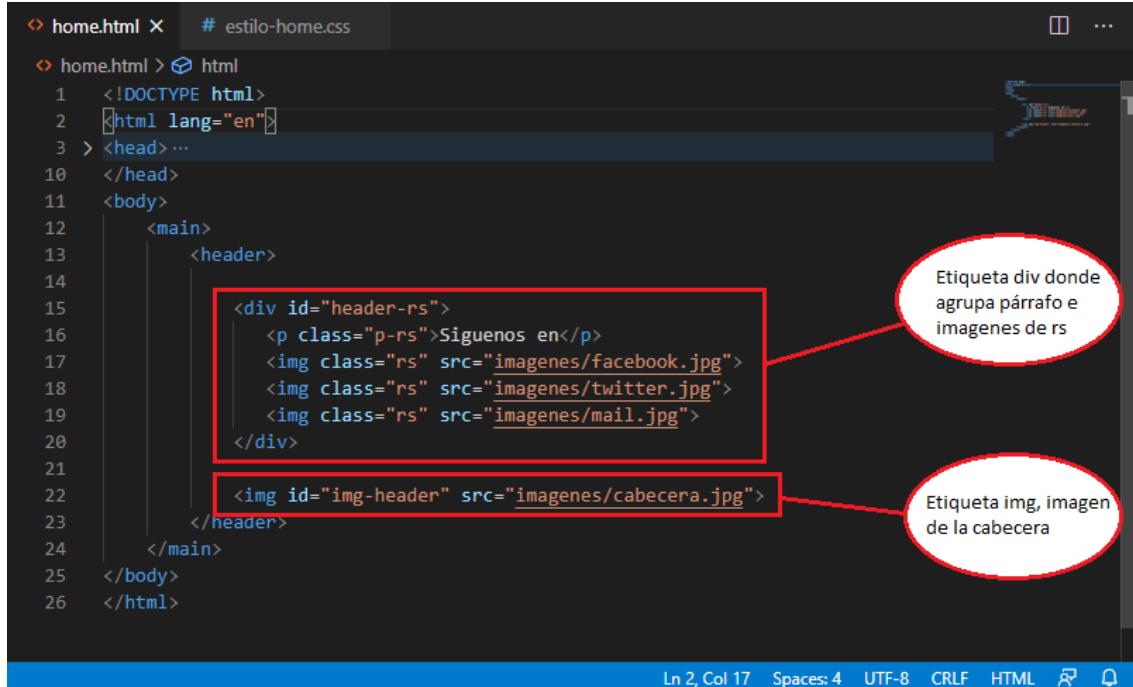
```

Ln 1, Col 6 Spaces: 4 UTF-8 CRLF CSS ⚙️ ⚙️

Figura 148: Captura de pantalla
Fuente. - Elaboración propia

Agregando la etiqueta <header> y su contenido en la página HTML

Definida las reglas en estilos.css, dentro del body define la etiqueta <header> agregando elementos de bloque <div id=header-rs>, para agrupar imágenes de redes sociales:, y la imagen de la cabecera .



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head> ...
10 </head>
11 <body>
12 <main>
13   <header>
14     <div id="header-rs">
15       <p class="p-rs">Siguenos en</p>
16       
17       
18       
19     </div>
20
21   
22 </header>
23 </main>
24 </body>
25 </html>

```

Etiqueta div donde agrupa párrafo e imágenes de rs

Etiqueta img, imagen de la cabecera

Figura 149: Captura de pantalla

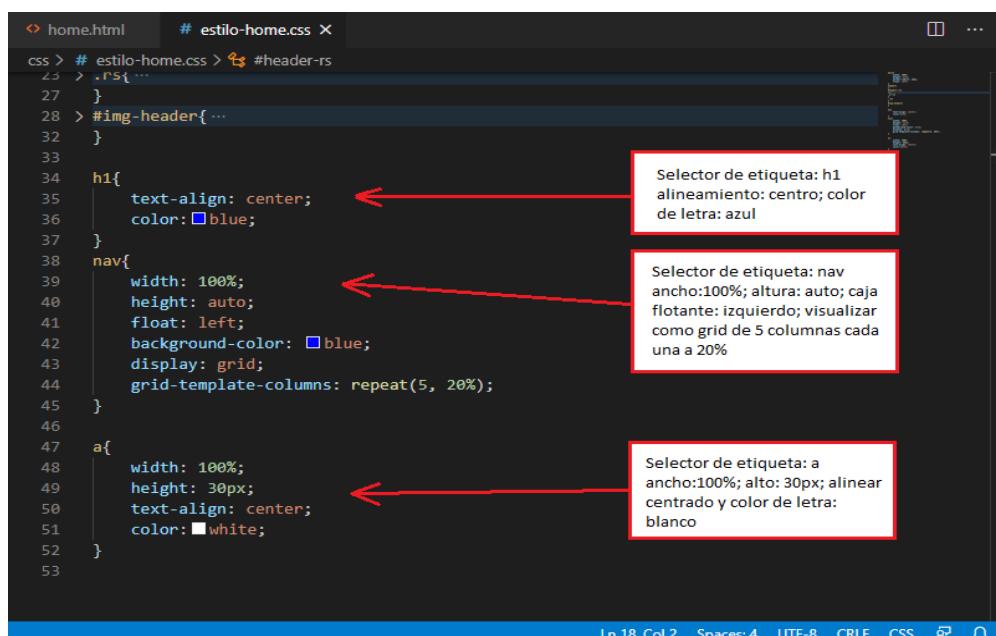
Fuente. - Elaboración propia

Agregando las reglas en el Menú de Navegación

En el archivo CSS, define las reglas de estilo para el menú de navegación:

- h1, etiqueta que define las reglas para <h1>
- nav, etiqueta que define las reglas para la etiqueta <nav>
- a, regla para los enlaces en el menú de navegación <a>

A continuación, guardar el archivo estilo-home.css: **CTRL+S**.



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head> ...
4   <meta charset="UTF-8" ...>
5   <link href="estilo-home.css" rel="stylesheet">
6 <body>
7   <header> ...
8     <div id="header-rs">
9       <p class="p-rs">Siguenos en</p>
10      
11      
12      
13    </div>
14
15    
16 </header>
17 <main>
18   <h1>Portafolio Web </h1>
19   <nav>
20     <ul>
21       <li><a href="#">Home</a></li>
22       <li><a href="#">Proyectos</a></li>
23       <li><a href="#">Sobre mí</a></li>
24     </ul>
25   </nav>
26
27   <div id="contenedor">
28     <div id="columna-izquierda">
29       <img alt="Logo de la Escuela de Tecnologías de la Información" ...>
30       <ul>
31         <li><a href="#">Enseñanza</a></li>
32         <li><a href="#">Investigación</a></li>
33         <li><a href="#">Extensión</a></li>
34       </ul>
35     </div>
36     <div id="columna-derecha">
37       <h2>Proyectos Recientes</h2>
38       <ul>
39         <li><a href="#">Proyecto 1</a></li>
40         <li><a href="#">Proyecto 2</a></li>
41         <li><a href="#">Proyecto 3</a></li>
42       </ul>
43     </div>
44   </div>
45
46 </main>
47 <footer> ...
48 </body>
49 </html>

```

Selector de etiqueta: h1
alineamiento: centro; color de letra: azul

Selector de etiqueta: nav
ancho:100%; altura: auto; caja flotante: izquierdo; visualizar como grid de 5 columnas cada una a 20%

Selector de etiqueta: a
ancho:100%; alto: 30px; alinear centrado y color de letra: blanco

Figura 150: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <nav> y sus enlaces en la página HTML

Definida las reglas en estilo-home.css, define la etiqueta <nav> agregando enlaces <a href>, tal como se muestra.

```

11  <body>
12    <main>
13      <header> ...
14      </header>
15      <nav>
16        <a href="#">Salud</a>
17        <a href="#">Educación</a>
18        <a href="#">Medicina</a>
19        <a href="#">Comunicaciones</a>
20        <a href="#">Robótica</a>
21      </nav>
22    </main>
23  </body>
24 </html>

```

Figura 151: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en el cuerpo de la página <section>, <article>

En el archivo CSS defina las reglas para el cuerpo de la página

- section, regla para el bloque <section>
- article, regla para el bloque <article>
- .img-article, regla para la etiqueta de cada article
- .h3-article, regla para la etiqueta <h3> de cada article

```

54  section{
55    width: 100%;
56    height: auto;
57    float: left;
58    display: grid;
59    grid-template-rows: 2;
60    grid-template-columns: 1fr 1fr 1fr;
61  }
62
63  article{
64    width: 98%;
65    height: auto;
66    float: left;
67    margin: 1%;
68    border-radius: 10px;
69    border: 1px solid blue;
70  }
71
72  .img-article{
73    width: 100%;
74    height: 250px;
75    float: left;
76  }
77
78  .h3-article{
79    text-align: center;
80    color: blue;
81  }

```

Figura 152: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <section> y <articulo> para la página HTML

Definida las reglas en estilo-home.css, define la etiqueta <section> y agrega bloques <article>. Dentro de cada bloque añade un elemento con selector de class img-article y un elemento <h3> con selector de clase h3-article.

```

home.html x # estilos-home.css
home.html > html > body > main > section > article > h3.h3-article
33
34 <section>
35 <article>
36 
37 <h3 class="h3-article">Comunicaciones</h3>
38 </article>
39 <article>
40 
41 <h3 class="h3-article">Digital</h3>
42 </article>
43 <article>
44 
45 <h3 class="h3-article">Educacion</h3>
46 </article>
47 <article>
48 
49 <h3 class="h3-article">Robotica</h3>
50 </article>
51 <article>
52 
53 <h3 class="h3-article">Salud</h3>
54 </article>
55 <article>
56 
57 <h3 class="h3-article">Espacio</h3>
58 </article>
59 </section>
60 </main>
61 </body>
62 </html>

```

Ln 57, Col 47 Spaces: 4 UTF-8 CRLF HTML ⚙️

Figura 153: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas al pie de página

En el archivo CSS, define las reglas para el pie de página.

- footer, etiqueta para el bloque <footer>
- #img-footer, identificador para la del footer

A continuación, guarda el archivo estilos-home.css: **CTRL+ S**.

```

css > # estilos-home.css > #img-footer
77
78 > .h3-article{ ...
81 }
82
83 footer{
84 width: 100%;
85 height: auto;
86 float: left;
87 margin-top: 10px;
88 }
89
90 #img-footer{
91 width: 100%;
92 height: 80px;
93 float: left;
94 }

```

Ln 93, Col 16 (4 selected) Spaces: 4 UTF-8 CRLF CSS ⚙️

Figura 154: Captura de pantalla

Fuente. - Elaboración propia

Agregando el pie de página <footer>

Definida las reglas en style.css, define la etiqueta <footer> agregando su , tal como se muestra. Luego, presiona la tecla F5 para visualizar el diseño.

```

home.html X # estilo-home.css
home.html > html > body > main > footer
33
34 >   <section>...
59   </section>
60
61   <footer>
62
63     
64
65   </footer>
66 </main>
67 </body>
68 </html>

```

Figura 155: Captura de pantalla
Fuente. - Elaboración propia

Para ejecutar a través del navegador, primero guarda la página: **CTRL+S** y presiona la tecla **F5** para ejecutar la página con el navegador.

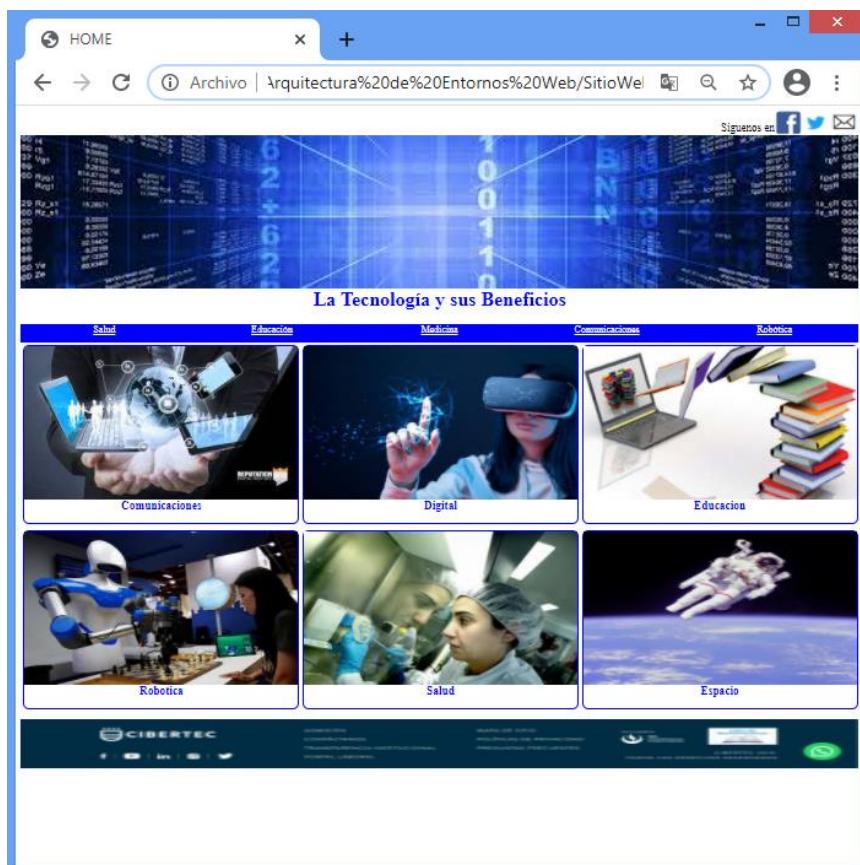


Figura 156: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 3

Se solicita diseñar una página acerca de tecnología y educación. Explicando, en forma breve, las aportaciones de la tecnología en diversas áreas. Utiliza bloques y estilos CSS.

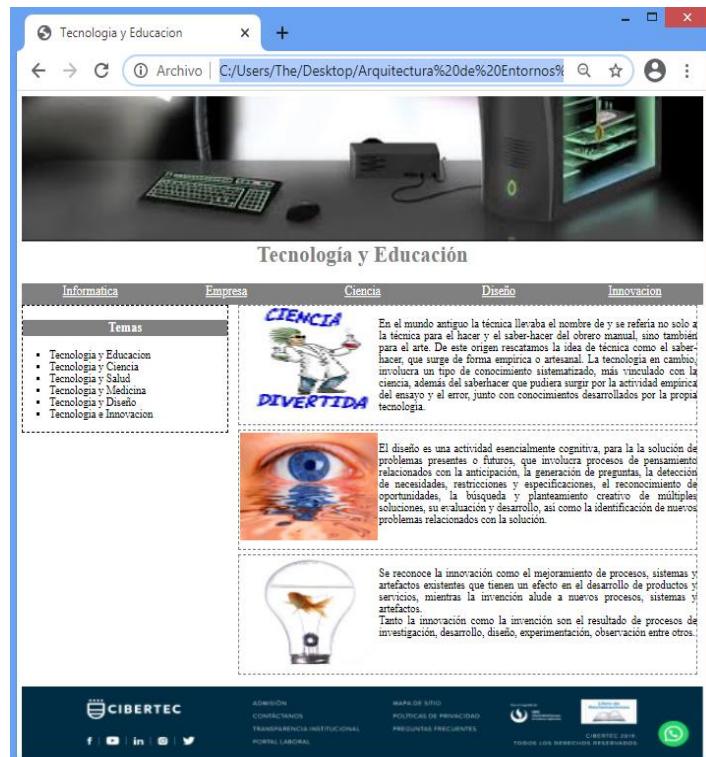


Figura 157: Captura de pantalla

Fuente. - Elaboración propia

Agregando la página HTML

Crea un archivo HTML, el cual no tendrá diseño (ninguno). Guarda la página con el nombre de **tecnología-educacion.html**, en la carpeta raíz del sitio.

```

EXPLORER ... < tecnologia-educacion.html # estilo-educacion.css
OPEN EDITORS ...
SITIOWEB_03 ...
css ...
# estilo-educacion.css
# estilo-home.css
# estilos.css
imagenes
js
videos
home.html
indice.html
tecnologia-educacion.html

# estilo-educacion.css
# estilo-home.css
# estilos.css
archivo de estilos css:
estilo-educacion.css

tecnologia-educacion.html
archivo html:
tecnologia-educacion.html

```

Figura 158: Captura de pantalla

Fuente. - Elaboración propia

Enlazando a la hoja de estilo

Creada la página, define dentro del **<head>** la etiqueta **<link>**, el cual hace una referencia al archivo **CSS estilo-educacion.css**.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Tecnología y Educación</title>
7
8      <link rel="stylesheet" href="css/estilo-educacion.css">
9
10 </head>
11 <body>
12
13 </body>
14 </html>

```

Ln 3, Col 7 Spaces: 4 UTF-8 CRLF HTML ⚡ 🔍

Figura 159: Captura de pantalla
Fuente. - Elaboración propia

Agregando reglas de estilo a la cabecera de la página

En el archivo CSS, defina las reglas de estilo para la cabecera:

- header, identificador que define las reglas para la etiqueta <header>
- #img-header, identificador que define las reglas a la del header
- #h1-header, identificador que define las reglas al <h1> del header

A continuación, guarda el archivo estilo-educacion.css: **CTRL+S**.

```

1  main{
2      width: 100%;
3      height: auto;
4      float: left;
5  }
6
7  header{
8      width: 100%;
9      height: auto;
10     float: left;
11 }
12
13 #img-header{
14     width: 100%;
15     height: auto;
16     float: left;
17 }
18
19 #h1-header{
20     text-align: center;
21     color: gray;
22 }

```

Ln 21, Col 16 Spaces: 4 UTF-8 CRLF CSS ⚡ 🔍

Figura 160: Captura de pantalla
Fuente. - Elaboración propia

Agregando la etiqueta <header> y su contenido en la página HTML

Definida las reglas en el CSS, dentro del body, define la etiqueta <header>. Agrega dos elementos al bloque:

- imagen de la cabecera de la página
- <h1 id="h1-header"> título de la página

```

    tecnologia-educacion.html x # estilo-educacion.css
    tecnologia-educacion.html ...
    !DOCTYPE html
    <html lang="en">
    <head> ...
    </head>
    <body>
        <main>
            <header>
                
                <h1 id="h1-header">Tecnología y Educación</h1>
            </header>
        </main>
    </body>
    </html>

```

Figura 161: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en el Menú de Navegación

En el archivo CSS, define las reglas de estilo para el menú de navegación:

- nav, identificador que define las reglas para la etiqueta <nav>
- .a-nav, regla para los enlaces en el menú de navegación <a>
- .a-nav:hover, regla para el evento hover de la etiqueta <a>

A continuación, guarda el archivo estilo-educacion.css: **CTRL+S**.

```

css > # estilo-educacion.css x
css > # estilo-educacion.css > header
nav{
    width: 100%;
    height: auto;
    float: left;
    background-color: #gray;
    display: grid;
    grid-template-columns: repeat(5, 20%);
}
.a-nav{
    width: 100%;
    height: 30px;
    font-size: 18px;
    text-align: center;
    color: white;
}
.a-nav:hover{
    background-color: white;
    color: gray;
}

```

Figura 162: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <nav> y sus enlaces en la página HTML

Definida las reglas en estilo-educacion.css, defina la etiqueta <nav> agregando enlaces <a href>.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 > <head> ...
10 </head>
11 <body>
12 >   <main>
13 >     <header> ...
16 >     <nav>
18 >       <a href="#" class="a-nav" >Informatica</a>
19 >       <a href="#" class="a-nav" >Empresa</a>
20 >       <a href="#" class="a-nav" >Ciencia</a>
21 >       <a href="#" class="a-nav" >Diseño</a>
22 >       <a href="#" class="a-nav" >Innovacion</a>
23 >     </nav>
24 >   </main>
25 > </body>
26 > </html>

```

Ln 2, Col 17 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍

Figura 163: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en la barra lateral <aside>

En el archivo CSS, defina las reglas para barra lateral.

- aside, regla para el bloque <aside>
- .h3-aside, regla para la etiqueta <h3> del bloque <aside>
- .ítem, regla para la lista dentro de cada bloque article

A continuación, guarda el archivo estilo.css: **CTRL+S**.

```

45 aside{
46   width:30%;
47   height:auto;
48   float: left;
49   border: 1px dashed;
50 }
52
53 .h3-aside{
54   text-align: center;
55   background-color: gray;
56   color: white;
57 }
58
59 .ítem{
60   list-style-type: square;
61   margin: 0px;
62 }

```

Ln 22, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🔍

Figura 164: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <aside> para la página HTML

Definida las reglas en estilo-educacion.css, define la etiqueta <aside> para mostrar la lista de los temas en tecnología. Presiona la tecla **F5** para visualizar el diseño.

```

17   <nav> ...
18   </nav>
19
20   <aside>
21     <h3 class="h3-aside">Temas</h3>
22     <ul>
23       <li class="item">Tecnología y Educación</li>
24       <li class="item">Tecnología y Ciencia</li>
25       <li class="item">Tecnología y Salud</li>
26       <li class="item">Tecnología y Medicina</li>
27       <li class="item">Tecnología y Diseño</li>
28       <li class="item">Tecnología e Innovación</li>
29     </ul>
30   </aside>
31
32   </main>
33
34 </body>
35 </html>

```

Figura 165: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en el cuerpo de la página <section>

En el archivo CSS, style-productos.css define las reglas para el cuerpo de la página:

- `#cuerpo`, regla para el bloque `<section>`
- `.articulo`, regla para el bloque `<article>`
- `.articulo img`, regla para las imágenes dentro de cada bloque `article`

A continuación, guarda el archivo estilo.css: **CTRL+S**

```

1 section{
2   width: 69%;
3   height: auto;
4   float: right;
5   display: grid;
6   grid-template-columns: 1fr;
7   grid-template-rows:1fr 1fr 1fr;
8 }
9
10 article{
11   width: 97.5%;
12   height: auto;
13   float: left;
14   margin: 0 1% 1% 1%;
15   border:1px dashed gray;
16 }
17
18 .img-article{
19   width: 30%;
20   height: 150px;
21   float: left;
22   margin: 2px;
23 }
24
25 .p-article{
26   text-align: justify;
27 }

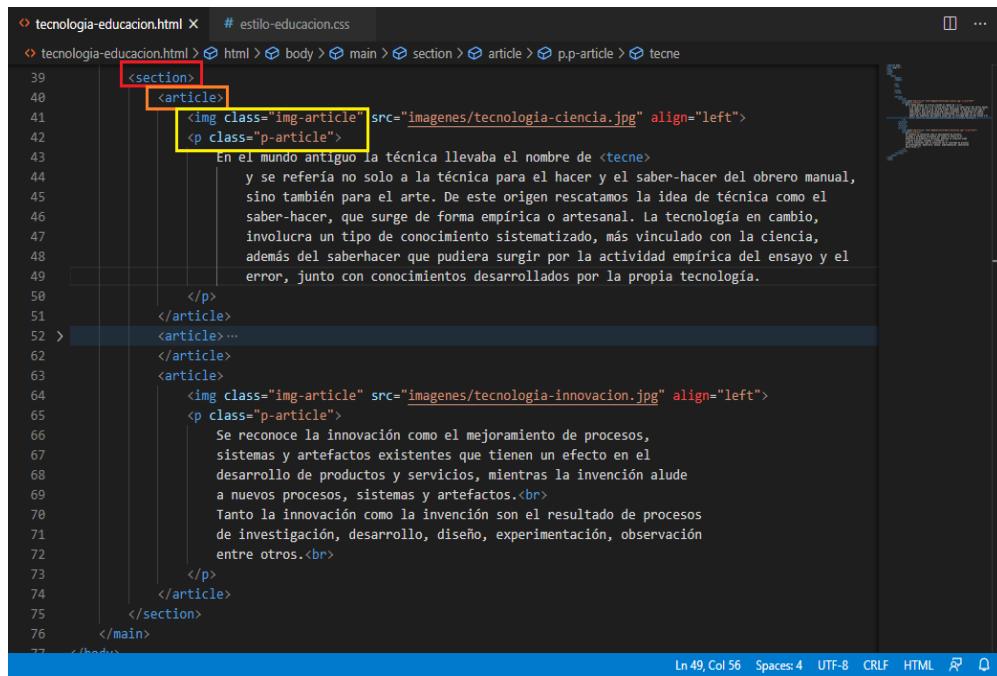
```

Figura 166: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <section> y <articulo> para la página HTML

Definida las reglas en estilo-educacion.css, agrega la etiqueta <section>. Agrega bloques <article> que incluya y <p> para definir la tecnología en sus diferentes ámbitos. Luego, presiona la tecla **F5**.



```

tecnologia-educacion.html X # estilo-educacion.css
tecnologia-educacion.html > html > body > main > section > article > p.p-article > tecne
39 <section>
40   <article>
41     
42     <p class="p-article">
43       En el mundo antiguo la técnica llevaba el nombre de <tecne>
44       y se refería no solo a la técnica para el hacer y el saber-hacer del obrero manual,
45       sino también para el arte. De este origen rescatamos la idea de técnica como el
46       saber-hacer, que surge de forma empírica o artesanal. La tecnología en cambio,
47       involucra un tipo de conocimiento sistematizado, más vinculado con la ciencia,
48       además del saber-hacer que pudiera surgir por la actividad empírica del ensayo y el
49       error, junto con conocimientos desarrollados por la propia tecnología.
50   </p>
51 </article>
52 > <article>...
53 </article>
54 <article>
55   
56   <p class="p-article">
57     Se reconoce la innovación como el mejoramiento de procesos,
58     sistemas y artefactos existentes que tienen un efecto en el
59     desarrollo de productos y servicios, mientras la invención alude
60     a nuevos procesos, sistemas y artefactos.<br>
61     Tanto la innovación como la invención son el resultado de procesos
62     de investigación, desarrollo, diseño, experimentación, observación
63     entre otros.<br>
64   </p>
65 </article>
66 </section>
67 </main>
68 </body>

```

Figura 167: Captura de pantalla

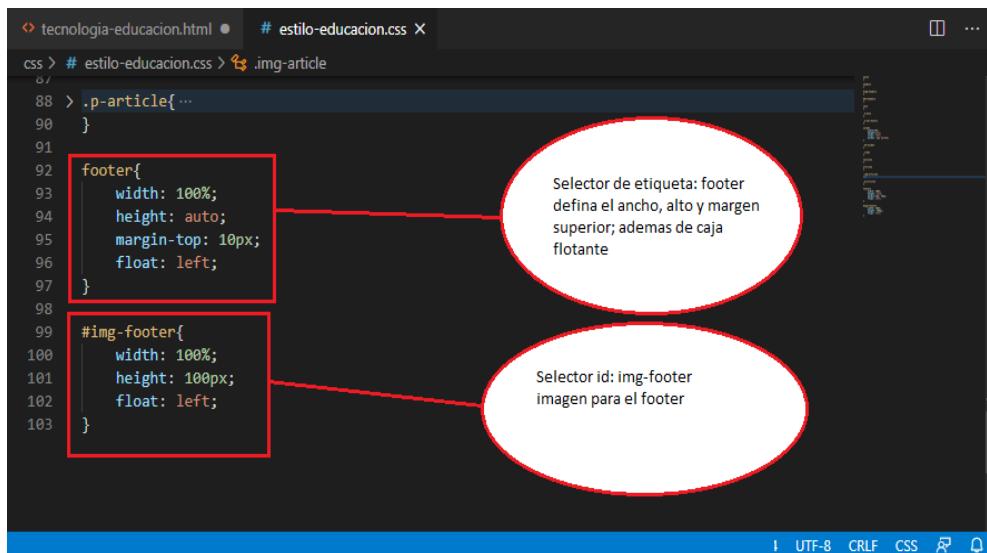
Fuente. - Elaboración propia

Agregando las reglas al pie de página

En el archivo CSS define las reglas para el pie de página.

- footer, reglas para el bloque <footer>
- #img-footer, regla para el elemento del footer

A continuación, guarda el archivo estilo.css: **CTRL+S**.



```

css > # estilo-educacion.css > .img-article
8/
88 > .p-article{ ...
89 }
90 }

91 footer{
92   width: 100%;
93   height: auto;
94   margin-top: 10px;
95   float: left;
96 }
97

98 #img-footer{
99   width: 100%;
100  height: 100px;
101  float: left;
102 }
103

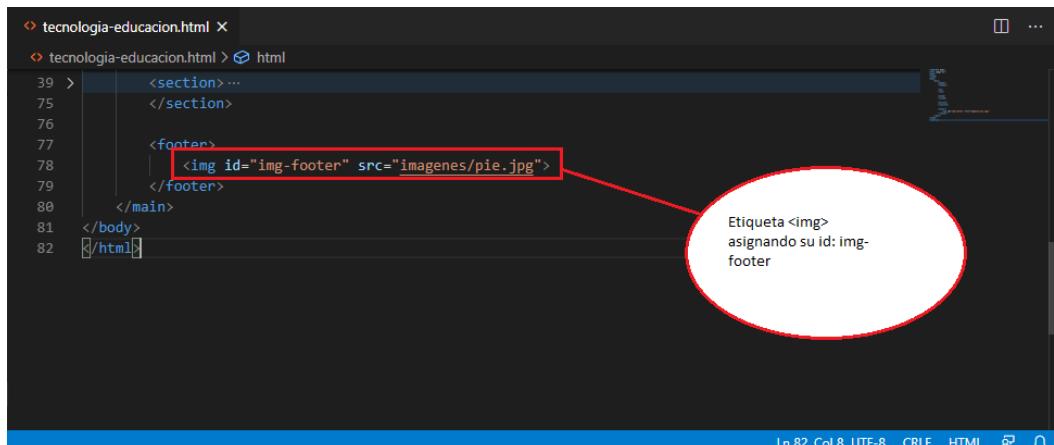
```

Figura 168: Captura de pantalla

Fuente. - Elaboración propia

Agregando el pie de página <footer>

Definida las reglas en estilo-educacion.css, define la etiqueta <footer> agregando su imagen, tal como se muestra.



```

    <!-- tecnologia-educacion.html -->
    <!-- tecnologia-educacion.html > html
    39   <section> ...
    40     </section>
    41
    42   <footer>
    43     |   
    44   </footer>
    45
    46   </main>
    47
    48 </body>
    49 </html>
  
```

Eiqueta
asignando su id: img-footer

Figura 169: Captura de pantalla
Fuente. - Elaboración propia

Para ejecutar a través del navegador, primero guarda la página: **CTRL+S** y presiona la tecla **F5** para ejecutar la página con el navegador.

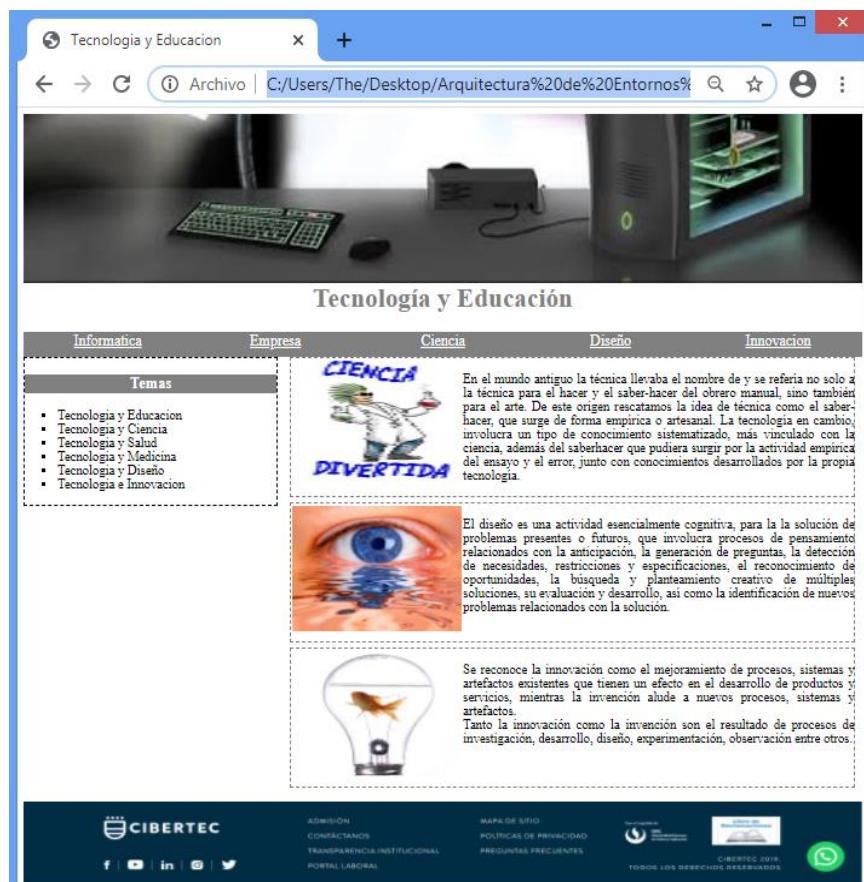


Figura 170: Captura de pantalla
Fuente. - Elaboración propia

Resumen

1. El modelo de cajas es el comportamiento de estilo CSS que hace que todos los elementos de una página se representen mediante cajas rectangulares. Este modelo es la característica más importante del lenguaje de estilos CSS, pues condiciona el diseño de todas las páginas web.
2. HTML5 aporta una serie de etiquetas nuevas que permiten mejorar la semántica de nuestra página. Los elementos de HTML5 nuevos: header, nav, section, article, aside, footer, hgroup, figure.
3. Las partes que compone un bloque y su orden de visualización desde el punto de vista del usuario son: content se trata del contenido HTML, width, controla el ancho de la caja, height, controla la altura de la caja, border, línea que encierra completamente el contenido y su relleno, margin, separación opcional entre la caja y el resto de cajas adyacentes.
4. Las nuevas propiedades CSS3 son extremadamente poderosas y deben ser estudiadas una por una, pero para facilitar su aprendizaje, están son: padding, espacio libre opcional existente entre el contenido y el borde, background-image, imagen que se muestra por detrás del contenido y el espacio de relleno; background-color, color que se muestra por detrás del contenido y el espacio de relleno; border-radius, efecto de las esquinas redondeadas; float, establece el posicionamiento flotante de la caja; clear, modifica el comportamiento de las cajas flotantes.
5. Para aplicar efectivamente estas propiedades a sus documentos, debes declararla con los correspondientes prefijos, para los navegadores más comunes son los siguientes: -moz- para Firefox; -webkit- para Safari y Chrome; -o- para Opera; -khtml- para Konqueror; -ms- para Internet Explorer; -chrome- específico para Google Chrome
6. Las propiedades **display** y **visibility** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.
7. La propiedad **box-sizing** permite cambiar cómo el espacio total ocupado por un elemento en pantalla será calculado forzando a los navegadores a incluir en el ancho original los valores de las propiedades padding y border. La propiedad **box-sizing** puede tomar dos valores. Por defecto es configurada como **content-box**, lo que significa que los navegadores agregarán los valores de padding y border al tamaño especificado por **width** y **height**.

Recursos

Pueden revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- http://librosweb.es/css/capitulo_4.html
- <http://www.desarrolloweb.com/articulos/aprende-html5-5-minutos.html>
- <http://blog.ikhuerta.com/css3-displaybox-maquetacion-con-el-modelo-de-caja-flexible>
- <https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

3.2 CREACIÓN DE MENÚ

3.2.1 Diseño de Menú

HTML5 y CSS3 son dos recursos esenciales para diseñadores y desarrolladores en la actualidad si lo que se busca es crear sitios web funcionales, así como estéticamente hermosos. Estos lenguajes permiten crear páginas web con efectos realmente alucinantes e innovadores y además son muy simples de usar.



Figura 171: Menú

Fuente. - Tomado de <http://www.cssblog.es/menu-con-varios-niveles-mediante-css/>

Otro elemento bastante típico y común en muchas páginas web es la zona de navegación, formada por enlaces a las zonas principales de la web, o por un menú de navegación, desplegable de enlaces, etc. Por lo visto Html5 interpreta que es una zona que deberíamos identificar, por ello, ha creado la nueva etiqueta `<nav> </nav>`. En Internet Explorer, por desgracia este elemento `<nav>` no es compatible, sin embargo, hay una solución sencilla que se puede utilizar para corregir el problema y que el menú se despliegue sin errores de visualización.

No todos los enlaces de una página web han de pertenecer obligatoriamente a un nav. Los enlaces de políticas, de datos de contacto, mapas del sitio, copyrights, enlaces a secciones secundarias etc, que aparecen normalmente en un pie de página no es necesario que los rodeemos con la etiqueta `<nav>`, pues la etiqueta `<footer>` ya recoge esos aspectos. Ésta etiqueta es más bien para enlaces que favorezcan la navegación por la web, que dirijan a zonas importantes.

El por qué HTML5 añade esta etiqueta es sencillo. Cada vez es más habitual que nuestras páginas aparezcan no solo en los navegadores que usamos en el ordenador, sino también en dispositivos móviles, teléfonos, iPads, etc. Al usar esta etiqueta el agente de usuario (llamado también navegador) podría dar la opción al usuario de esconder el menú de navegación para no molestar, para tener más espacio disponible para ver el contenido importante. También podría dar la opción de hacerlo aparecer cuando el usuario quiera cambiar de sección. Esto solo sería posible si se sabe dónde empieza y dónde termina ese menú de navegación y eso es precisamente lo que se consigue con la etiqueta `<nav>`.

En la norma donde se dictan todas estas reglas del HTML5 no se indica que no pueda existir más de un bloque del tipo `<nav>` dentro de una misma página. No obstante, en algunos sitios se recomienda que solo exista uno.

3.2.2 Creando un menú utilizando estilo a los elementos

Para diseñar un menú de navegación se utiliza listas, porque un menú básicamente es una lista, no es un párrafo, ni un título, es una lista de vínculos.

Para cambiar el aspecto de las listas, CSS permite cambiar el aspecto de la lista, para convertirla en el bonito menú que se desea, sin que deje de ser lista. Pero primero, observa en qué consisten los pre-formatos que trae de por sí una lista.

Las listas se renderizan en el navegador con ciertas características, como el margen/relleno (los cuales son interpretados de diferente manera en los distintos navegadores como opera, Chrome, Internet, entre otros) de la lista completa, el margen / relleno (idem) de cada ítem de la lista y las viñetas, además de un salto de línea para cada ítem por ser elementos de bloque.

LABORATORIO 1

Se solicita diseñar una página de inicio, donde se implemente un menú de navegación utilizando estilo CSS.

Creando el Sitio Web

Para este laboratorio, crea el **SitioWeb4**, tal como se muestra. Desde allí, agrega el archivo de estilo: **estilo-home.css** y la página web: **home.html**

```

EXPLORER      ...
OPEN EDITORS
SITIOWEB_04
  > CSS
    # estilos-home.css
  > imagenes
  > js
  > videos
  > home.html

home.html X  # estilos-home.css
home.html > ...
1  !DOCTYPE html
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Home</title>
7  </head>
8  <body>
9
10 </body>
11 </html>

```

Figura 172: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas de estilo a <main>, <header>

Para iniciar con el diseño de la página, primero define las reglas de estilo a los selectores **<main>** y **<header>**, y define el selector **id: img-header** para la imagen del header.

```

css > # estilos-home.css > #img-header
1  *{
2    margin: 0px;
3    padding: 0px;
4  }
5
6  main{
7    width: 100%;
8    height: auto;
9    float: left;
10   border-radius: 20px 20px 0 0;
11 }
12
13 header{
14   width: 100%;
15   height: auto;
16   float: left;
17   border-radius: 20px 20px 0 0;
18 }
19
20 #img-header{
21   width: 100%;
22   height: 250px;
23   float: left;
24   border-radius: 20px 20px 0 0;
25 }

```

Figura 173: Captura de pantalla
Fuente. - Elaboración propia

Agregando los elementos a la página

Definida las reglas en el CSS, dentro del body, define la etiqueta `<header>`. Luego, agrega dos elementos al bloque:

- `` imagen de la cabecera de la página
- `<h1 id="h1-header">` título de la página

![Captura de pantalla de un editor de código (Visual Studio Code) mostrando el archivo home.html. Se resaltan secciones del código con óvalos rojos. Un óvalo rodea la línea 7: <link rel=](imagenes/cabecera-tecnologia.jpg)

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Home</title>
7      <link rel="stylesheet" href="css/estilo-home.css"> enlazando a la hoja de estilo: estilo-home.css
8  </head>
9  <body>
10     <main>
11         <header>
12             
13             <h1 id="h1-header">La Tecnología</h1>
14         </header>
15     </main>
16
17 </body>
18 </html>
19

```

Figura 174: Captura de pantalla

Fuente. - Elaboración propia

Creando un menú de navegación

A continuación, crea un menú con HTML en el bloque `<nav>`

1. Crea la lista no ordenada en la página HTML, tal como se muestra.

```

9  <body>
10     <main>
11
12     <header>...
13     </header>
14
15     <nav>
16         <ul>
17             <li><a href="#">Inicio</a></li>
18             <li><a href="#">Tipos de Tecnología</a></li>
19             <li><a href="#">Clasificación</a></li>
20             <li><a href="#">Más Visitados</a></li>
21             <li><a href="#">Contactenos</a></li>
22         </ul>
23     </nav>
24
25     </main>
26
27 </body>
28 </html>
29

```

Figura 175: Captura de pantalla

Fuente. - Elaboración propia

Guardar el archivo como pagina HTML. Abre el documento con el navegador Chrome, donde se visualiza la página.



Figura 176: Captura de pantalla
Fuente. - Elaboración propia

2. En el archivo CSS, define las etiquetas `<nav>`, ``, ``

- `<nav>`, regla para el bloque del menú de navegación
- ``, retira las viñetas a través de la propiedad `list-style: none`
- ``, para que el menú sea horizontal, agrega la propiedad `float`, define el ancho y altura.

```

home.html # estilo-home.css •
css > # estilo-home.css > ...
30
31 nav{
32   width: 100%;
33   height: auto;
34   float: left;
35 }
36
37 ul{
38   list-style-type: none;
39 }
40
41 ul li{
42   float: left;
43   text-align: center;
44   width: 20%;
45   height: 30px;
46 }
47
48

```

Ln 51, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🔍

Figura 177: Captura de pantalla
Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome y podrás visualizar la página.

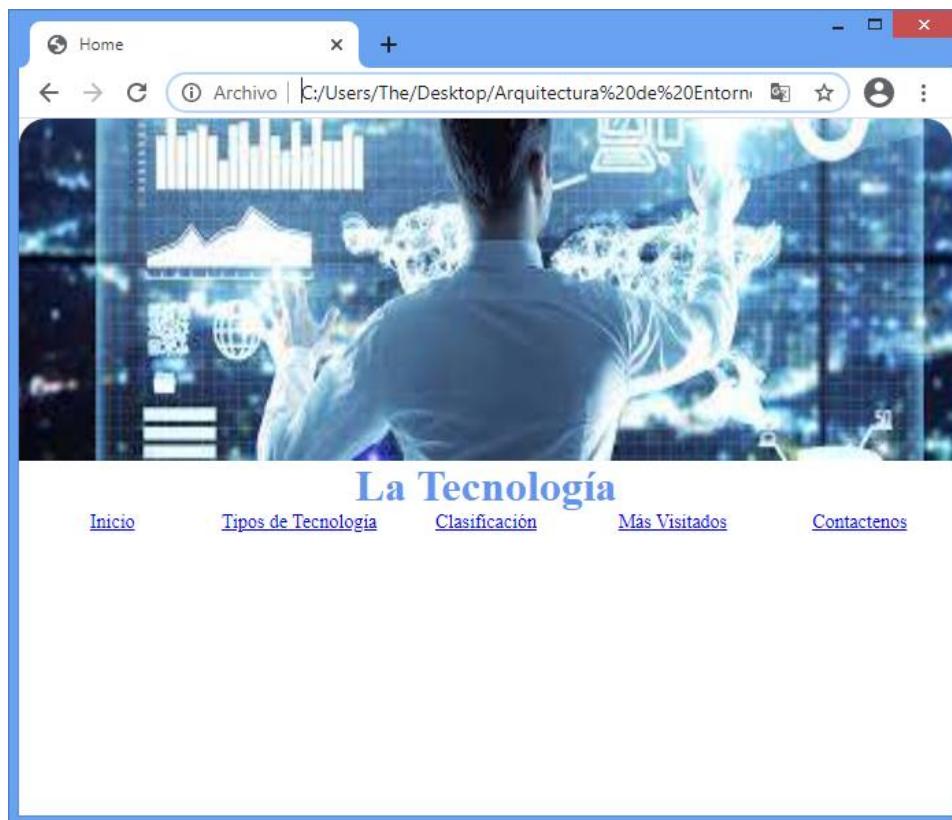


Figura 178: Captura de pantalla
Fuente. - Elaboración propia

3. A continuación, define estilos directamente sobre los vínculos, `<a>` asignando: bloque, tamaño, tipo de letra, colores, decoración, etc. Para seleccionar los vínculos dentro de los elementos de lista, encadena los nombres de las etiquetas en el selector CSS como se ha realizado con los ``:

A screenshot of a code editor showing a CSS file named '# estilo-home.css'. The code is as follows:

```
css > # estilo-home.css > ...
40
41 > ul li{...
42 }
43
44 ul li a{
45   display: block;
46   width: 100%;
47   height: 30px;
48   padding-top: 10px;
49   text-decoration: none;
50   color: white;
51   background-color: cornflowerblue;
52 }
53
54
55
56
57 }
```

A red box highlights the entire block of CSS for the `ul li a` selector. A red circle with a line pointing to it contains the text: 'Selector de etiqueta `<a>`, define las reglas para el hipervínculo'.

Figura 179: Captura de pantalla
Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome y podrás visualizar la página.



Figura 180: Captura de pantalla
Fuente. - Elaboración propia

Por último, define propiedades del evento hover de la etiqueta `<a>`, para cambiar el color de letra y de fondo al acercar el cursor en el vínculo.

```
home.html      # estilo-home.css X
css > # estilo-home.css > ul li a:hover
40
41 > ul li{ ...
47 }
48
49 ul li a{
50   display: block;
51   width: 100%;
52   height: 30px;
53   padding-top: 10px;
54   text-decoration: none;
55   color: black;
56   background-color: cornflowerblue;
57 }
58
59 ul li a:hover{
60   text-decoration: underline;
61   color: cornflowerblue;
62   background-color: white;
63 }
```

hover para el selector `<a>`, donde al pasar el mouse, cambiar el color de letra y color de fondo

Figura 181: Captura de pantalla
Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome, para visualizar la página. Al pasar el mouse en uno de los elementos, cambia la apariencia de sus colores.

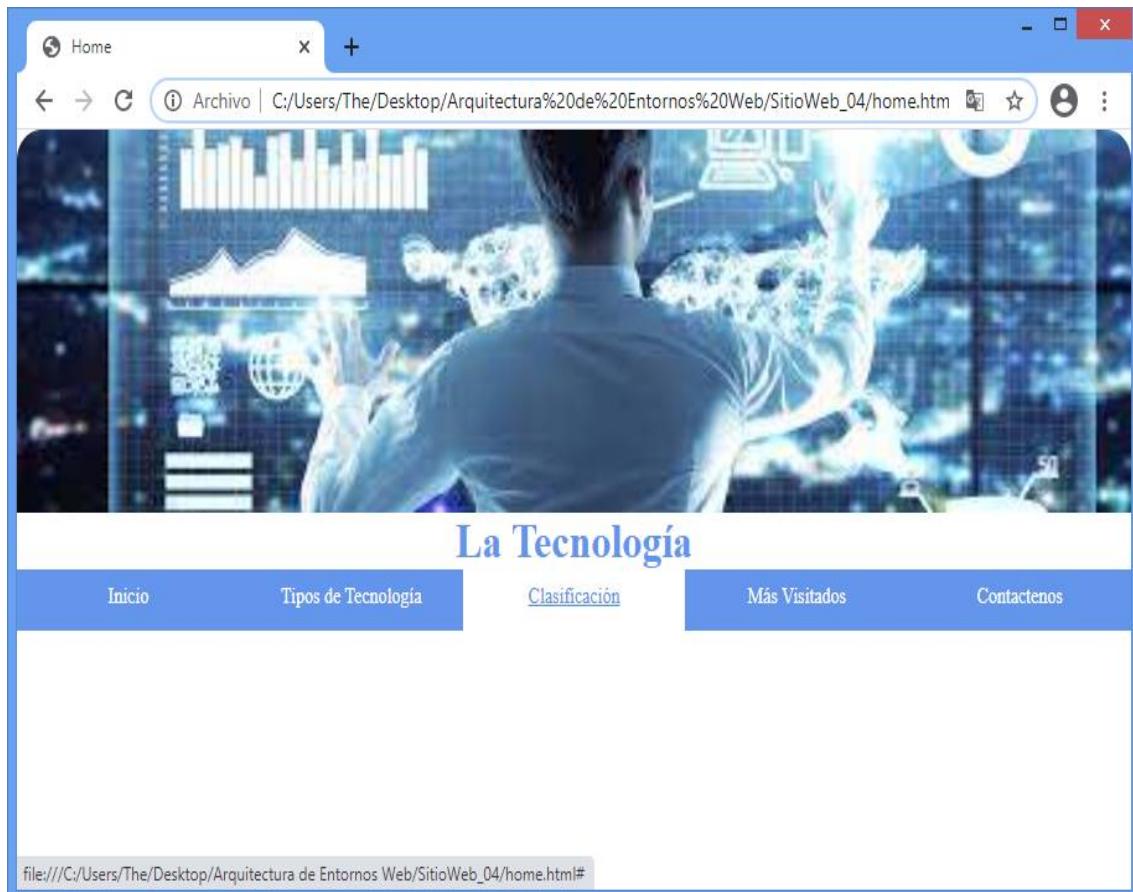
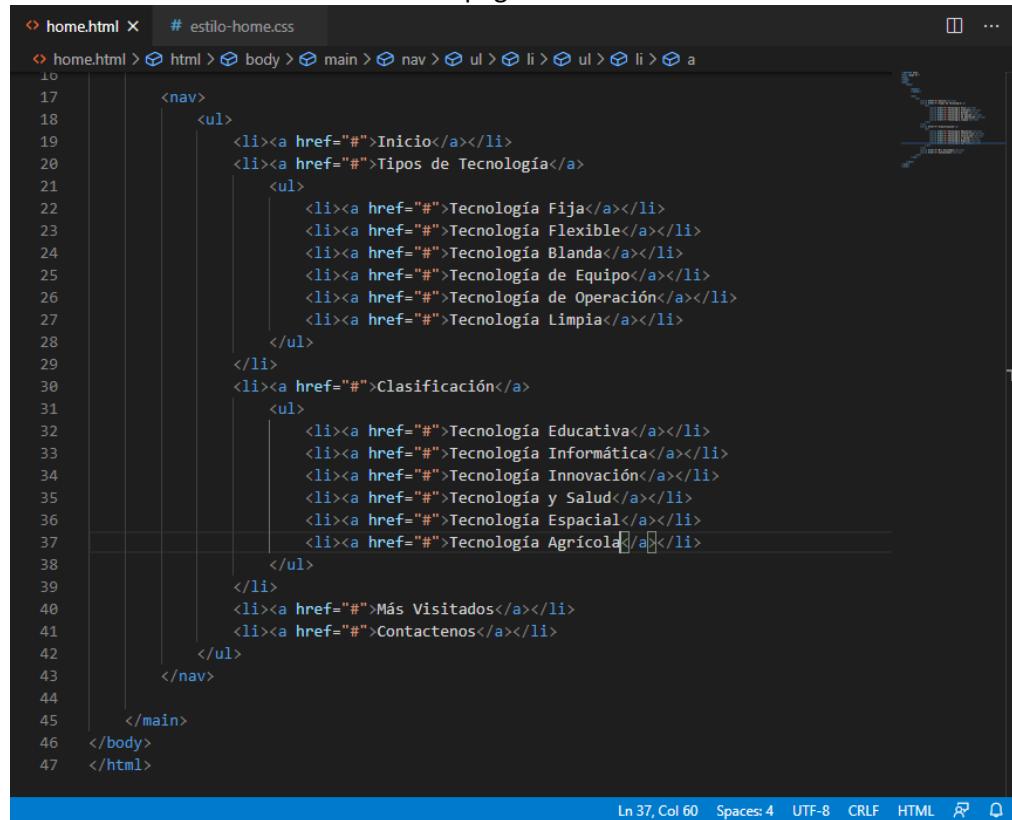


Figura 182: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 2

Se solicita diseñar una página de inicio, donde se implemente un menú desplegable de navegación utilizando estilo CSS.

1. Primero crea la lista no ordenada en la página HTML.



```

<home.html> # estilos-home.css
<home.html> html body main nav ul ul li a
16
17     <nav>
18         <ul>
19             <li><a href="#">Inicio</a></li>
20             <li><a href="#">Tipos de Tecnología</a>
21                 <ul>
22                     <li><a href="#">Tecnología Fija</a></li>
23                     <li><a href="#">Tecnología Flexible</a></li>
24                     <li><a href="#">Tecnología Blanda</a></li>
25                     <li><a href="#">Tecnología de Equipo</a></li>
26                     <li><a href="#">Tecnología de Operación</a></li>
27                     <li><a href="#">Tecnología Limpia</a></li>
28                 </ul>
29             </li>
30             <li><a href="#">Clasificación</a>
31                 <ul>
32                     <li><a href="#">Tecnología Educativa</a></li>
33                     <li><a href="#">Tecnología Informática</a></li>
34                     <li><a href="#">Tecnología Innovación</a></li>
35                     <li><a href="#">Tecnología y Salud</a></li>
36                     <li><a href="#">Tecnología Espacial</a></li>
37                     <li><a href="#">Tecnología Agrícola</a></li>
38                 </ul>
39             </li>
40             <li><a href="#">Más Visitados</a></li>
41             <li><a href="#">Contactenos</a></li>
42         </ul>
43     </nav>
44
45 </main>
46 </body>
47 </html>

```

Ln 37, Col 60 Spaces: 4 UTF-8 CRLF HTML

Figura 183: Captura de pantalla

Fuente. - Elaboración propia

Guarda el archivo HTML. Luego, abre el documento con el navegador Chrome para visualizar la página.



Figura 184: Captura de pantalla

Fuente. - Elaboración propia

En las reglas de estilo define la primera lista ; define las propiedades para quitar las viñetas y márgenes de la lista en la etiqueta a través de la propiedad list-style-type: none. Para ocultar los submenús (ul ul) define la propiedad visibility: hidden, position, márgenes y relleno.

```

home.html      # estilo-home.css X
css > # estilo-home.css > 48 ul ul
26
27   nav{
28     width: 100%;
29     height: auto;
30     float: left;
31   }
32
33   ul{
34     list-style-type: none;
35     margin:0;
36     padding: 0;
37   }
38
39   ul ul{
40     position: absolute;
41     margin: 0;
42     padding: 0;
43     list-style-type: none;
44     opacity: 0;
45     transition: opacity 0.35s easy 0.2s;
46     -webkit-transition: opacity 0.35s easy 0.2s;
47     -moz-transition: opacity 0.35s easy 0.2s;
48     visibility: hidden;
49   }

```

Ln 46, Col 48 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🔍

Figura 185: Captura de pantalla

Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome para visualizar la página.

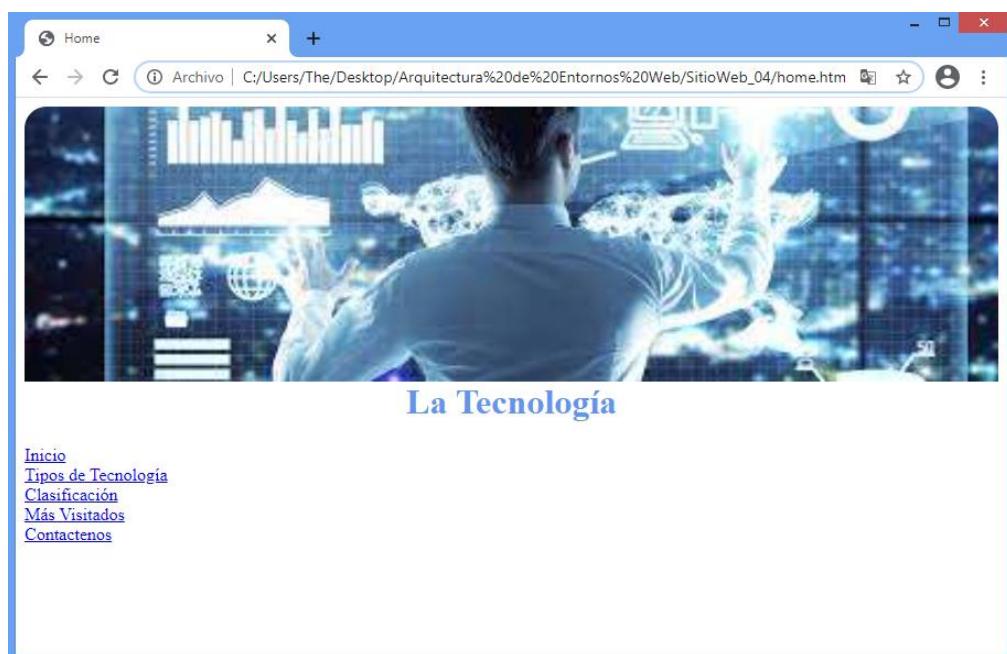
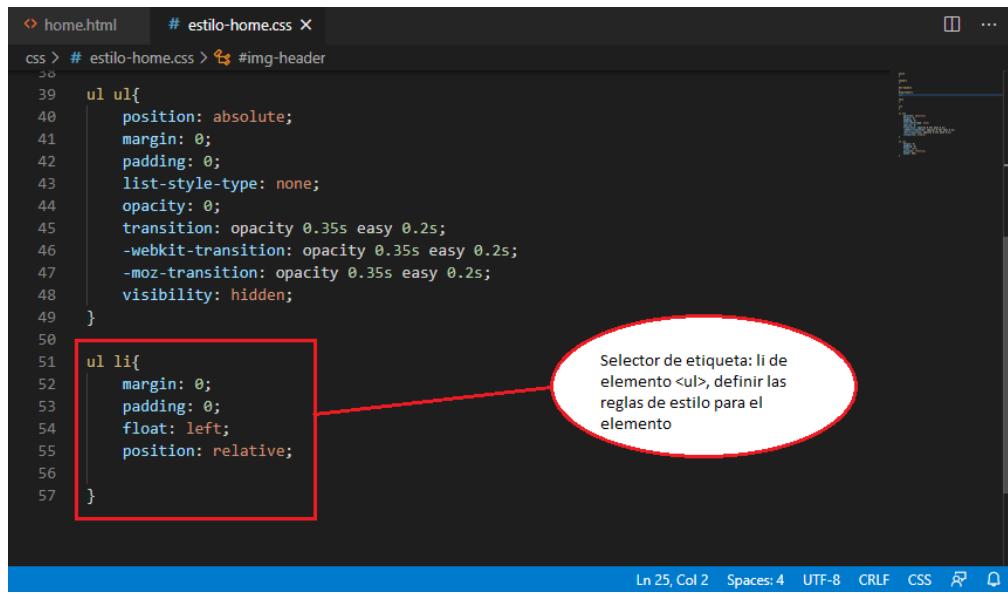


Figura 186: Captura de pantalla

Fuente. - Elaboración propia

Para que el menú sea horizontal, en la etiqueta de la lista, agrega la propiedad **float: left**, para hacer que los elementos de la lista se coloquen uno al lado del otro. También define los bordes, posición, márgenes y relleno.



```

home.html      # estilos-home.css
css > # estilos-home.css > #img-header
38
39   ul ul{
40     position: absolute;
41     margin: 0;
42     padding: 0;
43     list-style-type: none;
44     opacity: 0;
45     transition: opacity 0.35s easy 0.2s;
46     -webkit-transition: opacity 0.35s easy 0.2s;
47     -moz-transition: opacity 0.35s easy 0.2s;
48     visibility: hidden;
49   }
50
51   ul li{
52     margin: 0;
53     padding: 0;
54     float: left;
55     position: relative;
56   }
57

```

Ln 25, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙ 🔍

Figura 187: Captura de pantalla
Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome, para visualizar la página.

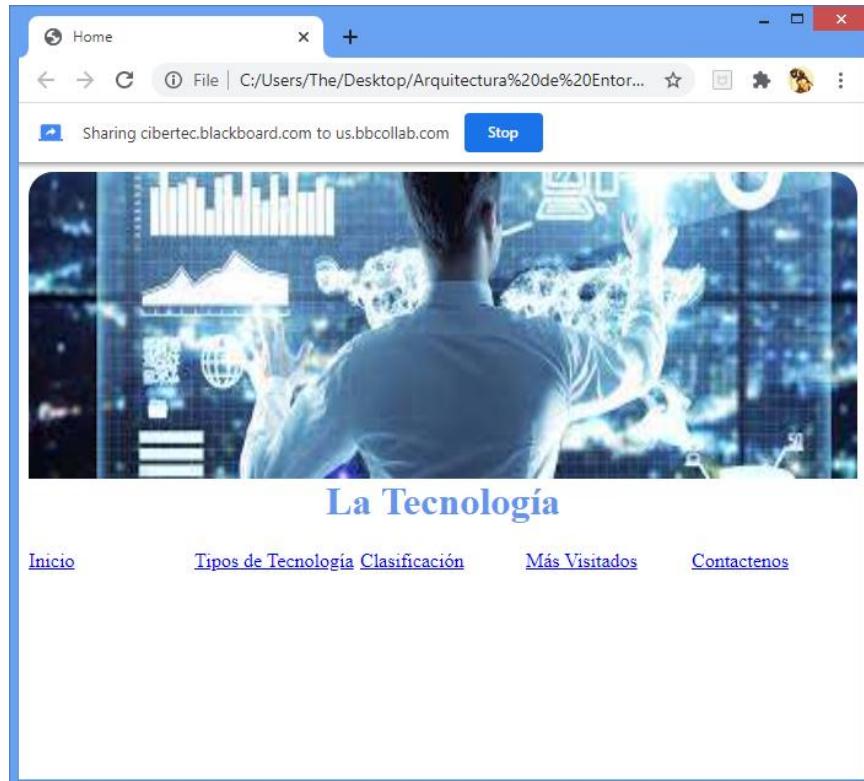


Figura 188: Captura de pantalla
Fuente. - Elaboración propia

A continuación, define estilos directamente sobre los vínculos, `<a>` asignando: bloque, tamaño, tipo de letra, colores, decoración, etc. Para seleccionar los vínculos dentro de los elementos de lista, encadena los nombres de las etiquetas en el selector CSS como se realizó con los ``. Además, define propiedades del evento hover de la etiqueta `<a>`, para cambiar el color de letra y de fondo al acercar el cursor en el vínculo.

```

home.html      # estilo-home.css x
css > # estilo-home.css > ul li a
58
59 ul li a{   Selector de etiqueta <a>, regla para los hipervínculos contenidos en <li>
60   display: block; 
61   width: 200px; 
62   height: 30px; 
63   line-height: 30px; 
64   border: 1px solid black; 
65   text-decoration: none; 
66   color: black; 
67   background-color: cornflowerblue; 
68 
69   position: relative; 
70 } 
71 
72 ul li a:hover{ Evento hover, del selector de 
73   color: cornflowerblue; 
74   background-color: white; 
75 } 
76

```

Ln 68, Col 1 Spaces: 4 UTF-8 CRLF CSS ⌂ ⌂

Figura 189: Captura de pantalla

Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome para visualizar la página.

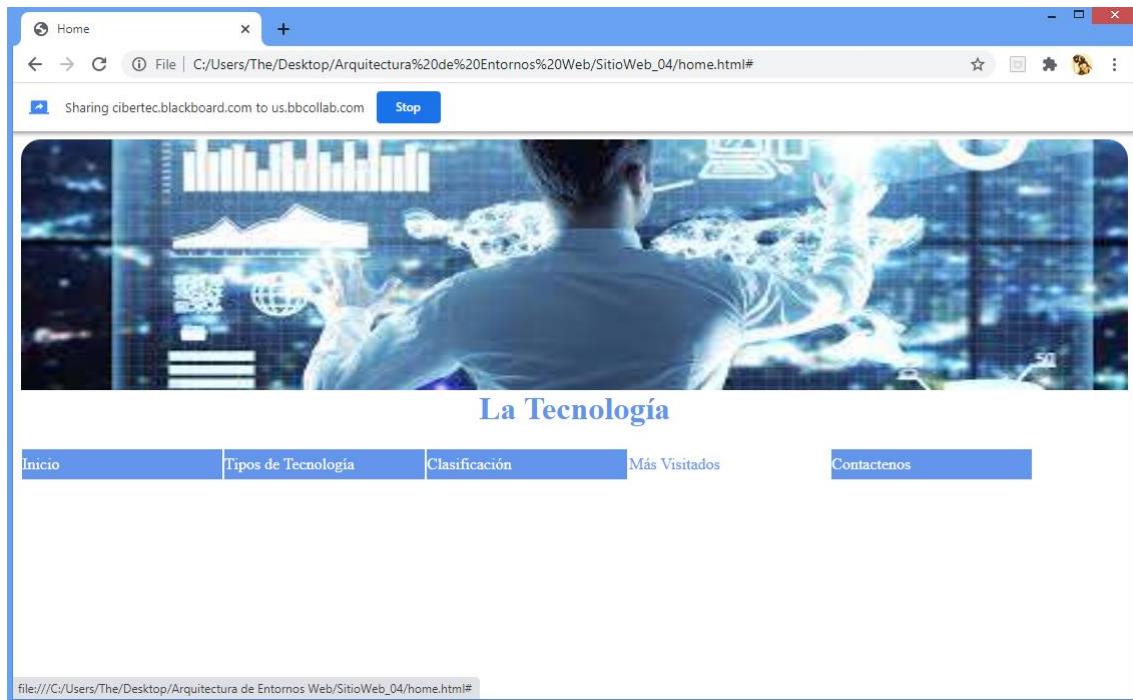


Figura 190: Captura de pantalla

Fuente. - Elaboración propia

Por último, define propiedades del evento hover de la etiqueta para cada uno de los subitems, donde al pasar el cursor sobre el elemento del menú, se visualiza los elementos del subitem.

```

css > # estilo-home.css > ul li a
58
59 > ul li a[...]
60 ]
61
62 ul li a:hover{
63   color: cornflowerblue;
64   background-color: white;
65 }
66
67 ul li:hover ul{
68   opacity: 1;
69   visibility: visible;
70 }
71
72
73
74
75
76
77
78
79
80

```

Cuando el cursor esté encima de un elemento, visualiza sus subítems

Figura 191: Captura de pantalla

Fuente. - Elaboración propia

Actualiza el documento en el navegador Chrome para visualizar la página.

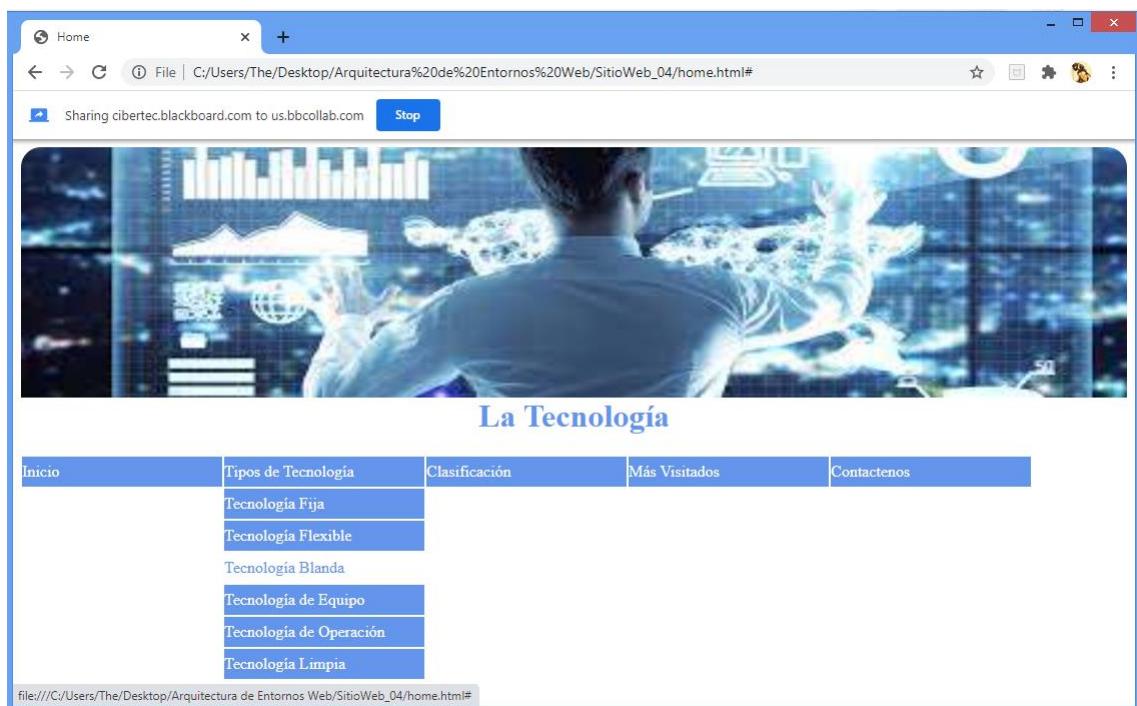


Figura 192: Captura de pantalla

Fuente. - Elaboración propia

Resumen

1. HTML5 y CSS3 son dos recursos esenciales para diseñadores y desarrolladores en la actualidad, si lo que se busca es crear sitios web funcionales, así como estéticamente hermosos. Estos lenguajes permiten crear páginas web con efectos realmente alucinantes e innovadores y además son muy simples de usar.
2. Otro elemento bastante típico y común en muchas páginas web es la zona de navegación, formada por enlaces a las zonas principales de la web, o por un menú de navegación, desplegable de enlaces, etc. Por lo visto Html5 interpreta que es una zona que deberíamos identificar y ha creado para ello la nueva etiqueta `<nav> </nav>`.
3. Para diseñar un menú de navegación se utiliza listas, porque un menú básicamente es una lista, no es un párrafo, ni un título, es una lista de vínculos.
4. Para cambiar el aspecto de las listas, CSS permite cambiar el aspecto de la lista, para convertirla en el bonito menú que se desea, sin que deje de ser lista.
5. Las listas se renderizan en el navegador con ciertas características, como el margen/relleno (los cuales son interpretados de diferente manera en los distintos navegadores como opera, Chrome, Internet, entre otros) de la lista completa, el margen / relleno (idem) de cada ítem de la lista y las viñetas, además de un salto de línea para cada ítem por ser elementos de bloque.

Recursos

Puede revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- <http://www.cristalab.com/tutoriales/crear-menus-de-navegacion-en-css-usando-listas-c130/>
- <http://www.antocas.com/menu-desplegable-basico-html5-y-css3-drop-down-submenu/>



MULTIMEDIA, TRANSICIONES ANIMACIONES

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña y construye páginas web para un sitio web aplicando etiquetas HTML5, estilos CSS 3 y multimedia.

TEMARIO

4.1 Tema 7 : Audio, video e iframe

- 4.1.1. : Etiqueta de audio: propiedades y atributos
- 4.1.2. : Etiqueta de video: propiedades y atributos
- 4.1.3. : Etiqueta Iframe, reproducir videos de Youtube

4.2 Tema 8 : Animaciones, transiciones y transformaciones

- 4.2.1. : Animaciones, propiedades
- 4.2.2. : Transiciones, propiedades
- 4.2.3. : Transformaciones, propiedades

ACTIVIDADES PROPUESTAS

- Los alumnos diseñan páginas web con etiquetas HTML5, hojas de estilo CSS3 y bloques de contenidos.
- Los alumnos diseñan una página web con audio, video y reproductor de video de youtube.

4.1 AUDIO, VIDEO E IFRAME

Algunos de los cambios más notables en HTML5 es la adición de etiquetas para soportar elementos multimedia y gráficos dinámicos. Esto incluye la etiqueta **<audio>** para reproducir sonidos, **<video>** para videos y **<canvas>** para gráficos dinámicos, todo esto dentro del web browser y sin depender de software externo. Los nuevos elementos multimedia habilitan a los browsers modernos para proveer experiencias enriquecidas y gráficos dinámicos a partir del código HTML.

El soporte para usar video, manipular imágenes, y visualización de datos permitirá a los diseñadores y desarrolladores crear experiencias ricas y atractivas que funcionen en múltiples plataformas.

4.1.1 Etiqueta de audio: propiedades y atributo

Hasta ahora, para incluir un elemento multimedia en un documento, se hacía uso del elemento **<object>**, cuya función es incluir un elemento externo genérico. Debido a la incompatibilidad entre navegadores, se hacía también necesario el uso del elemento **<embed>** y duplicar una serie de parámetros.

HTML5 introduce soporte integrado para el contenido multimedia gracias a los elementos **<audio>** y **<video>**, ofreciendo la posibilidad de insertar contenido multimedia en documentos HTML.

En este ejemplo, se inserta el audio abc.ogg en un documento HTML.

```
<audio src="../audios/abc.ogg" controls>
</audio>
```

En este caso, el elemento **<audio>** define el origen del archivo multimedia a través de la propiedad **src**. Este atributo especifica la URL del archivo a ser reproducido. Este atributo, normalmente será reemplazado por el elemento **<source>** para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.

Atributos de la etiqueta **<audio>**

src:

URL/path del audio a reproducir.

autoplay:

Este atributo especifica que el audio comience automáticamente. Se puede especificar de 2 maneras `autoplay="autoplay"` o `autoplay`.

preload:

En caso de omisión su valor es “auto”. Sirve para especificar si se desea que el audio se vaya cargando independientemente si se da al play o no. Aumentaría el consumo de ancho de banda, pero el usuario tendría menos cortes en la reproducción.

Existen 3 posibles valores para este atributo:

- preload = "none": el audio no comienza a descargarse hasta que el usuario seleccione play.
- preload = "metadata": solo se cargan los metadatos del audio (dimensiones, primer frame, duración...).
- preload = "auto": el audio se descarga independientemente de la ejecución del play.

controls:

Especifica si se muestran los típicos controles del audio (play, stop, volumen...). Tiene 2 posibles sintaxis: controls="controls" o controls.

loop:

Especifica la reproducción continua del audio, es decir, una vez acabada la reproducción comienza de nuevo a reproducirse. Tiene 2 posibles sintaxis: loop="loop" o loop.

muted:

Funcionalidad para quitar el volumen al audio.

mediagroup:

Establece un grupo de reproducción al que pertenece el audio.

Formato de audio

Los archivos de audio están codificados mediante un codec específico, y es importante conocer estos detalles para saber qué formatos están soportados por los diferentes navegadores.

A continuación, se presenta un listado de los formatos/codecs de audio más conocidos y utilizados:

Formato	Codec utilizado	Características
MP3	MPEG Layer-3	Buena calidad
AAC	Advanced Audio Coding	Mejora el MP3. Usado como audio en MP4
OGG	Ogg Vorbis	Buena calidad. Alternativa libre a MP3
Opus	Opus	Buena calidad. Alternativa libre a MP3
FLAC	FLAC Audio Lossless	Compresión sin pérdidas. Alto tamaño

Audio (modo avanzado)

La etiqueta <audio> también puede actuar como etiqueta contenedora e incluir varias etiquetas HTML para dotar de mayor compatibilidad, o capacidades adicionales.

La etiqueta <source> tiene como atributos src y type, establece un archivo de audio o lo añade como alternativa.

```
<audio>
  <source src="audio.opus" />
  <source src="audio.ogg" />
  <source src="audio.mp3" />
</audio>
```

4.1.2 Etiqueta de video: propiedades y atributos

En HTML5 se puede mostrar videos directamente desde el navegador. De hecho, si se arrastra un video a la ventana del navegador, se observa que comienza a reproducirse en él.

Para poder insertar videos en las páginas HTML se tiene que utilizar la etiqueta <video>, que junto a la etiqueta <source> se puede utilizar estas capacidades multimedia de HTML5.

Atributos de la etiqueta <video>

src:

URL/path, donde se va a reproducir el video. Es obligatoria si actúa como etiqueta.

poster:

URL/Path, muestra una imagen a modo de presentación. El atributo poster indica la imagen que el navegador debe mostrar mientras el vídeo se está descargando, o hasta que el usuario reproduce el vídeo. Si no se indica este atributo, el navegador muestra el primer fotograma del vídeo.

autoplay:

Este atributo especifica que el video comience automáticamente. Se puede especificar de estas 2 maneras autoplay="autoplay" o autoplay.

preload:

En caso de omisión su valor es "auto". Sirve para especificar si se desea que el video se cargue independientemente si se da al play o no. Aumentaría el consumo de ancho de banda pero el usuario tendría menos cortes en la reproducción.

Existen 3 posibles valores para este atributo:

- preload = "none": el video no comienza a descargarse hasta que el usuario seleccione play.
- preload = "metadata": solo se cargan los metadatos del video (dimensiones, primer frame, duración...).
- preload = "auto": el video se descarga independientemente de la ejecución del play.

controls:

Especifica si se muestran los típicos controles del video (play, stop, volumen...). Tiene 2 posibles sintaxis: controls="controls" o controls.

loop:

Especifica la reproducción continua del video, es decir, una vez acabada la reproducción comienza de nuevo a reproducirse. Tiene 2 posibles sintaxis: loop="loop" o loop.

muted:

Funcionalidad para quitar el volumen al video.

mediagroup:

Establece un grupo de reproducción al que pertenece el video.

width, height:

Los atributos height y width indican al navegador el tamaño del vídeo en pixels. Si no se indican estas medidas, el navegador utiliza las medidas definidas en el vídeo de origen, si están disponibles. De lo contrario, utiliza las medidas definidas en el fotograma poster, si están disponibles. Si ninguna de estas medidas está disponible, el ancho por defecto es de 300 pixels.

Si únicamente se especifica una de las dos medidas, el navegador automáticamente ajusta la medida de la dimensión no proporcionada, conservando la proporción del vídeo. Si, por el contrario, se especifican las dos medidas, pero no coinciden con la proporción del vídeo original, el vídeo no se deforma a estas nuevas dimensiones, sino que se muestra en formato letterbox manteniendo la proporción original.

Formato de video

Los archivos de video están codificados mediante un codec específico, y es importante conocer estos detalles para saber qué formatos están soportados por los diferentes navegadores.

A continuación, se presenta un listado de los formatos/codecs de video más conocidos y utilizados:

Formato	Codec utilizado	Características
MP4	X264, DivX H264	Alta calidad. Codec x264 libre
WebM	VP8, VP9	Alternativa libre a MP4 de Google
AV1	Basado en VP01, Daala y Thor	Compete con HEVC/H.265
HEVC	X265, DivX HEVC	Futura evolución de MP4
AVI	XviD, DivX 3/5	Menor compresión que MP4

Video (Modo avanzado)

La etiqueta <video> también puede actuar como etiqueta contenedora e incluir varias etiquetas HTML para dotar de mayor compatibilidad, o capacidades adicionales.

La etiqueta <source> tiene como atributos src y type, establece un archivo de video o lo añade como alternativa.

```
<video width="640" height="480">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
  <source src="video.ogv" type="video/ogg" />
</video>
```

4.1.3 Etiqueta Iframe, reproducir videos de YouTube

El elemento HTML inline frame (<iframe>) representa un contexto de navegación anidada, el cual permite incorporar otra página HTML en la página actual. Cada contexto de navegación tiene su propia historia, sesión y documento activo. El contexto de navegación que incluye el contenido implícito se llama contexto de navegación principal. El contexto de navegación de nivel superior (que no tiene padre) es típicamente la ventana del navegador. Las propiedades de esta etiqueta en HTML5.

src,

URL del documento HTML que se visualiza en el ifram.

height,

Altura que ocupará el iframe en el documento, se expresa en pixeles.

width,

Ancho que ocupará el iframe en el documento, se expresa en pixeles.

align,

Alineamiento del frame en la página.

name,

Nombre que identifica al iframe.

longdesc,

Dirección url en la que puede encontrarse una descripción larga del contenido del iframe.

frameborder,

Especifica el grosor del border del frame, por defecto es cero.

framespacing,

Especifica el espacio entre la caja y el contenido del frame.

scrolling,

"yes | no | auto" - Indica si el iframe debe mostrar barras de scroll (horizontal y vertical) cuando el contenido incluido no cabe en el iframe.

Convertidor de audio y video online: mp3, ogg, mp4

Un desarrollador, si desea implementar páginas HTML con reproductores de video, va a tener que codificarse al menos en dos (idealmente en tres) formatos de video para que puedan funcionar con la especificación de HTML5. Actualmente, se dispone de varias herramientas que pueden ayudar con esta labor.

- **Miro Video Converter:** es tu herramienta de referencia, este conversor de código abierto está disponible en versiones para Mac y para Windows, convierte prácticamente cualquier formato de video de origen a todos los formatos de HTML5 que quieras utilizar y también a otros formatos específicos de dispositivos y plataformas.
- **Handbrake:** otro buen conversor de código abierto para Mac y Windows (y Linux también). Handbrake es una buena opción concretamente para convertir a H.264.
- **MPEG Streamclip ()**: aunque no es código abierto, MPEG Streamclip se puede conseguir gratuitamente en versiones para Mac y Windows. Se maneja también con H.264, aunque puede convertir vídeos a otros muchos formatos que pueden servir para utilizarlos fuera de la especificación de HTML5.
- **Firefogg**: es una herramienta más especializada. Firefogg es una extensión de Firefox que convierte videos únicamente a formato Ogg, de ahí el nombre. Para poder utilizarla necesitarás descargar antes Firefox y luego instalar esta extensión.

En este ejemplo, la etiqueta html <iframe> se utiliza para incrustar un video de YouTube.

```
<iframe width="560" height="315" src="//www.youtube.com/embed/5AKY58ZSJG" frameborder="0" allowfullscreen="">
</iframe>
```

Para hacer un posicionamiento más específico o darle unas características más concretas a los IFRAMES puedes utilizar la etiqueta **style**.

LABORATORIO 1

Se solicita diseñar una página HTML para la empresa Toshiba, en la cual se debe agregar videos en la barra lateral de la página.



Figura 193: Captura de pantalla
Fuente. - Elaboración propia

Creando la pagina HTML, CSS y enlazando la página de estilo

Primero, crea la página html: **home.html**. Luego, crea el archivo de estilo: **estilo-home.css**. Para enlazar la hoja de estilo, utiliza la etiqueta **<link>**.

```

EXPLORER          ...      ◊ home.html X # estilos-home.css
OPEN EDITORS
SITIOWEB_05
  audios
  css
    # estilos-home.css
  imagenes
  js
  videos
  home.html

```

```

◊ home.html > ...
  1 <!DOCTYPE html>
  2 <html lang="en">
  3   <head>
  4     <meta charset="UTF-8">
  5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
  6     <title>Home</title>
  7     <link rel="stylesheet" href="css/estilos-home.css">
  8   </head>
  9   <body>
 10
 11 </body>
 12 </html>

```

Página html:
home.html

Hoja de estilo css:
estilos-home.css

Figura 194: Captura de pantalla
Fuente. - Elaboración propia

Agregando reglas de estilo a la cabecera de la página: redes sociales

En el archivo css, define las reglas de estilo para la cabecera de las redes sociales:

- #header-rs, identificador que define las reglas para el bloque donde agrupa las imágenes
- .rs, clase que define el sector para las opciones de redes sociales
- .span-rs, clase que define las reglas del elemento

```

1  home.html      # estilos-home.css X
2
3  css > # estilos-home.css > #header-rs
4
5  1
6  2   main{
7  3     width: 100%; ← Defina las reglas para
8  4     height: auto; el selector main
9  5     float: left;
10 6   }
11
12 8   #header-rs[← Defina las reglas para el
13 9     width: 98%; selector id: header-rs, el cual
14 10    padding-left: 1%; agrupa las imágenes de redes
15 11    padding-right: 1%; sociales
16 12    height: auto;
17 13    float: left;
18 14    display: grid;
19 15    grid-template-columns: 90% 1fr 1fr 1fr;
20 16  ]
21
22
23 18   .rs{← Defina las reglas del selector
24 19     width: 40px; de class: rs, imágenes de las
25 20     height: 40px; redes sociales
26 21   }
27
28 23   .span-rs{← Defina las reglas para el
29 24     padding-top: 10px; selector class: span-rs
30 25     text-align: right;
31 26   }
32
33 27

```

Ln 11, Col 22 Spaces: 4 UTF-8 CRLF CSS ⚙️ ⌂

Figura 195: Captura de pantalla

Fuente. - Elaboración propia

Agregando las etiquetas en la página HTML

Definida las reglas en el css, dentro del body define la etiqueta <div> asignando el identificador header-rs. Después, agrega los elementos e para posicionar las imágenes y el texto de las redes sociales.

```

1  home.html      # estilos-home.css
2
3  home.html > html > body > main
4
5  1   <!DOCTYPE html>
6  2   <html lang="en">
7  3     <head>
8  4       <meta charset="UTF-8">
9  5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 6       <title>Home</title>
11 7       <link rel="stylesheet" href="css/estilos-home.css">
12 8     </head>
13 9     <body>
14 10    <main>
15 11      <div id="header-rs">
16 12        <span class="span-rs">Siguenos en...</span>
17 13        
18 14        
19 15        
20 16      </div>
21 17    </main>
22 18  </body>
23 19  </html>

```

Ln 17, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚙️ ⌂

Figura 196: Captura de pantalla

Fuente. - Elaboración propia

Agregando reglas de estilo a la cabecera de la página: header.

En el archivo css, define las reglas de estilo para la cabecera y los elementos del bloque <header>

- header, define las reglas para el selector <header>
- #h1-header, define las reglas para el título de la página
- #img-header, define las reglas para la imagen de la cabecera de la página

```

home.html      # estilo-home.css X
css > # estilo-home.css > #header-rs
22
23 > .span-rs{ ...
24 }
25
26
27
28 header{
29     width: 100%; ← Defina las reglas para el
30     height: auto; selector header
31     float: left;
32 }
33
34 #h1-header{
35     text-align: center; ← Defina las reglas para el
36     color: cornflowerblue; selector id: h1-header, título
37 }
38 #img-header{
39     width: 100%; ← Defina las reglas para el selector
40     height: 250px; selector id: img-header, imagen de la
41     float: left; cabecera
42 }

```

Figura 197: Captura de pantalla

Fuente. - Elaboración propia

Agregando las etiquetas en la página HTML

Definida las reglas en el css, dentro del body define la etiqueta <header>. Después, agrega los elementos <h1> (título de la página) e (imagen de la cabecera de la página).

```

home.html X      # estilo-home.css
home.html > html > body > main
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head>...
4  </head>
5  <body>
6      <main>
7          <div id="header-rs">...
8      </div>
9
10     <header>
11         
12         <h1 id="h1-header">La Tecnología y la Informática</h1>
13     </header>
14
15     </main>
16
17 </body>
18 </html>

```

Figura 198: Captura de pantalla

Fuente. - Elaboración propia

Para visualizar el diseño de la página html, presiona F5 y selecciona la opción **Chrome (Preview)**.



Figura 199: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en la barra lateral <aside>

En el archivo CSS, define las reglas para barra lateral.

- #cuerpo, regla para el bloque <div> que agrupará al <aside> y <section>
- aside, regla para el bloque de barra lateral
- .titulo, regla para la etiqueta <h3> del bloque <aside>
- .videos, regla para la etiqueta <video>

```

css > # estilo-home.css > main
css > # estilo-home.css > main
44  #cuerpo{
45    width: 100%;
46    height: auto;
47    margin-top: 5px;
48    margin-bottom: 5px;
49    float: left;
50  }
51
52  aside{
53    width: 30%;
54    height: auto;
55    border: 1px dashed;
56    float: left;
57  }
58
59  .titulo{
60    text-align: center;
61    color: white;
62    background-color: cornflowerblue;
63  }
64  .videos{
65    width: 97.5%;
66    height: 150px;
67    margin: 1%;
68    border: 1px solid;
69    float: left;
70  }

```

Defina la regla para el selector id: cuerpo, agrupa al aside y section

Defina el selector aside, barra lateral, agrupa los videos

Defina el selector de class: titulo, heading para colocar titulos

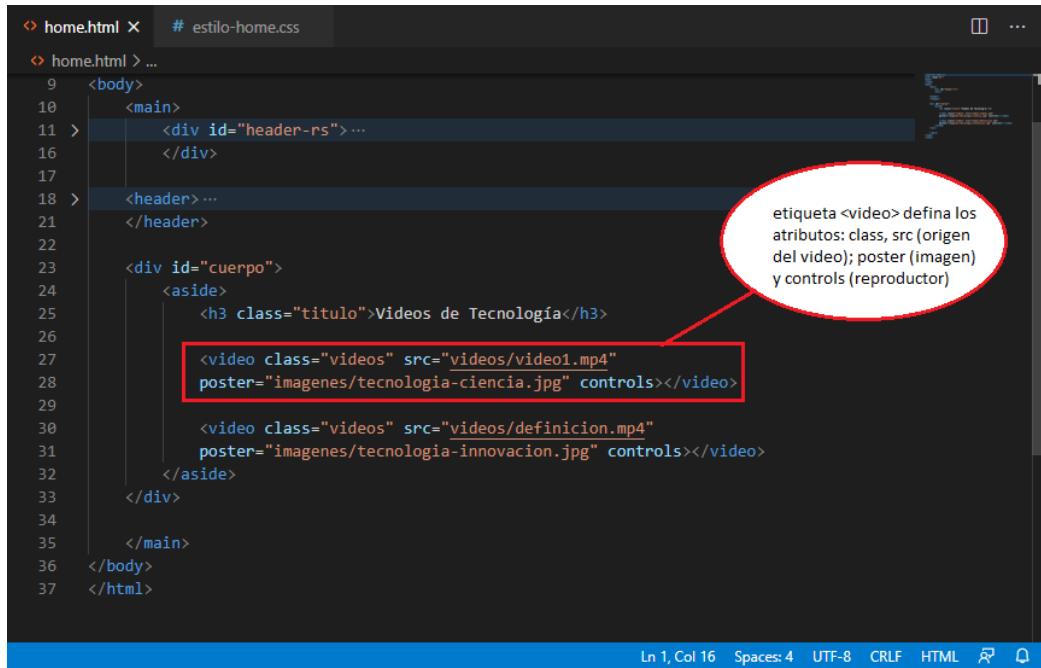
Defina el selector de class: videos, reglas para los videos agregados a la página

Figura 200: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <aside> para la página HTML

Definida las reglas en el archivo css, define la etiqueta <aside> para mostrar los videos.



```

1 <div id="cuerpo">
2   <aside>
3     <h3 class="titulo">Videos de Tecnología</h3>
4     <video class="videos" src="videos/video1.mp4"
5        poster="imagenes/tecnologia-ciencia.jpg" controls></video>
6
7     <video class="videos" src="videos/definicion.mp4"
8        poster="imagenes/tecnologia-innovacion.jpg" controls></video>
9   </aside>
10  </div>
11
12  </main>
13</body>
14</html>

```

Figura 201: Captura de pantalla
Fuente. - Elaboración propia

Presiona la tecla **F5** para hacer un preview a la página.



Figura 202: Captura de pantalla
Fuente. - Elaboración propia

Agregando las reglas en el cuerpo de la página <section>

En el archivo CSS, define las reglas para el cuerpo de la página

- section, regla para el bloque <section>, su ancho será 69%, alinear derecha
- .parrafo, regla para los elementos <p> del bloque section
- .img-section, regla para los elementos agrupados en <figure>

```

css > # estilo-home.css > .titulo
65 > .videos{ ...
71 }
72
73 section{
74     width: 69%; ← Reglas de estilo para el selector <section>
75     height: auto;
76     float: right;
77 }
78
79 .parrafo{
80     text-align: justify; ← Reglas de estilo para el selector de class: parrafo
81     font-size: 12px;
82 }
83 figure{
84     margin: 0; ← Reglas de estilo para el selector <figure>, agrupa las imágenes
85     padding: 0;
86 }
87
88 .img-section{
89     width: 30%; ← Reglas de estilo para el selector de class: img-section
90     height: 120px;
91 }

Ln 63, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🔍

```

Figura 203: Captura de pantalla

Fuente. - Elaboración propia

Agregando la etiqueta <section> para la página HTML

Definida las reglas, define la etiqueta <section> y sus elementos para relatar el tema de la página: tecnología.

```

home.html X # estilo-home.css
home.html > html > body > main > div#cuerpo > section > figure > img.img-section
34 <section>
35   <p class="parrafo">
36     La denominada IT, como su nombre lo indica se basa en el estudio, el
37     desarrollo y la práctica de los sistemas informáticos, especialmente en
38     lo que se refiere al uso del software y el hardware, por lo que podríamos
39     definir brevemente su actividad dentro del mundo de las computadoras. <br>
40     Por ende, la tecnología informática se ocupa de todos los procesos y los
41     medios requeridos para tratar la información.
42   <br>
43   No obstante, lo cierto es que con el avance exponencial del campo de la
44     tecnología, en la actualidad el término tecnología informática debe ser
45     utilizado para una mayor plataforma en lo que respecta a la computación.
46     Tengamos en cuenta que en definitiva, detrás de cada nuevo equipamiento,
47     detrás de cada nueva tarea desarrollada por el software, se encuentra la
48     tecnología, permitiendo extender sus usos a lugares inimaginables.
49   </p>
50   <figure>
51     
52     
53     
54   </figure>
55   <p class="parrafo">
56     La denominada IT, como su nombre lo indica se basa en el estudio, el
57     desarrollo y la práctica de los sistemas informáticos, especialmente en
58     lo que se refiere al uso del software y el hardware, por lo que podríamos
59     definir brevemente su actividad dentro del mundo de las computadoras. <br>
60     Por ende, la tecnología informática se ocupa de todos los procesos y los
61     medios requeridos para tratar la información.
62   </p>
63 </section>

Ln 53, Col 70 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🔍

```

Figura 204: Captura de pantalla

Fuente. - Elaboración propia

Presiona la tecla F5 para visualizar (preview) la página **home.html**



Figura 205: Captura de pantalla

Fuente. - Elaboración propia

Agregando las reglas en el pie de la página <footer>

En el archivo CSS, define las reglas para el pie de la página

- footer, defina su display: grid y su plantilla será de 2 columnas
- .div-footer, regla para los elementos <div> del bloque footer.
- .img-enciclopedia, regla para los elementos

```

home.html # estilo-home.css X
css > # estilo-home.css > ...
92
93 footer{
94   width: 100%;
95   height: auto;
96   display: grid;
97   grid-template-columns: 1fr 1fr;
98   float: left;
99 }
100
101 .div-footer{
102   width: 100;
103   height: auto;
104   float: left;
105 }
106
107 .img-enciclopedia{
108   width: 50px;
109   height: 50px;
110 }
111
112 .img-enciclopedia:hover{
113   border:1px dashed;
114 }

```

Figura 206: Captura de pantalla

Fuente. - Elaboración propia

Agregando el pie de página <footer>

Definida las reglas en style.css, define la etiqueta <footer> agregando su identificador **pie**.

```

    <div class="div-footer">
        <h4>Tecnología</h4>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Tipos de Tecnología</a></li>
            <li><a href="#">Clasificación de las Tecnologías</a></li>
            <li><a href="#">Tecnología e Informática</a></li>
            <li><a href="#">Contactenos</a></li>
        </ul>
    </div>
    <div class="div-footer">
        <h4>Puede revisar en...</h4>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
    </div>
</div>
</main>
</body>
</html>

```

Figura 207: Captura de pantalla

Fuente. - Elaboración propia

Presiona la tecla **F5** para visualizar (preview) la página html.



Figura 208: Captura de pantalla

Fuente. - Elaboración propia

LABORATORIO 2

Se solicita diseñar una página HTML para la empresa Toshiba, donde, utilizando la herramienta VideoLightBox se agregue videos en el bloque footer.



Figura 209: Captura de pantalla
Fuente. - Elaboración propia

Creando la página html y la hoja de estilo

Continuando con el desarrollo del sitio web 5, agrega la página html: **tendencia.html** y su hoja de estilo **estilo-tendencia.css**.

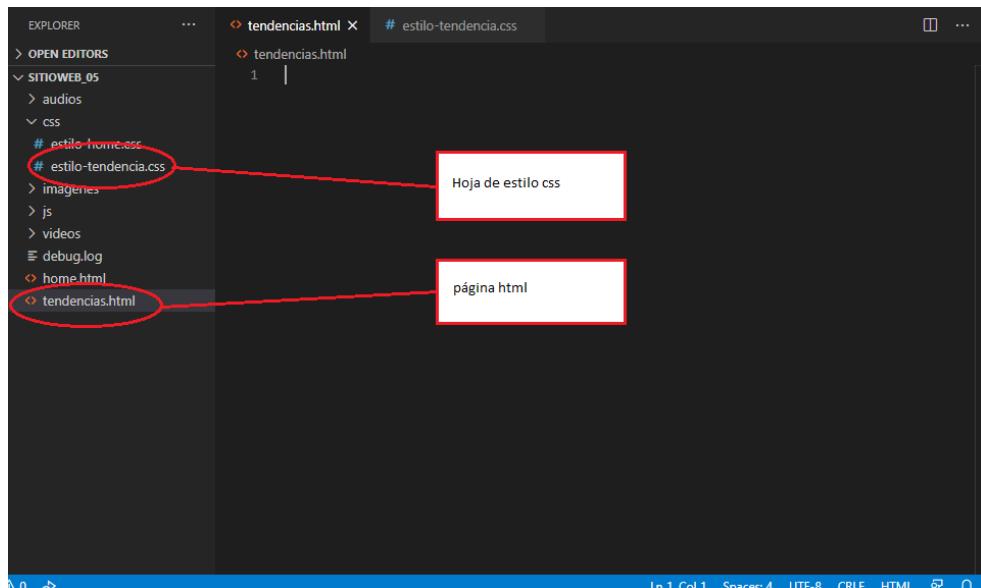


Figura 210: Captura de pantalla
Fuente. - Elaboración propia

Enlazando hoja de estilo

Creado los archivos, en la página html enlaza la hoja de estilo css a través de la etiqueta <link>.

```

<!-- tendencias.html --> # estilos-tendencia.css
<!-- tendencias.html > html > head
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Tendencia</title>
7
8       <link rel="stylesheet" href="css/estilo-tendencia.css">
9
10    </head>
11    <body>
12
13    </body>
14  </html>

```

Figura 211: Captura de pantalla

Fuente. -Elaboración propia

Agregando reglas de estilo a la cabecera de la página: header.

En el archivo css, define las reglas de estilo para la cabecera y los elementos del bloque <header>

- header, define las reglas para el selector <header>
- #h1-header, define las reglas para el título de la página
- #img-header, define las reglas para la imagen de la cabecera de la página

```

css > # estilos-tendencia.css > main
css > # estilos-tendencia.css > header
css > # estilos-tendencia.css > #h1-header
css > # estilos-tendencia.css > #img-header

```

```

1  main{
2    width: 100%;
3    height: auto;
4    float: left;
5  }
6
7  header{
8    width: 100%;
9    height: auto;
10   float: left;
11 }
12
13 #h1-header{
14   text-align: center;
15   color: cornflowerblue;
16 }
17 #img-header{
18   width: 100%;
19   height: 250px;
20   float: left;
21 }

```

Figura 212: Captura de pantalla

Fuente. - Elaboración propia

Agregando las etiquetas en la página HTML

Definida las reglas en el css, dentro del body define la etiqueta <header>. Agrega los elementos <h1> (título de la página) e (imagen de la cabecera de la página).

```

    <!-- tendencias.html --> # estilos-tendencia.css      # estilos-home.css
    <!-- tendencias.html > <html> > <body> > <main> > <header> > h1#h1-header
    1   <!DOCTYPE html>
    2   <html lang="en">
    3   <head>
    4       <meta charset="UTF-8">
    5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
    6       <title>Tendencia</title>
    7
    8       <link rel="stylesheet" href="css/estilos-tendencia.css">
    9
    10  </head>
    11  <body>
    12      <main>
    13          <header>
    14              
    15              <h1 id="h1-header">Tendencias Tecnológicas</h1>
    16          </header>
    17      </main>
    18  </body>
    19 </html>

```

Ln 15, Col 55 Spaces: 4 UTF-8 CRLF HTML

Figura 213: Captura de pantalla

Fuente. - Elaboración propia

Presiona la tecla **F5** para visualizar la página html.



Figura 214: Captura de pantalla

Fuente. - Elaboración propia

Agregando reglas de estilo al bloque <nav>

En el archivo CSS, define las etiquetas **<nav>, , **

- **<nav>**, regla para el bloque del menú de navegación
- ****, regla para el selector de lista, quitar las viñetas: `list-style: none`
- ****, regla para el selector ****, la lista sea horizontal

- <a>, regla para el selector de etiqueta, contenida en

```

23 nav{
24   width: 100%;
25   height: auto;
26   float: left;
27 }
28 ul{
29   list-style-type: none;
30   margin: 0;
31   padding: 0;
32 }
33 ul li{
34   margin: 0;
35   padding: 0;
36   float: left;
37   position: relative;
38   width: 20%;
39 }
40 ul li a{
41   display: block;
42   width: 100%;
43   line-height: 30px;
44   border: 1px solid #white;
45   text-decoration: none;
46   color: #white;
47   background-color: #cornflowerblue;
48   text-align: center;
49   position: relative;
50 }
51 ul li a:hover{
52   color: #cornflowerblue;
53   background-color: #white;
54 }

```

Ln 50, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙️ ⚙️

Figura 215: Captura de pantalla
Fuente. - Elaboración propia

Agregando las etiquetas a la página html

Definida las reglas de estilo en el archivo css, agrega las etiquetas en la página web, tal como se muestra.

```

10 <body>
11   <main>
12     <header>...
13   </header>
14 
15   <nav>
16     <ul>
17       <li><a href="#">Home</a></li>
18       <li><a href="#">Tipos</a></li>
19       <li><a href="#">Clasificación</a></li>
20       <li><a href="#">Tendencias</a></li>
21       <li><a href="#">Contactenos</a></li>
22     </ul>
23   </nav>
24 
25 </main>
26 </body>
27 </html>

```

Ln 9, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚙️ ⚙️

Figura 216: Captura de pantalla
Fuente. - Elaboración propia

Para visualizar la página, presiona la tecla **F5**.



Figura 217: Captura de pantalla
Fuente. - Elaboración propia

Agregando las reglas de estilo al bloque <section>

Defina las reglas de estilo en el archivo CSS, tal como se muestra:

- <section>, reglas para el selector de etiqueta de bloque
- .titulo, reglas para el selector de clase: alineamiento y color de letra
- .parrafo, reglas para el selector de clase: alineamiento y tamaño de letra
- .videos, reglas para el selector de clase: asignado a los iframe.

```

css > # estilos-tendencia.css > #img-header
55
56 section{
57     width: 100%;
58     height: auto;
59     margin-top: 10px;
60     margin-bottom: 10px;
61     float: left;
62 }
63
64 .titulo{
65     text-align: center;
66     color: cornflowerblue;
67 }
68
69 .parrafo{
70     text-align: justify;
71     font-size: 12px;
72 }
73
74 .videos{
75     width: 60%;
76     height: 250px;
77     margin: 10px 20% 10px 20%;
78     border-radius: 20px;
79 }

```

Figura 218: Captura de pantalla
Fuente. - Elaboración propia

Agregando la etiqueta <section> para la página HTML

Definida las reglas, define la etiqueta <section> y sus elementos para relatar el tema tendencias tecnológicas, utilizando <iframe>.

```

27 <section>
28   <h2 class="titulo">CiberSeguridad</h2>
29   <p class="parrafo">
30     Es uno de los temas más importantes en el panorama tecnológico actual y
31     lo seguirá siendo en los próximos años, ya que influye de manera transversal
32     en cualquier sector. Con el auge de los ciberataques, todas aquellas cuestiones
33     relacionadas con la defensa contra las amenazas externas y protección de los datos
34     internos son las principales preocupaciones de los encargados de TI en casi todas
35     las organizaciones. Además, en el 2020, las organizaciones deberán estar ya totalmente
36     tadas al nuevo marco de gestión y protección de datos.
37   </p>
38   <iframe class="videos" src="https://www.youtube.com/embed/uR2NRE1ID34" allowfullscreen="">
39   </iframe>
40
41 <h2 class="titulo">Internet de las cosas (IoT)</h2>
42 <p class="parrafo">
43   IoT forma la base de la próxima revolución industrial y es considerado como la próxima gran novedad en la
44   tecnología de automatización del hogar. Hoy en día vemos que hay una invasión de IoT en nuestras vidas,
45   las casas y oficinas e incluso en las grandes ciudades y los medios de transporte. <br>
46   Para 2020, tenemos la intención de ver 50 mil millones de objetos interconectados. Los propietarios
47   podrán controlar todo en sus hogares, desde la iluminación hasta la temperatura.
48   Sin duda alguna, IoT estará redefiniendo las experiencias entre personas, máquinas e instalaciones.
49 </p>
50 <iframe class="videos" src="https://www.youtube.com/embed/gV712Y0SOQ4" allowfullscreen=""></iframe>
51 </section>
52 </main>
53 </body>
54 </html>

```

Figura 219: Captura de pantalla

Fuente. - Elaboración propia

Para visualizar la página, presiona la tecla F5.



Figura 220: Captura de pantalla

Fuente. - Elaboración propia

Resumen

1. Algunos de los cambios más notables en HTML5 es la adición de etiquetas para soportar elementos multimedia. Esto incluye la etiqueta **<audio>** para reproducir sonidos, **<video>** para videos, todo esto dentro del web browser y sin depender de software externo.
2. El elemento **<audio>** trabaja del mismo modo y comparte varios atributos con el elemento **<video>**. **src**, especifica la URL del archivo a ser reproducido; **controls**, activa la interface que cada navegador provee por defecto para controlar la reproducción del audio; **autoplay**, reproducción automática; **loop**, el navegador reproducirá el audio una y otra vez de forma automática; **preload**, este atributo puede tomar tres valores diferentes: none, metadata o auto.
3. Entre los atributos de la etiqueta **<video>**: **poster**, indica la imagen que el navegador debe mostrar mientras el video se está descargando, o hasta que el usuario reproduce el video; **height, width**: indican al navegador el tamaño del video en pixels. Si no se indican estas medidas, el navegador utiliza las medidas definidas en el video de origen, si están disponibles.
4. Una vez que se dispone del video o el audio en los distintos formatos, es necesario indicar todas las ubicaciones de estos formatos, para que sea el navegador el que decida qué formato reproducir. Para especificarlos se tiene que hacer de manera separada utilizando el elemento **<source>**.
5. El elemento HTML inline frame (**<iframe>**) permite incorporar otra página HTML en la página actual. Cada contexto de navegación tiene su propia historia, sesión y documento activo. El contexto de navegación que incluye el contenido implícito se llama contexto de navegación principal. El contexto de navegación de nivel superior (que no tiene padre) es típicamente la ventana del navegador
6. Para incrustar un video de YouTube en un documento HML:

```
<iframe width="560" height="315" src="//www.youtube.com/embed/5AKY58ZSJGs" frameborder="0" allowfullscreen">
</iframe>
```

7. Video Lightbox JS es un programa gratuito que te ayuda a integrar fácilmente vídeo al sitio web, página web o blog, en unos pocos clics sin tener que escribir una sola línea de código.

Recursos

Revisa los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- <https://developer.mozilla.org/es/docs/Web/HTML/UsandoaudioyvideoconHTML5>
- <http://ocw.uv.es/ingenieria-y-arquitectura/programacionmultimedia/t2-3-html5.pdf>
- <https://developer.mozilla.org/es/docs/Web/HTML/Elemento/iframe>
- <http://personales.upv.es/moimacar/download/html5.pdf>

- <https://desarrolladoresweb.org/html5/etiqueta-de-audio-html5/>
- <https://lenguajehtml.com/html/multimedia/etiquetas-html-de-audio/>
- <https://www.anerbarrena.com/atributos-audio-html5-5572/>
- <https://www.campusmvp.es/recursos/post/como-insertar-audio-en-html5.aspx>
- <https://lenguajehtml.com/html/multimedia/etiquetas-html-de-video/>

4.2 ANIMACIONES, TRANSICIONES Y TRANSFORMACIONES

Las nuevas etiquetas HTML5 y CSS3 proporcionan a los diseñadores web cada vez más recursos para el desarrollo de sus sitios web.

CSS3 pretende estandarizar dos nuevas características: las transiciones y las animaciones que actúan sobre el estado de un elemento transformándolo de manera gradual. Con esto se consiguen efectos muy agradables que mejoran notablemente la experiencia del usuario al interactuar con la página.

Estas nuevas características son muy modernas por lo que los navegadores las van incluyendo poco a poco en sus nuevas versiones. De hecho, Internet Explorer todavía no las soporta en su versión 9, aunque Microsoft asegura que las incluirá a partir de la versión 10.

En este capítulo se explica qué son las transiciones y animaciones en CSS; se menciona sus características y aprenderá a transformar y animar elementos de las páginas web únicamente con CSS. Un hecho que antes era impensable sin ayuda de Flash o Javascript.

4.2.1 Animaciones, propiedades

Las animaciones son bastante parecidas a las transiciones. Ambas tienen en común que actúan sobre los valores de las propiedades de un elemento transformándolos gradualmente en otros valores. Sin embargo, las animaciones ofrecen bastante más control que las transiciones en el proceso de transformación del elemento.

Las transiciones se ejecutan cuando se produce un cambio en las propiedades de un elemento. Ya se observó en el punto anterior, cuando el usuario pasa el ratón por encima de un elemento y la css indica que el valor de determinadas propiedades cambia (: hover), entonces se lanza la transición.

Las animaciones son algo distintas, ya que se lanzan explícitamente. No se espera a que suceda un cambio en el estado del elemento como pasaba con las transiciones. Es por ello que, para usar animaciones se necesita "keyframes". Estos "keyframes" describirán el comportamiento que tendrá la animación durante su periodo de ejecución.

Propiedades de las animaciones

Las propiedades de las animaciones son bastante parecidas a las de las transiciones. Para poder aplicar animaciones sobre los elementos considera las siguientes propiedades:

- **animation-name:** es el nombre de la animación que actuará sobre el elemento. Recuerda que en el punto anterior (keyframes) para definir una animación mediante un keyframe se debe añadir también el nombre de la animación. Pues es ese nombre.
- **animation-duration:** el tiempo que tardará la animación desde su inicio hasta su fin.
- **animation-timing-function:** del mismo modo que en las transiciones, esta propiedad indica cómo progresará la animación. Acepta los mismos valores que la propiedad transition-timing-function.
- **animation-iteration-count:** indica el número de veces que se repetirá la animación. Su valor por defecto es 1. El valor "infinite" indica que se repetirá indefinidamente.
- **animation-direction:** indica el orden en el que ejecutará la animación. Acepta los valores "normal" y "alternate". Si se tuviera dos repeticiones de la animación (animation-iteration-count: 2) y el valor de esta propiedad fuese alternate, primero se ejecutaría la

animación en orden normal 0 a 100% y luego al revés 100% a 0. Si fuese normal, las dos veces se ejecutaría de 0 a 100%. El valor por defecto es normal.

- **animation-play-state:** indica si la animación está corriendo o pausada. Acepta los valores running (valor por defecto) y paused.
- **animation-delay:** igual que en las transiciones, indica el retardo con el que se iniciará la animación. Por defecto 0.
- **animation-fill-mode:** pues básicamente lo que hace esta propiedad es decidir si los valores de las propiedades que ha modificado la animación vuelven a su estado original (antes de la animación) o no. Realmente vale para ver qué se hace con los valores de las propiedades fuera del periodo de duración de la animación. Acepta los valores: "none", "forwards", "backwards" y "both". El valor por defecto es none que indica que cuando acabe la animación el elemento recuperará los valores que tenían de sus propiedades antes de iniciarse la animación.
- **animation:** permite unificar las propiedades anteriores en una sola. Sería algo así, animation: mi-animacion 3s ease 0.5s 1 normal none.

4.2.2 Transiciones, propiedades

Las transiciones es el efecto de un cambio en el estado de un elemento de forma gradual, en un intervalo de tiempo. Cuando se habla de cambio de estado, se refiere a cambios en los valores de sus propiedades.

Las transiciones únicamente son soportadas por los navegadores: Firefox, Chrome, Safari y Opera. Internet Explorer (al menos hasta su versión 9) no las soporta.

Propiedades de las Transiciones

Para aplicar una transición sobre un determinado elemento, se cuenta con las siguientes propiedades:

- **transition-property:** determina la(s) propiedad(es) a ser animadas; puede ser cualquier propiedad CSS. El valor por defecto es "all" que la transición se aplicará sobre todas las propiedades del elemento. Un valor válido para esta propiedad sería "width".
- **transition-duration:** indica la duración de la transición del inicio al fin en segundos. Por defecto el valor es 0, ejemplos de valores válidos para esta propiedad pueden ser: 2s (2 segundos), 300ms (300 milisegundos), etc.
- **transition-timing-function:** especifica la velocidad que se desarrollará la transición utilizando las llamadas curvas de Bezier. Así se puede programar la transición para que empiece a una velocidad, luego frene y después se acelere o darle movimiento. Los valores que acepta esta propiedad son los siguientes:
 - **linear:** se produce en forma de progresión lineal. Tarda lo mismo desde que empieza hasta que acaba.
 - **ease:** la transición se produce en forma lenta al principio, luego más rápido, y termina otra vez lenta (valor por defecto).
 - **ease-in:** la transición comienza lenta y luego va más rápido.
 - **ease-out:** la transición comienza más rápido y termina lenta.
 - **ease-in-out:** la transición comienza y acaba lenta. Igual que 'ease' pero de forma más notable.
- **cubic-bezier(n, n, n, n):** permite definir tu propio intervalo de progresión. n: valor entre 0 y 1.
 - **transition-delay:** tiempo en que la animación debe pausarse antes de comenzar. Por defecto es 0.

- ***transition:*** permite unificar todas las propiedades anteriores en una sola.

La lista de propiedades que se pueden modificar mediante transiciones:

- background-color (color)
- background-image (solo gradientes)
- background-position (porcentaje y longitud)
- border-bottom-color (color)
- border-bottom-width (longitud)
- border-color (color)
- border-left-color (color)
- border-left-width (longitud)
- border-right-color (color)
- border-right-width (longitud)
- border-spacing (longitud)
- border-top-color (color)
- border-top-width (longitud)
- border-width (longitud)
- bottom (longitud y porcentaje)
- color (color)
- crop (rectángulo)
- font-size (longitud y porcentaje)
- font-weight (número)
- grid-* (diversos valores)
- height (longitud y porcentaje)
- left (longitud y porcentaje)
- letter-spacing (longitud)
- line-height (número, longitud y porcentaje)
- margin-bottom (longitud)
- margin-left (longitud)
- margin-right (longitud)
- margin-top (longitud)
- max-height (longitud y porcentaje)
- max-width (longitud y porcentaje)
- min-height (longitud y porcentaje)
- min-width (longitud y porcentaje)
- opacity (número)
- outline-color (color)
- outline-offset (entero)
- outline-width (longitud)
- padding-bottom (longitud)
- padding-left (longitud)
- padding-right (longitud)
- padding-top (longitud)
- right (longitud y porcentaje)
- text-indent (longitud y porcentaje)
- text-shadow (sombra)
- top (longitud y porcentaje)
- vertical-align (palabras clave, longitud y porcentaje)
- visibility (visibilidad)
- width (longitud y porcentaje)

- word-spacing (longitud y porcentaje)
- z-index (entero)
- zoom (número)

Ejemplos:

En este ejemplo de transiciones se va a aplicar un zoom en un texto. Cuando un usuario pase el ratón por encima de un texto contenido en un párrafo, el tamaño de letra se incrementa y cambia el color de la letra.

- La transición se aplique a la propiedad font-size (transition-property: font-size).
- La transición dure 300 milisegundos (transition-duration: 300s).
- La transición se produzca de forma siempre a la misma velocidad (transition-timing-function: linear).
- La transición se produzca sin retardo (transition-delay: 0s).

Primero, define la clase zoom, en dicha clase define la transición del texto. Esta transición es aplicada para los navegadores Firefox, Chrome, Safari.

```
p.zoom {
    font-size:14px;
    color:blue;
    text-align:justify;
    width:250px;
    height:auto;
    /* Firefox */
    -moz-transition: font-size 300ms linear 0s;
    /* Chrome / Safari */
    -webkit-transition: font-size 300ms linear 0s;
    /* Default */
    transition: font-size 300ms linear 0s;
}
```

Luego, define las propiedades al evento hover de la clase zoom, donde al pasar el cursor sobre el texto aumenta el tamaño de la letra y cambia el color de la letra.

```
p.zoom:hover {
    font-size:20px;
    color:red;
    text-align:center;
}
```

En la página, define la etiqueta <p> definiendo la clase zoom, para aplicar el zoom al texto redactado dentro del párrafo <p>.

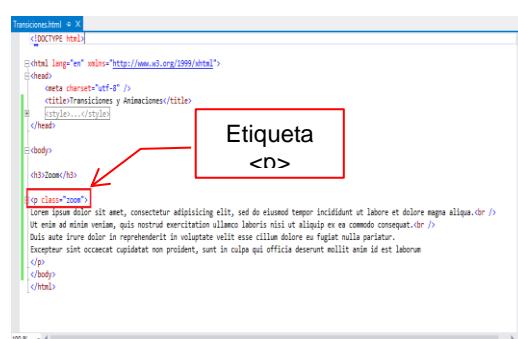


Figura 221: Transiciones
Fuente. - Elaboración propia

Al finalizar el diseño de la página, ejecútala en un navegador, Chrome. Al inicio se visualiza como la página de la izquierda; pero al colocar el cursor sobre el texto, se muestra como la página de la derecha.

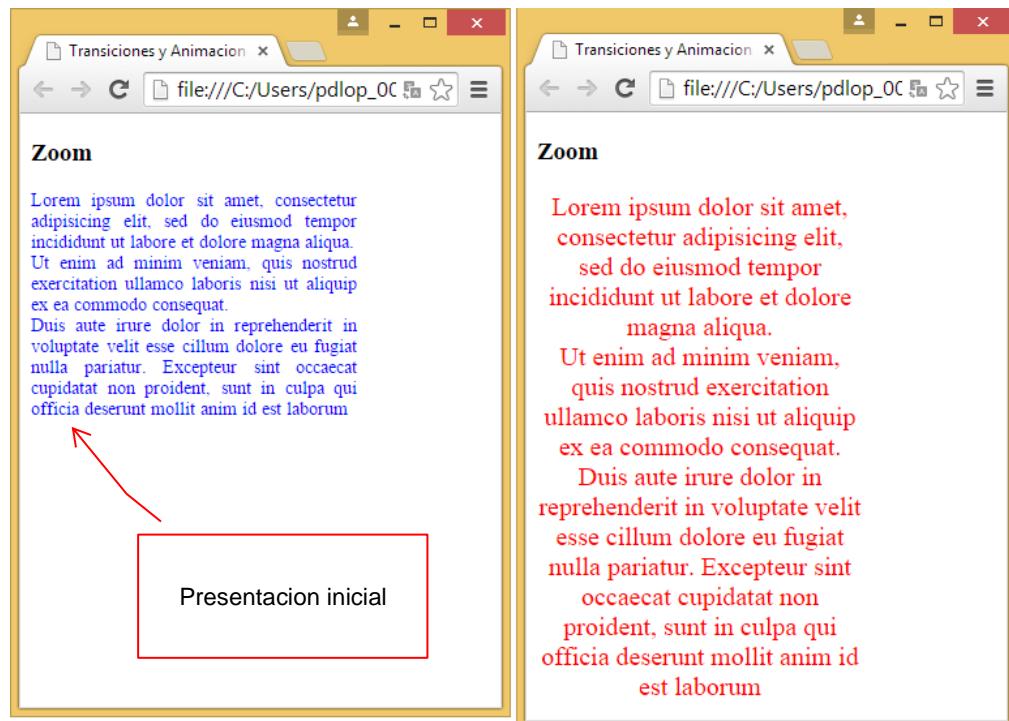


Figura 222: Transiciones
Fuente. - Elaboración propia

El siguiente ejemplo de transición es aplicado a las imágenes. Produce una transición en las propiedades de las imágenes: width, height y border cuando el usuario pase el mouse por encima de la imagen.

Como primer paso define las propiedades de la transición:

- La transición debe aplicarse a todos los elementos que cambien (transition-property: all).
- La transición dure 2 segundos (transition-duration: 1s).
- La transición se produzca de forma lenta al principio, luego rápido y acabe lenta (transition-timing-function: ease).
- La transición se produzca con un retardo de 500ms tras pasar el mouse por encima de la imagen (transition-delay: 0.5s).

```
img.growth {  
    width:150px;  
    height:130px;  
    border:1px solid;  
    /* Firefox */  
    -moz-transition-property: all;  
    -moz-transition-duration: 2s;  
    -moz-transition-timing-function: ease;  
    -moz-transition-delay: 0.5s;  
  
    /* Chrome / Safari */  
    -webkit-transition-property: all;  
    -webkit-transition-duration: 2s;  
    -webkit-transition-timing-function: ease;  
    -webkit-transition-delay: 0.5s;  
  
    /* Default */  
    transition-property: all;  
    transition-duration: 2s;  
    transition-timing-function: ease;  
    transition-delay: 0.5s;  
}
```

A continuación, define las propiedades del evento hover de la imagen, tal como se muestra.

```
img.growth:hover {  
    width:250px;  
    height:230px;  
    border:2px solid;  
}
```

En la página, define la etiqueta `` definiendo la clase growth, para aplicar el zoom a la imagen.



```

Transiciones.html ✘ X
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Transiciones y Animaciones</title>
    <style>...</style>
</head>
<body>
    <h3>Imagenes</h3>
    
    
    
</body>
</html>

```

Figura 223: Transiciones
Fuente. - Elaboración propia

Al finalizar el diseño de la página, ejecútala en un navegador, Chrome. Al inicio se visualiza como la página de la izquierda; pero al colocar el cursor sobre una imagen, se muestra como la página de la derecha.

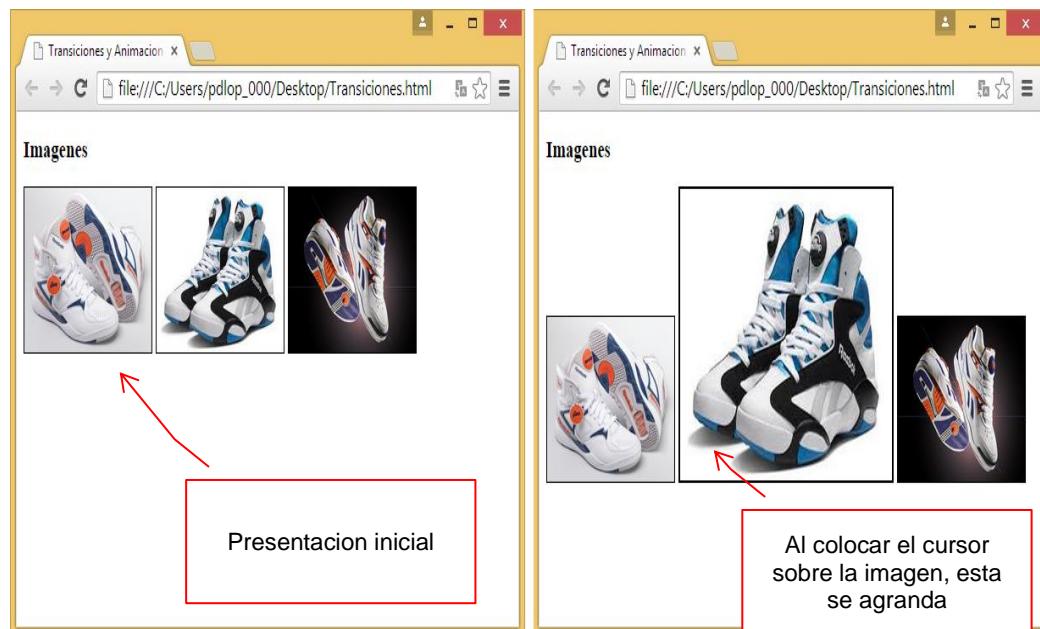


Figura 224: Transiciones
Fuente. - Elaboración propia

4.2.3 Transformaciones, propiedades

Las transformaciones es uno de los elementos más interesantes que se introducen en CSS3 para convertir el lenguaje de hojas de estilo en un sistema capaz de realizar todo tipo de efectos visuales, incluido 2D y 3D, siendo estas últimas considerablemente más complejas al tener en cuenta los tres ejes de coordenadas.

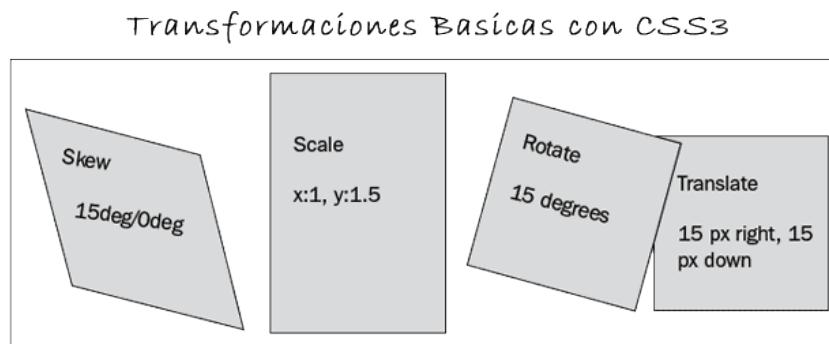


Figura 225: Transformaciones
Fuente. - Tomado de <https://www.csslab.cl/2011/01/31/animaciones-y-transformaciones-con-css3/>

De las transformaciones de CSS3 en 2D, las más usadas son:

- **Rotate.** Permite rotar un elemento dándole un ángulo de giro en grados.
- **Skew:** Desplazamiento de los ejes horizontales.
- **Scale.** Permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales.
- **Translate.** Permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.

Ejemplo:

En este ejemplo de transición es aplicado a las imágenes. El objetivo es que se produzca una transformación aumentando la escala de las imágenes 50% más cuando el usuario pase el mouse por encima de la imagen.

Como primer paso define las propiedades de la transición:

- La transición debe aplicarse a todos los elementos que cambien: all.
- La transición dure 2 segundos (transition-duration: 1s).
- La transición se produzca de forma lenta al principio y acabe lenta (transition-timing-function: ease).

```
img.growth {
    width:150px;
    height:130px;
    border:1px solid;

    -webkit-transition:all 2s ease-in-out;
    -moz-transition:all 2s ease-in-out;
    transition: all 2s ease-in-out;
}
```

A continuación, define las propiedades del evento hover de la imagen, donde al pasar el cursor sobre la imagen, esta aumentara su tamaño 50% más.

```
img.growth:hover {
    transform: scale(1.5,1.5);
}
```

En la página, define la etiqueta **** definiendo la clase growth, para aplicar el zoom a la imagen.

```
Transiciones.html ✘ X
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Transiciones y Animaciones</title>
    <style>...</style>
</head>
<body>
    <h3>Imagenes</h3>
    
    
    
</body>
</html>
```

Figura 226: Transiciones
Fuente. - Elaboración propia

Al finalizar el diseño de la página, ejecútala en un navegador, Chrome. Al inicio se visualiza como la página de la izquierda; pero al colocar el cursor sobre una imagen, se muestra como la página de la derecha.

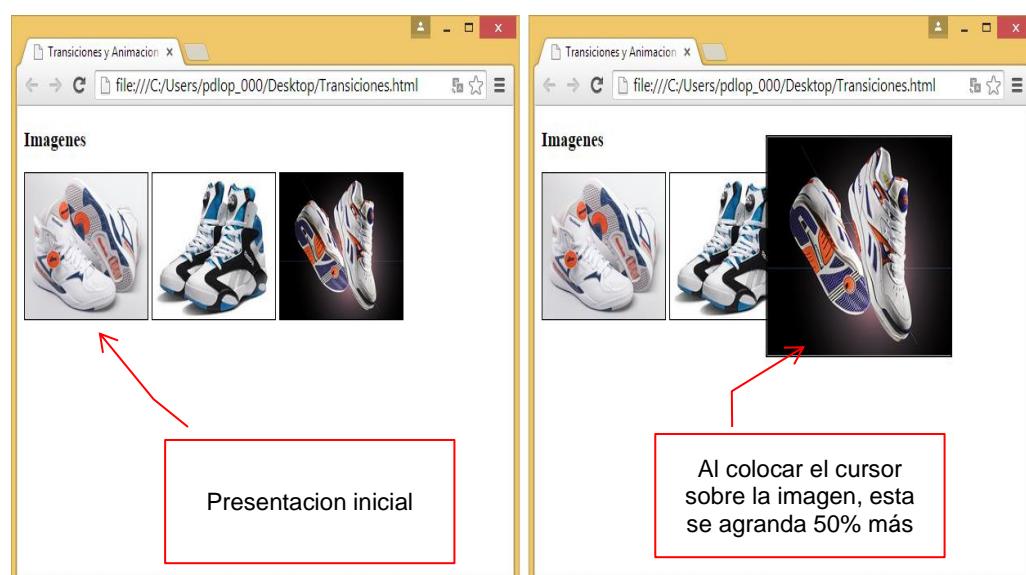


Figura 227: Transformaciones
Fuente. - Elaboración propia

LABORATORIO 1

Implementa una página, la cual contenga animaciones, transiciones y transformaciones al momento de ejecutar el evento hover.

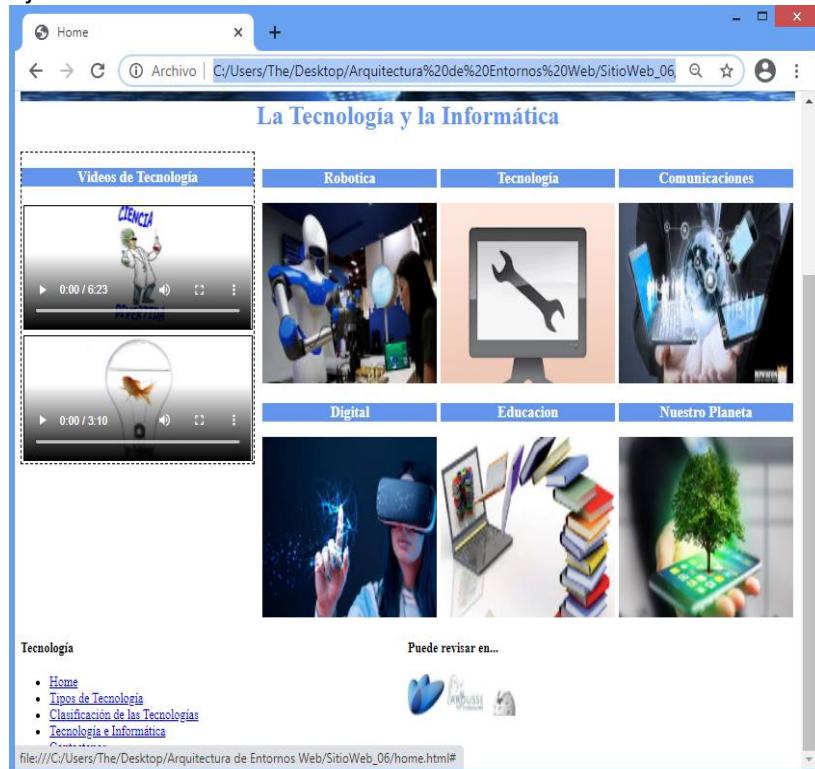


Figura 228: Captura de pantalla

Fuente. - Elaboración propia

Agregando la pagina HTML y el archivo CSS

Antes de iniciar el desarrollo de la página, primero agrega los archivos **CSS** y **HTML**, en el **sitioWeb_6**, tal como se muestra.

```

EXPLORER ... home.html # estilos-home.css
OPEN EDITORS
SITIOWEB_06
audios
css
# estilos-home.css
imagenes
js
videos
home.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link rel="stylesheet" href="css/estilo-home.css">
</head>
<body>
    <main>...
    </main>
</body>
</html>

```

Figura 229: Captura de pantalla

Fuente. - Elaboración propia

Enlazando la hoja de estilo

A continuación, enlaza la hoja de estilo CSS a la página HTML.

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar: Shows a project structure with a folder named "SITIOWEB_06" containing "audios", "css" (with a file "# estilos-home.css"), "imagenes", "js", and "videos". A file "home.html" is also listed.
- OPEN EDITORS** tab: Shows two files: "home.html" and "# estilos-home.css".
- home.html** content (lines 1-148):


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Home</title>
7      <link rel="stylesheet" href="css/estilos-home.css">
8  </head>
9  <body>
10     <main>...
146    </main>
147  </body>
148  </html>
```
- # estilos-home.css** content (lines 1-27):


```

1 main{
2     width: 100%;
3     height: auto;
4     float: left;
5 }
6 #header-rs{
7     width: 98%;
8     padding-left: 1%;
9     padding-right: 1%;
10    height: auto;
11    float: left;
12    display: grid;
13    grid-template-columns:90% 1fr 1fr 1fr;
14 }
```
- Annotations:**
 - A red box highlights the line `<link rel="stylesheet" href="css/estilos-home.css">` in the HTML code.
 - A red callout bubble points to this line with the text: "Enlazando la hoja de estilo a la pagina HTML".

Figura 230: Captura de pantalla

Fuente. - Elaboración propia

Definiendo las reglas en el CSS: redes sociales

En el CSS define las reglas para los selectores de la cabecera de las redes sociales. Además, incluye la regla para el hover de rs, donde al pasar el mouse la imagen gire 360 grados sobre su eje Y durante 1 segundo.

The screenshot shows a code editor interface with the following details:

- OPEN EDITORS** tab: Shows two files: "home.html" and "# estilos-home.css".
- home.html** content (lines 1-27):


```

1 main{
2     width: 100%;
3     height: auto;
4     float: left;
5 }
6 #header-rs{
7     width: 98%;
8     padding-left: 1%;
9     padding-right: 1%;
10    height: auto;
11    float: left;
12    display: grid;
13    grid-template-columns:90% 1fr 1fr 1fr;
14 }
```
- # estilos-home.css** content (lines 1-27):


```

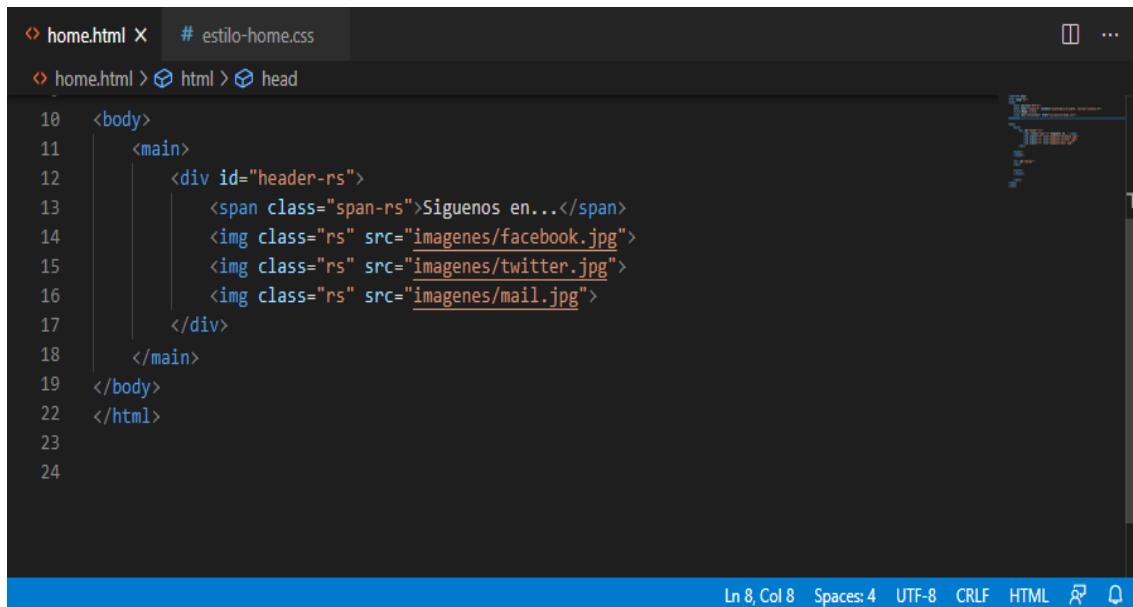
1 main{
2     width: 100%;
3     height: auto;
4     float: left;
5 }
6 #header-rs{
7     width: 98%;
8     padding-left: 1%;
9     padding-right: 1%;
10    height: auto;
11    float: left;
12    display: grid;
13    grid-template-columns:90% 1fr 1fr 1fr;
14 }
15 .rs{
16     width: 40px;
17     height: 40px;
18     transform: rotateY(0deg);
19     transition: all ease 0.5s ;
20 }
21 .rs:hover{
22     transform: rotateY(360deg);
23     transition: all ease 1s ;
```
- Annotations:**
 - A red box highlights the class selector ".rs" in the CSS code.
 - A red callout bubble points to this line with the text: "Reglas al selector de class rs: definimos transicion y transform".
 - A red box highlights the pseudo-class selector ".rs:hover" in the CSS code.
 - A red callout bubble points to this line with the text: "Reglas al hover del selector de class rs: al pasar el mouse, gira por el eje Y 360 grados en el lapso de 1 seg".

Figura 231: Captura de pantalla

Fuente. - Elaboración propia

Agregando los elementos a la página HTML

En este proceso define las etiquetas de la página hml y asigna el selector de class a las etiquetas `` de redes sociales y a la etiqueta ``.



```
home.html X # estilo-home.css
home.html > html > head
10  <body>
11    <main>
12      <div id="header-rs">
13        <span class="span-rs">Síguenos en...</span>
14        
15        
16        
17      </div>
18    </main>
19  </body>
20 </html>
21
22
23
24
```

Ln 8, Col 8 Spaces: 4 UTF-8 CRLF HTML ⚙️ 📁

Figura 232: Captura de pantalla

Fuente. - Elaboración propia

Presiona **F5** para visualizar el proceso de la animación, al pasar el mouse sobre las imágenes de las redes sociales, éstas girarán en su eje Y.

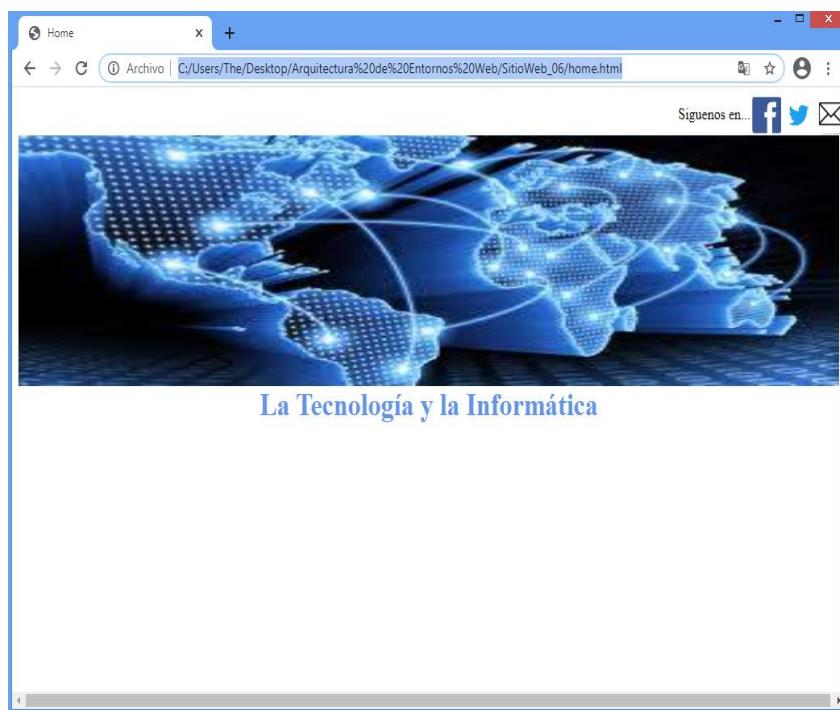


Figura 233: Captura de pantalla

Fuente. - Elaboración propia

Definiendo las reglas en el CSS: header

En el CSS define las reglas para el selector de bloque **<header>**. Define los selectores de id: h1-header asignado para el título e img-header, asignado para la imagen.

```

home.html      # estilo-home.css

css > # estilo-home.css > #header-rs

26
27 .span-rs{
28     padding-top: 10px; ← Regla del selector de class: span-rs, asignado al texto asociado a las redes sociales
29     text-align: right;
30 }
31
32 header{
33     width: 100%; ← reglas para el selector de etiqueta y bloque: header
34     height: auto;
35     float: left;
36 }
37
38 #h1-header{
39     text-align: center; ← reglas para el selector de id: h1-header, titulo
40     color: cornflowerblue;
41 }
42
43 #img-header{
44     width: 100%; ← Reglas para el Selector de id: img-header, imagen de cabecera
45     height: 250px;
46     float: left;
47 }
48

```

Ln 14, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚙️ 🗑️

Figura 234: Captura de pantalla

Fuente. - Elaboración propia

Agregando los elementos a la página HTML: <header>

En este proceso define las etiquetas del **<header>** de la página html y asigna el selector de id a las etiquetas **** y a la etiqueta **<h1>** del título.

```

home.html X  # estilo-home.css

home.html > html > head > link

10 <body>
11   <main>
12     <div id="header-rs">
13       <span class="span-rs">Síguenos en...</span>
14       
15       
16       
17     </div>
18
19   <header> ← Red box surrounds this section
20     
21     <h1 id="h1-header">La Tecnología y la Informática</h1>
22   </header>
23 </main>

```

Ln 7, Col 55 Spaces: 4 UTF-8 CRLF HTML ⚙️ 🗑️

Figura 235: Captura de pantalla

Fuente. - Elaboración propia

Para visualizar la página, presiona **F5**.



Figura 236: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas en el CSS: cuerpo y aside

En el CSS define las reglas para el selector de bloque `<div>` y `<aside>`. Luego, define el selector de class: título asignado para el título y videos asignado para la etiqueta `<video>`.

```

48
49 #cuerpo{
50     width: 100%;
51     height: auto;
52     margin-top: 5px;
53     margin-bottom: 5px;
54     float: left;
55 }
56
57 aside{
58     width: 30%;
59     height: auto;
60     border: 1px dashed;
61     float: left;
62 }
63
64 .titulo{
65     text-align: center;
66     color: white;
67     background-color: cornflowerblue;
68 }
69
70 .videos{
71     width: 97.5%;
72     height: 150px;
73     margin: 1%;
74     border: 1px solid;
75     float: left;
76 }
77

```

reglas para el selector id: cuerpo, asignado para el bloque del cuerpo de la página

reglas para el selector de bloque: aside, barra lateral

reglas para el selector de class: titulo, asignado para el <heading> del título de los videos

reglas para el selector de class: videos, asignado para la etiqueta <video>

Figura 237: Captura de pantalla

Fuente. - Elaboración propia

Agregando los elementos a la página HTML: cuerpo y <aside>

En este proceso define las etiquetas del bloque <div> de la página hml y el selector de bloque <aside>. En el contenido del <aside>, define una etiqueta heading <h3> con su selector de class y define las etiquetas <video> para reproducir un video relacionado al tema.

```

10 <body>
11     <main>
12     <div id="header-rs">...
13     </div>
14
15     <header>...
16     </header>
17
18     <div id="cuerpo">
19         <aside>
20             <h3 class="titulo">Videos de Tecnología</h3>
21
22             <video class="videos" src="videos/video1.mp4"
23                 poster="imagenes/tecnologia-ciencia.jpg" controls></video>
24
25             <video class="videos" src="videos/definicion.mp4"
26                 poster="imagenes/tecnologia-innovacion.jpg" controls></video>
27         </aside>
28     </div>
29
30     </main>
31
32 </body>
33 </html>

```

Figura 238: Captura de pantalla

Fuente. - Elaboración propia

Para visualizar la página, presiona **F5**.

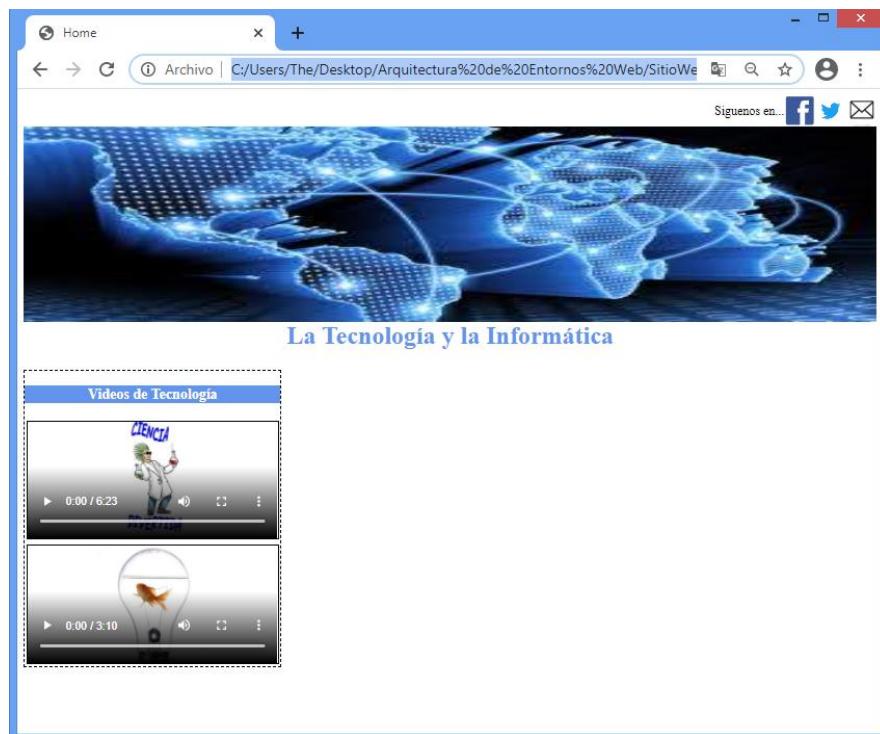


Figura 239: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas en el CSS: cuerpo y section

En el CSS define las reglas para el selector de bloque **<section>**, en dicho bloque se visualiza como una cuadrícula formada por dos filas y tres columnas. Define el selector de bloque **<article>**, el cual oculta los elementos que estén fuera de su dimensión. Define el selector de class: **img-article** y su evento hover para dar animación al elemento ****.

```

home.html # estilo-home.css X
css > # estilo-home.css > ...
79  section{
80      width: 69%;
81      height: auto;
82      float: right;
83      display: grid;
84      grid-template-rows: 2;
85      grid-template-columns: 1fr 1fr 1fr;
86  }
87
88  article{
89      width: 98%;
90      height: 280px;
91      margin: 1%;
92      overflow: hidden;
93  }
94
95  article:hover .img-article{
96      height: 0px;
97      transition: all ease 1s;
98  }
99
100 .img-article{
101     width: 100%;
102     height: 250px;
103     float: left;
104     transition: all ease 0.5s ;
105 }
106
107 .parrafo{
108     text-align: justify;
109     font-size: 12px;
110 }
111

```

reglas para el selector de bloque: article, definimos la propiedad overflow: hidden para ocultar los elementos que desbordan del tamaño del bloque

reglas del hover del article, al pasar el mouse, la imagen se contrae en el lapso de un segundo

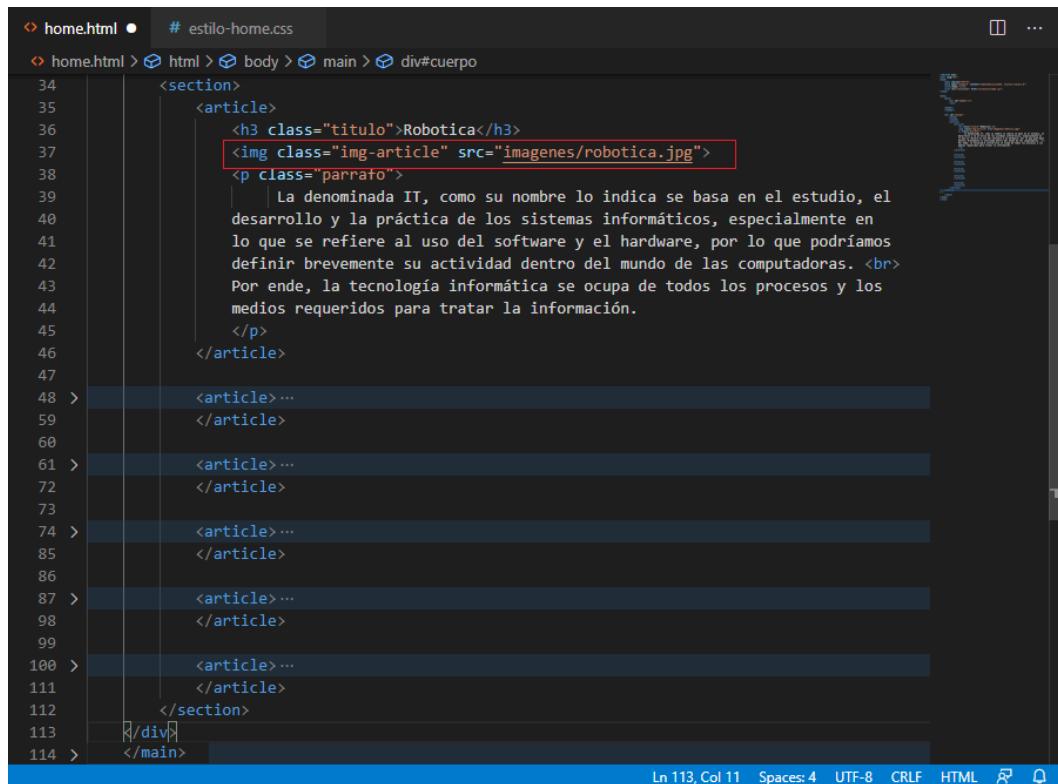
reglas del selector de class: img-article, imagen del bloque <section>, asignando una transition de 0.5s, al momento que el mouse salga del elemento

Figura 240: Captura de pantalla

Fuente. - Elaboración propia

Agregando los elementos a la página HTML: <section>

En este proceso define las etiquetas dentro del bloque <div>, el bloque <section>. En el contenido del bloque, define selectores de bloque <article>, donde cada bloque contendrá: un título, imagen y un párrafo (estará oculto). Al pasar el mouse, se oculta la imagen y se visualiza el texto.



```

<!-- home.html -->
<!-- estilos-home.css -->

<home.html> <html> <body> <main> <div id="cuerpo">
  34 <section>
  35   <article>
  36     <h3 class="titulo">Robotica</h3>
  37     
  38     <p class="parrafo">
  39       La denominada IT, como su nombre lo indica se basa en el estudio, el
  40       desarrollo y la práctica de los sistemas informáticos, especialmente en
  41       lo que se refiere al uso del software y el hardware, por lo que podríamos
  42       definir brevemente su actividad dentro del mundo de las computadoras. <br>
  43       Por ende, la tecnología informática se ocupa de todos los procesos y los
  44       medios requeridos para tratar la información.
  45   </p>
  46 </article>
  47 <article>...
  48 </article>
  49 <article>...
  50 </article>
  51 <article>...
  52 </article>
  53 <article>...
  54 </article>
  55 <article>...
  56 </article>
  57 <article>...
  58 </article>
  59 <article>...
  60 </article>
  61 <article>...
  62 </article>
  63 <article>...
  64 </article>
  65 <article>...
  66 </article>
  67 <article>...
  68 </article>
  69 <article>...
  70 </article>
  71 <article>...
  72 </article>
  73 <article>...
  74 </article>
  75 <article>...
  76 </article>
  77 <article>...
  78 </article>
  79 <article>...
  80 </article>
  81 <article>...
  82 </article>
  83 <article>...
  84 </article>
  85 <article>...
  86 </article>
  87 <article>...
  88 </article>
  89 <article>...
  90 </article>
  91 <article>...
  92 </article>
  93 <article>...
  94 </article>
  95 <article>...
  96 </article>
  97 <article>...
  98 </article>
  99 <article>...
 100 </article>
111 </section>
112 </div>
113 </main>
114 </body>

```

Ln 113, Col 11 Spaces: 4 UTF-8 CRLF HTML ⚡ D

Figura 241: Captura de pantalla

Fuente. - Elaboración propia

Presiona la tecla **F5**, para visualizar la página HTML.

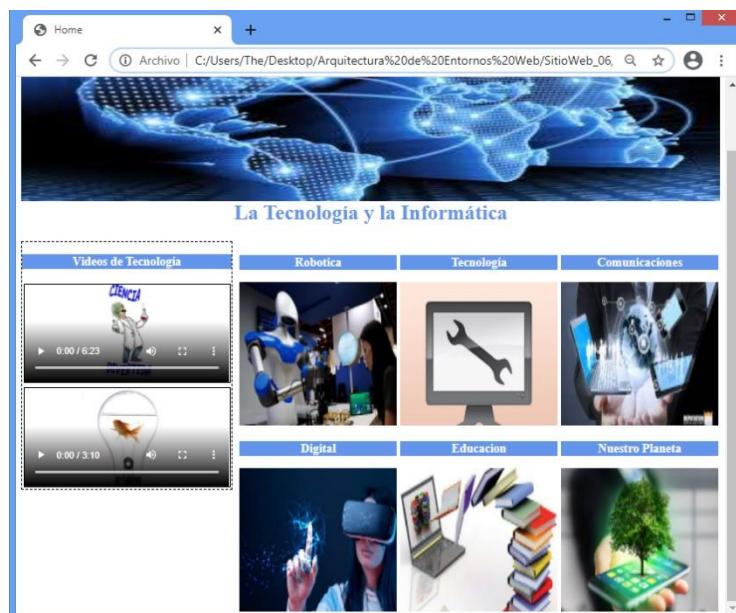


Figura 242: Captura de pantalla

Fuente. - Elaboración propia

Al pasar el mouse sobre el bloque <article>, la imagen se contrae y se muestra el texto, tal como se muestra.

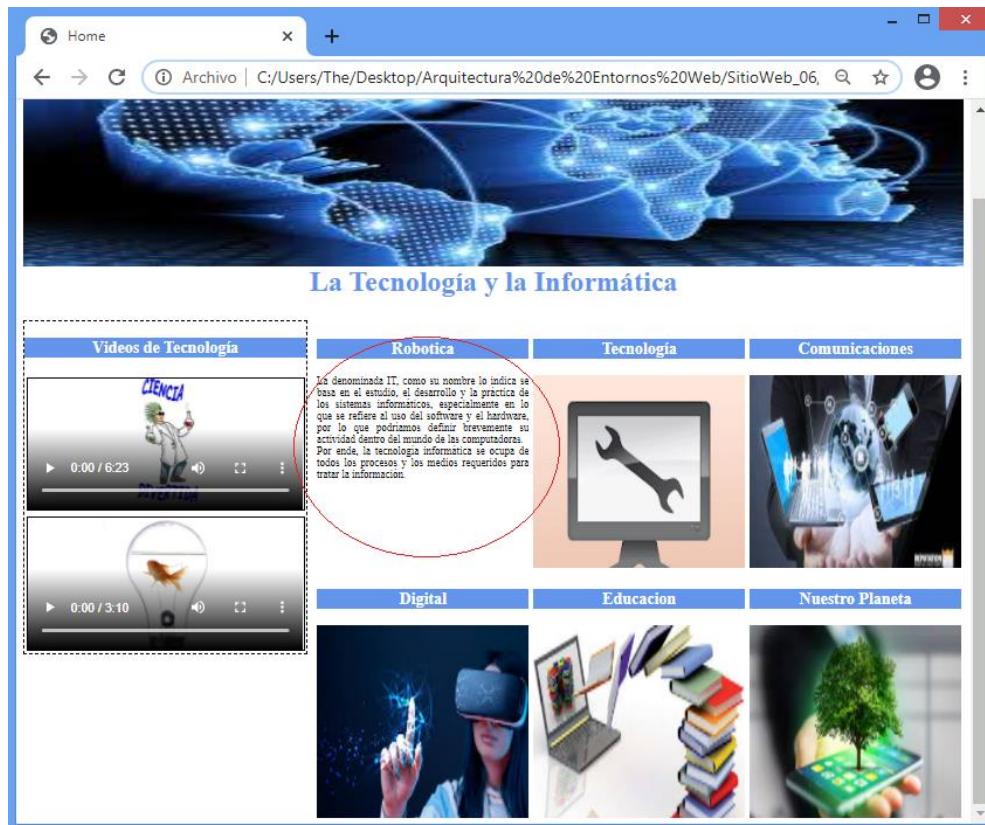


Figura 243: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas en el CSS: footer y <div-footer>

En el CSS define las reglas para el selector de bloque <footer>, en dicho bloque se visualiza como una cuadrícula formada por dos columnas. Luego, define el selector de bloque <div> div-footer, el cual contiene elementos del pie.

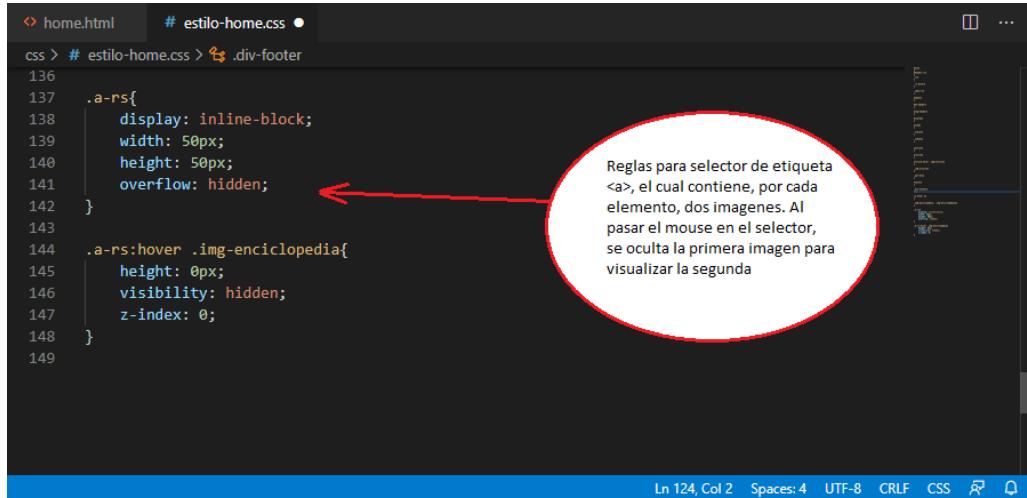
```
home.html # estilo-home.css ●
css > # estilo-home.css > ...
112 footer{
113   width: 100%;
114   height: auto;
115   display: grid;
116   grid-template-columns: 1fr 1fr;
117   float: left;
118 }
119
120 .div-footer{
121   width: 100;
122   height: auto;
123   float: left;
124 }
125
126 li:hover a{
127   background-color: cornflowerblue;
128   color: white;
129   font-style: oblique;
130 }
131
132 .img-enciclopedia, .img-enciclopedia1{
133   width: 50px;
134   height: 50px;
135 }
```

Selector de clase div-footer, bloque contenido en el footer

Evento hover del selector <a>, al pasar el mouse cambiamos el estilo del elemento

Selector de class para las imágenes del footer

Figura 244: Captura de pantalla
Fuente. - Elaboración propia



```

home.html # estilos-home.css
css > # estilos-home.css > .div-footer
136
137 .a-rs{
138   display: inline-block;
139   width: 50px;
140   height: 50px;
141   overflow: hidden;
142 }
143
144 .a-rs:hover .img-encyclopedia{
145   height: 0px;
146   visibility: hidden;
147   z-index: 0;
148 }
149

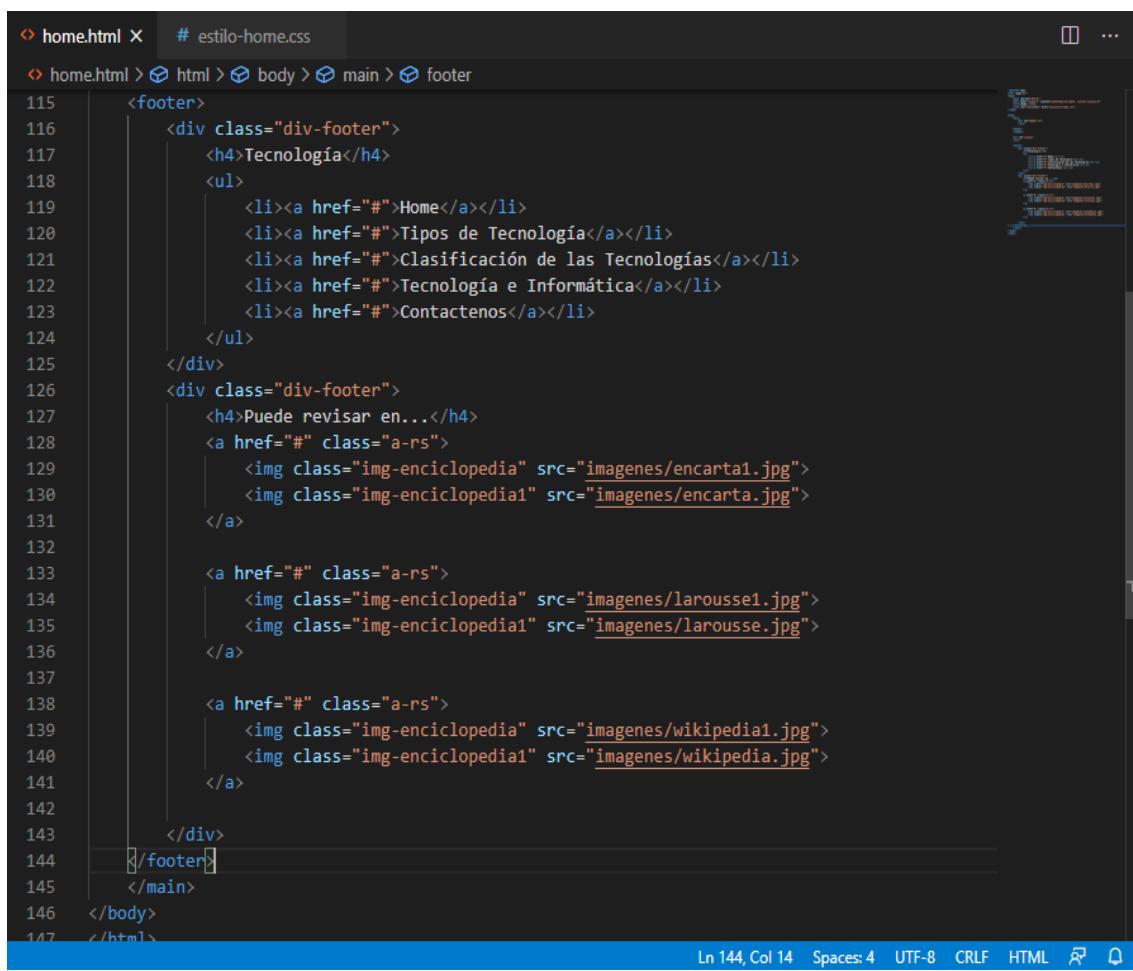
```

Reglas para selector de etiqueta <a>, el cual contiene, por cada elemento, dos imágenes. Al pasar el mouse en el selector, se oculta la primera imagen para visualizar la segunda

Figura 245: Captura de pantalla
Fuente. - Elaboración propia

Agregando los elementos a la página HTML: <footer>

En este proceso define las etiquetas dentro bloque <footer>, el bloque <div>. En el contenido del bloque, define los elementos para visualizar, en el lado izquierdo, la lista de opciones de menú, en el lado derecho las imágenes donde se puede direccionar a otras páginas de revisión de tecnología.



```

home.html X # estilos-home.css
home.html > html > body > main > footer
115 <footer>
116   <div class="div-footer">
117     <h4>Tecnología</h4>
118     <ul>
119       <li><a href="#">Home</a></li>
120       <li><a href="#">Tipos de Tecnología</a></li>
121       <li><a href="#">Clasificación de las Tecnologías</a></li>
122       <li><a href="#">Tecnología e Informática</a></li>
123       <li><a href="#">Contactenos</a></li>
124     </ul>
125   </div>
126   <div class="div-footer">
127     <h4>Puede revisar en...</h4>
128     <a href="#" class="a-rs">
129       
130       
131     </a>
132
133     <a href="#" class="a-rs">
134       
135       
136     </a>
137
138     <a href="#" class="a-rs">
139       
140       
141     </a>
142
143   </div>
144 </footer>
145 </main>
146 </body>
147 </html>

```

Figura 246: Captura de pantalla
Fuente. - Elaboración propia

Presiona la tecla **F5**, para visualizar la página.

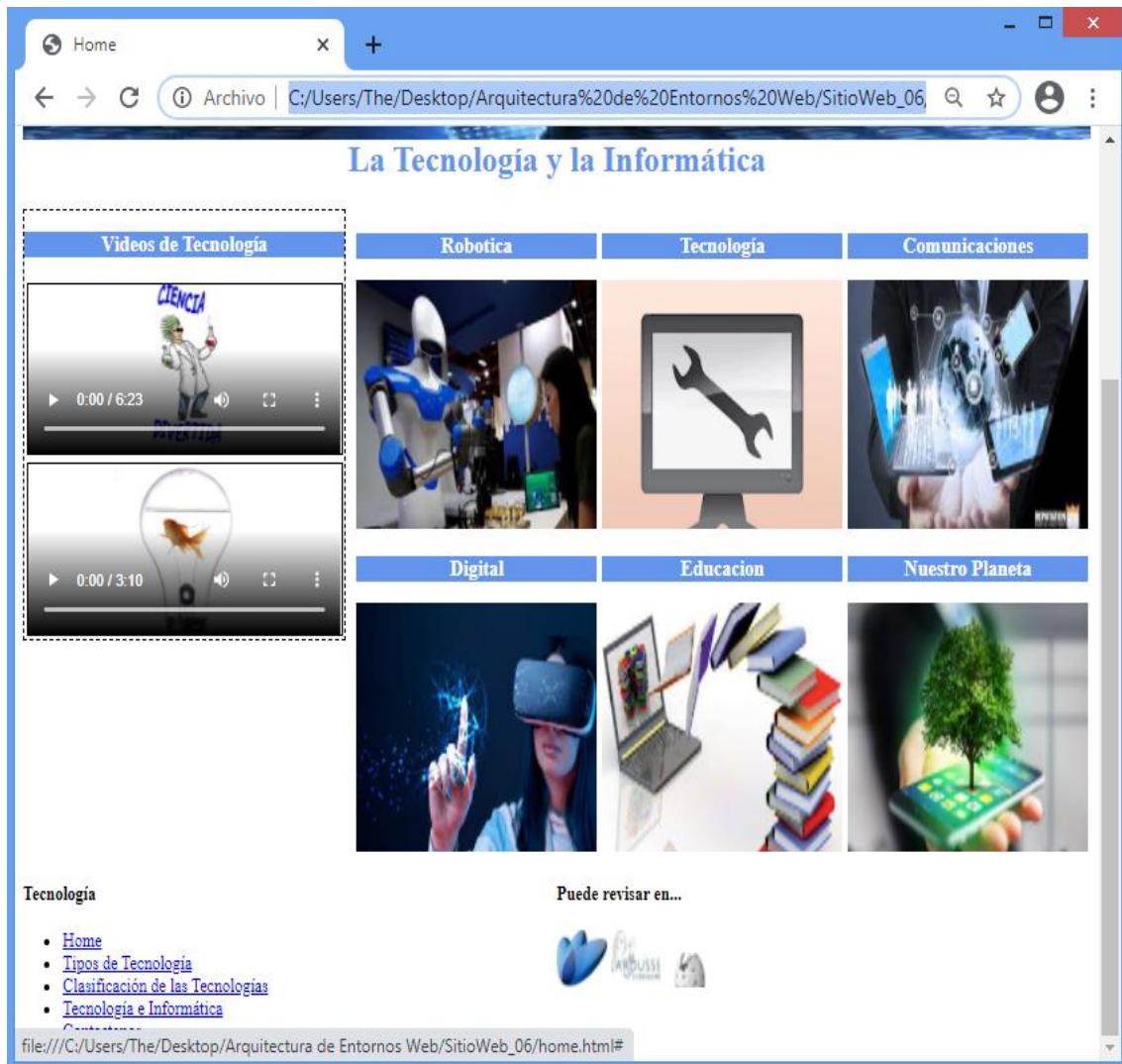


Figura 247: Captura de pantalla
Fuente. - Elaboración propia

Resumen

1. CSS3 pretende estandarizar dos nuevas características: las transiciones y las animaciones que actúan sobre el estado de un elemento transformándolo de manera gradual. Con esto se consiguen efectos muy agradables que mejoran notablemente la experiencia del usuario al interactuar con la página.
2. Las transiciones es el efecto de un cambio en el estado de un elemento, de forma gradual en un intervalo de tiempo. Cuando se habla de cambio de estado, esto hace referencia a cambios en los valores de sus propiedades. Las transiciones únicamente son soportadas por los navegadores: Firefox, Chrome, Safari y Opera. Internet Explorer (al menos hasta su versión 9) no las soporta.
3. Para aplicar una transición sobre un determinado elemento, se debe tener en cuenta las siguientes propiedades: transition-property: determina la(s) propiedad(es) a ser animadas; puede ser cualquier propiedad CSS. transition-duration: indica la duración de la animación del inicio al fin en segundos. Por defecto el valor es 0. transition-timing-function: especifica la velocidad que se desarrollará la transición utilizando las llamadas curvas de Bezier. Así se puede programar la transición para que empiece a una velocidad, luego frene y después acelere o dar movimiento. transition-delay: tiempo en que la animación debe pausarse antes de comenzar. Por defecto es 0. Transition: permite unificar todas las propiedades anteriores en una sola.
4. Las animaciones son bastante parecidas a las transiciones. Ambas tienen en común que actúan sobre los valores de las propiedades de un elemento transformándolos gradualmente en otros valores. Sin embargo, las animaciones ofrecen bastante más control que las transiciones en el proceso de transformación del elemento. Las animaciones son algo distintas, ya que se lanzan explícitamente. No se espera a que suceda un cambio en el estado del elemento como pasaba con las transiciones. Es por ello que, para usar animaciones se necesita "keyframes". Estos "keyframes" describirán el comportamiento que tendrá la animación durante su periodo de ejecución.
5. Las transformaciones de CSS3 pueden ser en 2D y en 3D, siendo estas últimas considerablemente más complejas al tener que tener en cuenta los tres ejes de coordenadas.
6. De las transformaciones de CSS3 en 2D, las más usadas son: Rotate, la cual permite rotar un elemento dándole un ángulo de giro en grados. Skew: desplazamiento de los ejes horizontales. Scale, permite escalar un elemento, toma valores positivos y negativos y se le pueden poner decimales. Translate, permite trasladar un elemento a la vez en el eje de las X y de las Y, dándole las coordenadas iniciales y finales.

Recursos

Puede revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- <http://www.csslab.cl/2011/01/31/animaciones-y-transformaciones-con-css3/>
- <http://www.desarrolloweb.com/articulos/transiciones-css3.html>
- <http://www.esandra.com/transformaciones-css3-transiciones-animaciones-css3>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=CSS3TransicionesAnimaciones>



FORMULARIOS

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al término de la unidad, el alumno diseña formularios para el envío de mensajes de los usuarios.

TEMARIO

5.1 Tema 9 : Formularios

- 5.1.1 : Elementos de un formulario : form, input, button, textarea
- 5.1.2 : Nuevos atributos para la etiqueta INPUT : number, range, url, email, date, etc.
- 5.1.3 : Atributos del formulario
- 5.1.3.1 : Vistosidad de las entradas: placeholder, autofocus, etc
- 5.1.3.2 : Validación de las entradas: required, pattern, etc

ACTIVIDADES PROPUESTAS

- Los alumnos diseñan formularios en una página web utilizando elementos input, button y textArea.

5.1 FORMULARIOS

HTML es un lenguaje de marcado, cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

Los años transcurridos desde la publicación de los estándares de HTML y XHTML ha provocado que no estén disponibles todos los elementos utilizados por los formularios más avanzados y modernos. No obstante, HTML/XHTML incluye los suficientes elementos de formulario para crear desde formularios sencillos que utilizan los buscadores hasta formularios complejos de las aplicaciones más avanzadas.

5.1.1 Elementos de un formulario: `form`, `input`, `button`, `textarea`

Los formularios se pueden crear utilizando solamente dos etiquetas: `<form>` y `<input>`.

La etiqueta `<form>` encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta `<input>` permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

Los elementos de formulario como botones y cuadros de texto también se denominan "campos de formulario" y "controles de formulario". La mayoría de controles se crean con la etiqueta `<input>`, por lo que su definición formal y su lista de atributos es muy extensa:

- **type** = "text | password | checkbox | radio | submit | reset | file | hidden | image | button" - Indica el tipo de control que se incluye en el formulario.
- **name** = "texto" - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario).
- **value** = "texto" - Valor inicial del control.
- **size** = "unidad de medida" - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en píxel).
- **maxlength** = "numero" - Máximo número de caracteres para los controles de texto y de password.
- **checked** = "checked" - Para los controles checkbox y radiobutton permite indicar qué opción aparece preseleccionada.
- **disabled** = "disabled" - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos.
- **readonly** = "readonly" - El contenido del control no se puede modificar.
- **src** = "url" - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario.
- **alt** = "texto" - Descripción del control.

Cuadro de texto

Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto. A continuación, se muestra el código HTML correspondiente al ejemplo anterior:

```
<input type="text" name="nombre" value="" />
```

El atributo **type** se diferencia a cada uno de los diez controles que se pueden crear con la etiqueta **<input>**. Para los cuadros de texto, su valor es **text**. El atributo **name** es el más importante en los campos del formulario. De hecho, si un campo no incluye el atributo **name**, sus datos no se envían al servidor. El valor que se indica en el atributo **name**, es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.

El atributo **value** se emplea para establecer el valor inicial del cuadro de texto. Si se crea un formulario para insertar datos, los cuadros de texto deberían estar vacíos. Si no se especifica un tamaño, el navegador muestra el cuadro de texto con un tamaño predeterminado. El atributo **size** permite establecer el tamaño, en caracteres, con el que se muestra el cuadro de texto. Su uso es imprescindible en muchos formularios, en los que algunos campos como la dirección deben mostrar más caracteres de lo normal (**<input size="100">**).

Además de controlar el tamaño con el que se muestra un cuadro de texto, también se puede limitar el tamaño del texto introducido. El atributo **maxlength** permite establecer el máximo número de caracteres que el usuario puede introducir en un cuadro de texto. Su uso es imprescindible para campos como el código postal, el número de la seguridad social y cualquier otro dato con formato predefinido y limitado.

Por último, el atributo **readonly** permite que el usuario pueda ver los contenidos del cuadro de texto, pero no pueda modificarlos y el atributo **disabled** deshabilita un cuadro de texto de forma que el usuario no pueda modificarlo y, además, el navegador no envía sus datos al servidor.

Cuadro de contraseña

La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escribe en un cuadro de contraseña no se ve en la pantalla. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos, por lo que es ideal para escribir contraseñas y otros datos sensibles.

```
<input type="password" name="contrasena" value="" />
```

Cambiando el valor del atributo **type** por **password** se transforma el cuadro de texto normal en un cuadro de contraseña. Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.

Checkbox

Los checkbox o "casillas de verificación" son controles de formulario que permiten al usuario seleccionar y deseleccionar opciones individualmente. Aunque en ocasiones se muestran varios checkbox juntos, cada uno de ellos es completamente independiente del resto. Por este motivo, se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas, pero no excluyentes.

Puestos de trabajo


```
<input name="ejecutivo" type="checkbox" value="direccion"/> Dirección  
<input name="tecnico" type="checkbox" value="tecnico"/> Técnico  
<input name="empleado" type="checkbox" value="empleado"/> Empleado
```

El valor del atributo **type** para estos controles de formulario es **checkbox**. Como se muestra en el ejemplo anterior, el texto que se encuentra al lado de cada checkbox no se puede establecer

mediante ningún atributo, por lo que es necesario añadirlo manualmente fuera del control del formulario.

El atributo **value**, junto con el valor del atributo **name**, es la información que llega al servidor cuando el usuario envía el formulario.

Si se quiere mostrar un checkbox seleccionado por defecto, se utiliza el atributo **checked**. Si el valor del atributo es checked, el checkbox se muestra seleccionado. En cualquier otro caso, el checkbox permanece sin seleccionar.

Radiobutton

Los controles de tipo radiobutton son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante: son mutuamente excluyentes. Los radiobutton se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecciona la otra opción que estaba seleccionada.

Sexo


```
<input type="radio" name="sx" value="hombre" checked="checked" /> Hombre  
<input type="radio" name="sx" value="mujer" /> Mujer
```

El valor del atributo type para estos controles de formulario es radio. El atributo **name** se emplea para indicar los radiobutton que están relacionados. Por lo tanto, cuando varios radiobutton tienen el mismo valor en su atributo name, el navegador sabe que están relacionados y puede desseleccionar una opción del grupo de radiobutton cuando se seleccione otra opción.

Botón de envío de formulario

La mayoría de los formularios dispone de un botón para enviar al servidor los datos introducidos por el usuario:

```
<input type="submit" name="buscar" value="Buscar" />
```

El valor del atributo **type** para este control de formulario es **submit**. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón. El valor del atributo value es el texto que muestra el botón. Si no se establece el atributo value, el navegador muestra el texto predefinido Enviar consulta.

Botón de reseteo del formulario

Aunque su uso era muy popular hace unos años, la mayoría de los formularios modernos ya no utilizan este tipo de botón. Se trata de un botón especial que borra todos los datos introducidos por el usuario.

```
<input type="reset" name="limpiar" value="Borrar datos del formulario" />
```

El valor del atributo **type** para este control de formulario es **reset**. Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original.

Campos ocultos

Los campos ocultos se emplean para añadir información oculta en el formulario.

```
<input type="hidden" name="url_previa" value="/articulo/primero.html" />
```

El valor del atributo **type** para este control de formulario es **hidden**. Los campos ocultos no se muestran por pantalla, de forma que el usuario desconoce que el formulario los incluye.

Botón de imagen

El aspecto de los botones de formulario se puede personalizar por completo, ya que incluso es posible utilizar una imagen como botón.

```
<input type="image" name="enviar" src="accept.png" />
```

El valor del atributo **type** para este control de formulario es **image**. El atributo **src** indica la URL de la imagen que debe mostrar el navegador en lugar del botón normal.

Su principal ventaja es que permite personalizar por completo la estética de los botones y mostrarlos con un aspecto homogéneo en todos los navegadores. El principal inconveniente es que ralentiza la carga del formulario y que, si se quiere modificar su aspecto, es necesario crear una nueva imagen.

Botón

Algunos formularios complejos necesitan botones más avanzados que los de enviar datos (**type="submit"**) y resetear el formulario (**type="reset"**).

```
<input type="button" name="guardar" value="Guardar Cambios" />
```

El valor del atributo **type** para este control de formulario es **button**. Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript.

Fieldset y Legend

La etiqueta **<fieldset>** agrupa campos del formulario y la etiqueta **<legend>** asigna un nombre a cada grupo.

```
<fieldset>
  <legend>Datos personales</legend>
  Nombre <br/>
  <input type="text" name="nombre" value="" />
  <br/>
  Apellidos <br/>
  <input type="text" name="apellidos" value="" />
  <br/>
  DNI <br/>
  <input type="text" name="dni" value="" size="10" maxlength="9" />
</fieldset>
```

La etiqueta **<fieldset>** agrupa todos los controles de formulario a los que encierra. El navegador muestra por defecto un borde resaltado para cada agrupación. La etiqueta **<legend>** se incluye dentro de cada etiqueta **<fieldset>** y establece el título que muestra el navegador para cada agrupación de elementos.

Área de Texto

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal.

```
<textarea id="idarea" name="txtdescripcion" cols="40" rows="5"></textarea>
```

Los atributos más utilizados en las etiquetas `<textarea>` son los que controlan su anchura y altura. La anchura del área de texto se controla mediante el atributo `cols`, que indica las columnas o número de caracteres que se podrán escribir como máximo en cada fila. La altura del área de texto se controla mediante `rows`, que indica directamente las filas de texto que serán visibles.

El principal inconveniente de los elementos `<textarea>` es que el lenguaje HTML no permite limitar el número máximo de caracteres que se pueden introducir. Mientras que, los elementos `<input type="text">` disponen del atributo `maxlength`, las áreas de texto no disponen de un atributo equivalente, por lo que sólo es posible limitar el número de caracteres mediante su programación con JavaScript.

Select

La etiqueta `<select>` define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta `<option>`. El atributo `value` de cada opción es obligatorio, ya que es el dato que se envía al servidor cuando el usuario envía el formulario. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo `selected` a la opción deseada.

```
<select id="so" name="so">
<option value="" selected="selected">- selecciona -</option>
<option value="windows">Windows</option>
<option value="mac">Mac</option>
<option value="linux">Linux</option>
<option value="otro">Otro</option>
</select>
```

OptGroup

La etiqueta `<optgroup>` permite agrupar opciones relacionadas dentro de una lista desplegable. Su definición formal se muestra a continuación:

El único atributo que suele utilizarse con la etiqueta `<optgroup>` es `label`, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación, de forma que el usuario pueda localizar más fácilmente la opción deseada.

5.1.2 Nuevos atributos para la etiqueta INPUT: number, range, url, email, date, etc.

El elemento `<input>` tiene nuevos valores para el atributo `type`:

- `<input type="email" />`
- `<input type="url" />`
- `<input type="date" />`
- `<input type="time" />`
- `<input type="datetime" />`
- `<input type="month" />`
- `<input type="week" />`

- <input type="number" />
- <input type="range" />
- <input type="tel" />
- <input type="search" />
- <input type="color" />

Campo email

En cuestión visual el campo de tipo **email** es básicamente una copia de text, pero con algunas peculiaridades propias, pues a la hora de usar **required** valida que el contenido del campo tenga el formato de una cuenta de email.

Campo Fecha, Mes, Semana, Hora

Con HTML5 la funcionalidad de fecha y tiempo viene integrada en el navegador en distintos tipos de campo: date, datetime, time, month y week.

El campo tipo **date** sería el más común, ofrece una forma directa de mostrar un calendario y seleccionar una fecha con su día, mes y año. En los distintos dispositivos móviles que se usa, muestra una ventana o menú nativo del sistema con el selector de fecha.

Los tipos **datetime** y **datetime-local** son parecidos y muestran un selector para elegir fecha y hora. El campo datetime se establece con fecha y horario en base a UTC y en cambio datetime-local no tiene definida nunca zona horaria.

El tipo **month** deja elegir directamente un año y su mes, mientras que **week** solo el número de la semana del año elegido.

Para elegir una hora, se utiliza **time** que da la selección numérica de hora y minuto, incluido la de segundos y fracciones de segundo en algunos navegadores.

Campo Número, Sliders: number & range

Para el manejo de campos numéricos se han introducido dos nuevos tipos **number** y **range**. Estos dos tipos comparten una serie de propiedades para fijar mínimo, máximo y número de pasos.

- **min="1"** define el mínimo valor aceptado por el campo.
- **max="15"** define el máximo valor.
- **step="2"** en base al mínimo define el rango de valores que sube hasta el máximo.

El campo **number** aparece como uno de texto con la singularidad que tiene flechas para sumar o restar a su valor. En cambio, **range** se muestra como un **slider** o campo deslizante que va desde el mínimo al máximo establecido por min y max y saltando de valor en valor según step.

Campo de búsqueda

El campo **search** funciona exactamente como text, pero con la peculiaridad del botón de borrado que aparece justo al escribir en él. Algunos navegadores incluyen el historial de búsqueda del propio site en este tipo de campo y los no compatibles simplemente muestran un campo text.

Campo para URLs

Un campo para escribir una dirección web, comúnmente denominada URL. Igual que el tipo anterior, el campo **url** es como text y en dispositivos móviles añade a su teclado virtual teclas de acceso rápido como las terminaciones de dominio .com, .net o .org e incluso el último historial de textos introducidos en este tipo de campo.

Campo Selector de color

El tipo **color** genera un campo que te mostrará el selector de paleta de colores según el sistema operativo que este usando, en el caso de Win o Mac o Linux, será el mismo que puedes usar en cualquier app de diseño como Photoshop o Gimp. En dispositivos móviles muestra una ventana por encima de la aplicación que normalmente deja escoger entre varios colores.

Campo Teléfono

El campo de teléfono **tel** está básicamente pensado para el uso en dispositivos móviles, debido a que el nivel de desktops no añade nada especial y hasta ahora la compatibilidad se basa en sus versiones mobile. Este te muestra el teclado virtual con el formato de teléfono para llenar el campo.

5.1.3 Atributos del formulario

La mayoría de formularios utilizan los atributos **action** y **method**. El atributo **action** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador. El atributo **method** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son GET y POST. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`.

El método POST permite el envío de mucha más información que el método GET. En general, el método GET admite como máximo el envío de unos 500 bytes de información. La otra gran limitación del método GET, es que no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante GET se observan en la barra de direcciones del navegador (se añaden al final de la URL de la página); mientras que, los datos enviados mediante POST no se pueden ver tan fácilmente. Por su parte, el método POST se debería utilizar cuando el formulario modifica la información original (insertar, modificar o borrar alguna información).

5.1.3.1 Vistosidad de las entradas: **placeholder**, **autofocus**

Con la remesa de nuevas entradas también ha llegado diferentes tipos de propiedades, las principales:

Autofocus

El navegador fija el foco al campo que lo tenga indicado automáticamente al cargar la página, ideal para sitios donde el campo de búsqueda es importante.

```
<input name="search" type="search" autofocus="" />
```

Placeholder

Mostrará un texto por defecto mientras el campo este vacío y al escribir desaparecerá, una mejora que ayuda a entender el propio campo.

```
<input name="search" type="search" placeholder="Buscar..." />
```

5.1.3.2. Validación de las entradas: **required**, **pattern**, etc

Required

El campo con este atributo será validado por el propio navegador, tanto si está vacío como si incluye un tipo como puede ser email o url aparecerá un pequeño globo informativo indicando el problema.

```
<input name="email" required="" type="email" />
```

Pattern

Esta propiedad permite usar una expresión regular para que el navegador valide el campo en base a un patrón establecido. Se debe especificar un title, ya que al validar si falla el navegador mostrará ese mismo.

```
<input title="Solo números" pattern="[0-9]*" type="text" />
```

Min y Max

Los atributos **min** y **max** del elemento <input> restringen los valores máximos y mínimos que pueden ser ingresados.

LABORATORIO 1

Se solicita diseñar un formulario en una página HTML, para registrar los datos de un contacto. Valide los datos antes de su envío.

Agregando la pagina HTML y el archivo CSS

Antes de iniciar el desarrollo de la página, primero agrega los archivos **CSS** y **HTML**, en el sitio **Web 7**.

```

EXPLORER      ...
OPEN EDITORS
SITIOWEB_07
  audios
  CSS
    # estilos-contacto.css
  imagenes
  js
  videos
  contactenos.html

contactenos.html X
contactenos.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Home</title>
7     <link rel="stylesheet" href="css/estilo-contacto.css">
8   </head>
9   <body>
10  </body>
11 </html>
12
13
14
15
16
17
18
19
20
21

```

Ln 1, Col 16 Spaces: 4 UTF-8 CRLF HTML ⚙️ ⚙️

Figura 248: Captura de pantalla
Fuente. - Elaboración propia

Enlazando la hoja de estilo

A continuación, enlaza la hoja de estilo CSS a la página HTML.

```

EXPLORER      ...
OPEN EDITORS
SITIOWEB_07
  audios
  CSS
    # estilos-contacto.css
  imagenes
  js
  videos
  contactenos.html

contactenos.html X
contactenos.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Home</title>
7     <link rel="stylesheet" href="css/estilo-contacto.css">
8   </head>
9   <body>
10  </body>
11 </html>
12
13
14
15
16
17
18
19
20
21

```

Ln 1, Col 16 Spaces: 4 UTF-8 CRLF HTML ⚙️ ⚙️

Figura 249: Captura de pantalla
Fuente. - Elaboración propia

Definiendo las reglas en el CSS: <header> y <section>

En el CSS define las reglas para el selector de bloque <header> y <section>. Luego, define los selectores de class: título asignado para el título e img-header, y el selector de id: img-header asignado para la imagen.

```

contactenos.html # estilo-contacto.css
css > # estilo-contacto.css > .titulo
1 main{
2     width: 100%;
3     height: auto;
4     float: left;
5 }
6
7 header{
8     width: 100%;
9     height: auto;
10    float: left;
11 }
12
13 .titulo{
14     text-align: center;
15     color: cornflowerblue;
16 }
17
18 #img-header{
19     width: 100%;
20     height: 250px;
21     float: left;
22 }
23
24 section{
25     width: 100%;
26     height: auto;
27     margin-top: 5px;
28     margin-bottom: 5px;
29     float: left;
30 }

```

Figura 250: Captura de pantalla

Fuente. - Elaboración propia

Agregando los elementos a la página HTML: <header> y <section>

En este proceso define las etiquetas del <header> de la página html y asigna el selector de id a las etiquetas y a la etiqueta <h1> del título. Además, agrega la etiqueta <section> y <h2>.

```

contactenos.html # estilo-contacto.css
contactenos.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 > <head> ...
4 </head>
5
6 <body>
7   <main>
8
9     <header>
10       
11       <h1 class="titulo">La Tecnología y la Informática</h1>
12     </header>
13
14     <section>
15       <h2 class="titulo">Contáctenos</h2>
16     </section>
17
18   </main>
19
20 </body>
21
22 </html>

```

Figura 251: Captura de pantalla

Fuente. - Elaboración propia

Presiona la tecla **F5** para visualizar la página html.



Figura 252: Captura de pantalla
Fuente. - Elaboración propia

Codificación en HTML

Para diseñar un formulario validado, codifica las etiquetas en la página HTML, tal como se muestra. Para que los inputs se encuentren validado, utiliza los atributos placeholder y required.

```
contactenos.html # estilo-contacto.css
contactenos.html > html > body
18 <section>
19   <h2 class="titulo">Contáctenos</h2>
20   <form method="POST">
21     <label>Nombre:</label><br>
22     <input name="nombre" placeholder="Tu Nombre" required autofocus /></p>
23
24     <label>Apellido:</label><br>
25     <input name="apellido" placeholder="Tu Apellido" required /></p>
26
27     <label>EMail:</label><br>
28     <input type="email" name="email" placeholder="tu@email.com" required /></p>
29
30     <label>Grado de Instrucción:</label><br>
31     <select name="grado">
32       <option value="secundaria" selected>Secundaria</option>
33       <option value="superior">Superior</option>
34       <option value="tecnico">Técnico-Superior</option>
35       <option value="posgrado">PosGrado</option>
36     </select><p>
37
38     <label>Genero</label><br>
39     <input type="radio" name="genero" checked value="M">Masculino
40     <input type="radio" name="genero" value="F">Femenino
41     </p>
42
43     <label>Comentario</label><br>
44     <textarea name="comentario" placeholder="Escriba su comentario"
45       cols="25" rows="10"></textarea>
46     </p>
47     <input type="submit" value="Enviar">
48   </form>
49 </section>
```

Figura 253: Captura de pantalla
Fuente. - Elaboración propia

Ejecutando la página

Al ejecutar la página a través del navegador, se visualiza el formulario y sus controles, tal como se muestra.

En este proceso se puede visualizar el valor del placeholder y cuando no se ingresa datos en un control con el atributo required, se muestra un mensaje.

La captura de pantalla muestra una ventana de navegador con un formulario titulado "Contáctenos". El formulario tiene los siguientes campos:

- Campo "Nombre": Un cuadro de texto que contiene "Tu Nombre".
- Campo "EMail": Un cuadro de texto que contiene "tu@email.com".
- Campo "Grado de Instrucción": Una lista desplegable que muestra "Secundaria".
- Campo "Genero": Una sección con dos radio buttons: "Masculino" (seleccionado) y "Femenino".
- Campo "Comentario": Un cuadro grande para escribir comentarios que dice "Escriba su comentario".
- Botón "Enviar": Un botón rectangular en la parte inferior izquierda.

Un cuadro emergente amarillo rodea el campo "Nombre" con el mensaje "Completa este campo".

Figura 254: Captura de pantalla
Fuente. - Elaboración propia

LABORATORIO 2

Se solicita publicar un formulario para realizar un envío de datos, por MelodySoft, a través de una página HTML.

Pasos para publicar un formulario

1. Ingresa a la página www.melodysoft.com
2. Escribe su login y password.



Figura 255: Captura de pantalla
Fuente. - Elaboración propia

Después de ingresar el login y clave del usuario creado, verás la siguiente pantalla.

3. Haz clic en la opción **Formularios de envío**.



Figura 256: Captura de pantalla
Fuente. - Elaboración propia

4. Haz clic en la opción **Crear nuevo formulario**.



Figura 257: Captura de pantalla
Fuente. - Elaboración propia

5. Lee las condiciones del servicio y finalmente, activa la opción **He leído y acepto las condiciones de servicio** y haz clic en el botón **Aceptar y continuar**.

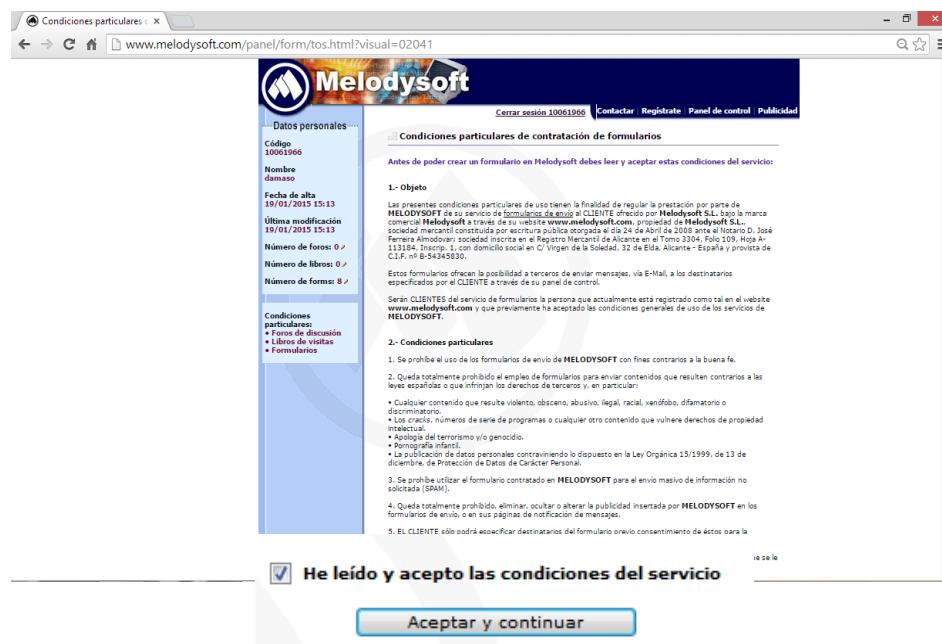


Figura 258: Captura de pantalla
Fuente. - Elaboración propia

6. Completa los campos que se solicitan, al finalizar presiona el botón **Crear**.

Código identificativo formulario

 El identificativo servirá como código de identificación del formulario. No debe contener espacios ni signos extraños.

Nombre del formulario

Asunto del formulario

 Es el asunto que figurará en el envío del formulario.

Dirección E-Mail destino

 Dirección de E-Mail principal a la que se enviará el formulario.
 Puedes indicarla de estas dos maneras: correo@dominio.com o bien:
["mi nombre" <correo@dominio.com>](mailto:'mi nombre' <correo@dominio.com>)

Formato del E-Mail

Temática

 Temática de la página del formulario.

Campo donde figura el nombre del usuario

 Indica el nombre del campo de donde se tomará el nombre del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.
 Cuando se envía el formulario, o una posible respuesta automática al usuario, en esos E-Mails figura su nombre, este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el nombre del usuario, si por ejemplo ponemos *nombre* en este campo, entonces la caja de texto donde el usuario indicaría su nombre se llamaría *nombre*:
`<input type="text" name="nombre">`

Campo donde figura el E-Mail del usuario

 Indica el nombre del campo de donde se obtiene el E-Mail del usuario para enviarle la respuesta personalizada y/o para que figure en el campo "From" del formulario.
 Cuando se envía el formulario, o una posible respuesta automática, se indica el E-Mail origen (formulario), o se envía al E-Mail del usuario (respuesta), este dato se obtiene de los propios datos del formulario, por lo que hay que indicar aquí qué campo es el que contiene el E-Mail del usuario, si por ejemplo ponemos *email* en este campo, entonces la caja de texto donde el usuario indicaría su E-Mail se llamaría *email*:
`<input type="text" name="email">`

Haz clic para crear el formulario

[Volver](#)

Creado el formulario, la plataforma indica que el formulario ha sido creado con éxito.

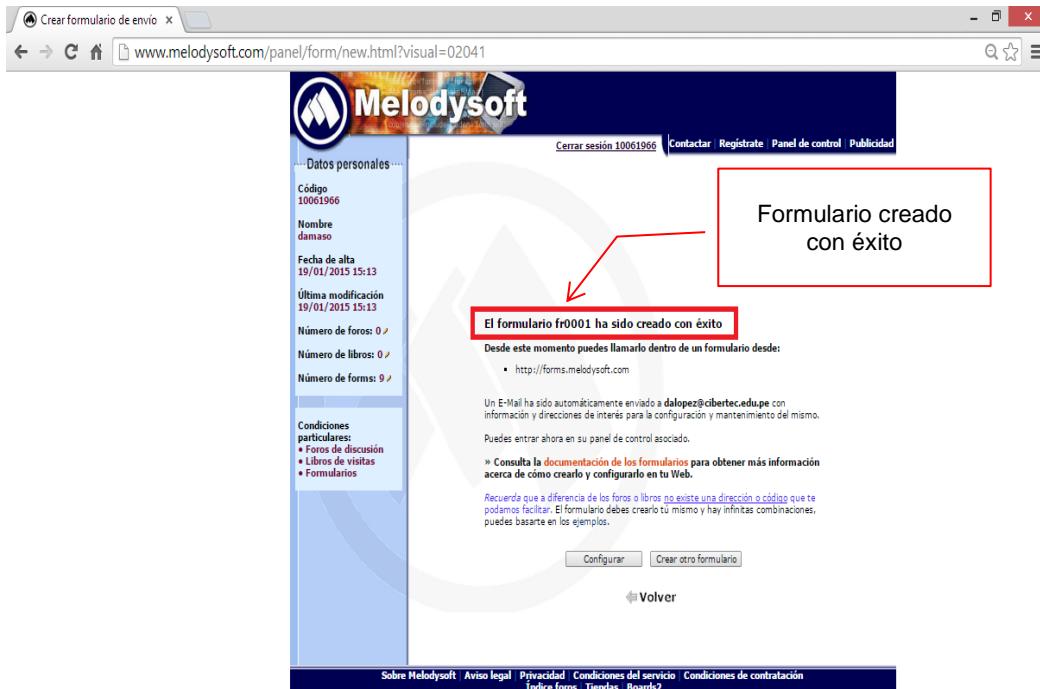


Figura 259: Captura de pantalla
Fuente. - Elaboración propia

En el formulario de la página HTML, registra el formulario de envío, tal como se muestra.

La Etiqueta form:

```
<form name="form1" method="post" action="http://forms.melodysoft.com" >
<input name="id" type="hidden" value="nombre_formulario">
```

Nota

El **nombre_formulario** es el código identificativo del formulario que se creó en la página de MelodySoft

Resumen

1. HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web; sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.
2. Los formularios se pueden crear utilizando solamente dos etiquetas: **<form>** y **<input>**. La etiqueta **<form>** encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto).
3. El elemento **<input>** tiene nuevos valores para el atributo **type**:
 - `<input type="email" />`
 - `<input type="url" />`
 - `<input type="date" />`
 - `<input type="time" />`
 - `<input type="datetime" />`
 - `<input type="month" />`
 - `<input type="week" />`
 - `<input type="number" />`
 - `<input type="range" />`
 - `<input type="tel" />`
 - `<input type="search" />`
 - `<input type="color" />`
4. La mayoría de formularios utilizan los atributos **action** y **method**. El atributo **action** indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador. El atributo **method** establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son GET y POST. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`.

5. Con la remesa de nuevas entradas también ha llegado diferentes tipos de propiedades: Autofocus, el navegador fija el foco al campo que lo tenga indicado automáticamente al cargar la página, ideal para sitios donde el campo de búsqueda es importante. Placeholder, muestra un texto por defecto mientras el campo este vacío y al escribir desaparecerá.
6. Para validar las entradas en un formulario, se tiene required, pattern.

Recursos

Puede revisar los siguientes enlaces para ampliar los conceptos vistos en esta unidad:

- <http://octuweb.com/formularios-html5>
- https://developer.mozilla.org/es/docs/HTML/HTML5/Forms_in_HTML5
- http://librosweb.es/libro/xhtml/capitulo_8.html
- <http://www.genbetadev.com/desarrollo-web/introduccion-a-los-formularios-de-html5-ii>

Bibliografía

- Acera Garcia, M.A. (2012). *CSS3*. Madrid: Anaya Multimedia
Centro de Información: Código 006.7 ACER
- Alvarez Garcia, A. (2012). *Manual imprescindible de HTML*. Madrid : Anaya Multimedia.
Centro de Información: Código 006.74HTML ALVA/M
- David, M.(2011). *HTML5*. Madrid: Anaya Multimedia
Centro de información: Código 006.74 HTML DAVI
- Gauchat, J.D (2019). *El gran libro de HTML5, CSS3 y JavaScript. 3ra Edición*. Barcelona:Marcombo. Recuperado de <https://elevaciondigital.pe/wp-content/uploads/2019/06/El-gran-libro-de-HTML5-CSS3-y-JavaScript.pdf>
- Goldstein, A. (2011). *Manual imprescindible de HTML 5 y CSS3*. Madrid : Anaya Multimedia.
Centro de Información :Código 006.74 HTML GOLD
- Luján, J. (2016). *HTML5, CSS y JAVASCRIPT: crea tu web y apps con el estándar de desarrollo*. Bogotá: Alfaomega.
Centro de información: 006.74HTML LUJA
- MacDonald, M. (2016). *Creación y diseño web*. Madrid: Anaya Multimedia
Centro de Información: Código 006.7MACD 2016
- Martin, D.C. (s.f.). *Diseño Web CSS*. Recuperado de <https://www.diegocmartin.com/category/web/css/>
- Meloni, J. C. (2012). *HTML5, CSS3 y JavaScript*. Madrid: Anaya Multimedia
Centro de Información: Código 006.74 HTML MELO
- Meloni, J.C. (2015). *Html5, CSS3 y JavaScript*. Madrid: Anaya Multimedia
Centro de Información: Código 006.74HTML MELO 2015
- Rubiales Gómez, M. (2018). *Curso de desarrollo web: HTML, CSS y JavaScript*. Madrid:
Anaya Multimedia
Centro de Información: Código 006.74HTML RUBI/C
- Uniwebsidad. (s.f.). *Introducción a CSS*. Recuperado de <https://uniwebsidad.com/libros/css>
- Uniwebsidad. (s.f.). Recuperado de <http://librosweb.es/libros/>