INSTITUTO SUPERIOR TECNOLÓGICO QUITO

Formamos tu PROPÓSITO DE VIDA

Guía general de DESARROLLO WEB I

DESDE 1984 - INSTITUTO SUPERIOR TECNOLÓGICO QUITO

SABER HACER Bien

INSTITUTO SUPERIOR TECNOLÓGICO QUITO
Excelencia en Educación Superior

GUÍA GENERAL DE DESARROLLO WEB I

AUTOR: ELVIS PACHACAMA

PRIMERA EDICIÓN

AÑO: 2024

TRABAJO EN EDICIÓN:

DOXA EDITION

DIRECCIÓN EDITORIAL: DIEGO JAVIER BASTIDAS LOGROÑO

EDICIÓN EXTERNA: DAVID FABIAN CEVALLOS SALAS

ISBN:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

SOBRE EL AUTOR

Elvis Pachacama Cabezas es un educador y profesional con un título de tercer nivel en Ingeniería en Tecnologías de la Información, obtenido en la Universidad Estatal de Sumy en Ucrania (SUMDU). Se destaca por su participación en la vida estudiantil, ya que fue representante de los estudiantes ecuatorianos en el Consejo Estudiantil de la universidad, lo que sugiere un compromiso activo con los asuntos estudiantiles.

Además, Elvis se graduó con honores y recibió reconocimiento por su destacado desempeño académico en su título universitario. Esta distinción refleja su dedicación y excelencia en sus estudios. Actualmente, Elvis Pachacama Cabezas es docente en el Instituto Superior Tecnológico Quito, donde imparte cátedra en la Carrera de Desarrollo de Software. Su posición como educador indica una dedicación al campo de la enseñanza y sugiere que posee conocimientos prácticos en el desarrollo de software, que comparte con los estudiantes.

Adicionalmente, es relevante mencionar que Elvis está próximo a cursar la maestría en inteligencia artificial en la Universidad Internacional de Valencia. Este detalle indica su interés en profundizar sus conocimientos en un campo avanzado y en constante evolución, como es la inteligencia artificial.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

CONTENIDO

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

ÍNDICE DE FIGURAS

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

ÍNDICE DE TABLAS

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

GUÍA GENERAL DE DESARROLLO WEB I

1. DESCRIPCIÓN DE LA ASIGNATURA

En la era digital actual, la asignatura de Desarrollo Web I, emerge como una pieza angular en la formación de los individuos interesados en el fascinante mundo de la programación web. Este curso de adentra en tres elementos cruciales que sustentan la creación de sitios web modernos. HTML, CSS y JavaScript.

HTML (Hyper Text Markup Language): Se rige como el cimiento sobre el cual se construyen todas las páginas web, permitiendo la estructuración y organización del contenido de forma lógica y coherentes es decir HTML es el esqueleto de nuestra página web.

CSS (Cascade Style Sheets): Se adentra en el mundo del diseño y estética web, proporcionando las herramientas necesarias para dar vida a las páginas web visualmente atractivas a través de la aplicación de estilos, colores y formatos.

JavaScript: JavaScript, es un lenguaje de programación versátil, alade un nivel adicional de interactividad y dinamismo a las páginas web, permitiendo a los desarrolladores crear experiencias más ricas y receptivas para los usuarios.

A lo largo de esta asignatura, los estudiantes no solo obtendrán un profundo conocimiento de estas tres tecnologías esenciales, sino que también desarrollan capacidades de integrarlas de manera efectiva para construir sitios web funcionales y atractivos.

Además, se hará hincapié en un enfoque práctico y aplicado, lo que significa que los estudiantes no solo adquirirán conocimientos teóricos, sino también tendrán la oportunidad de trabajar en proyectos reales, desde la creación de sitios web básicos hasta la implementación de características interactivas avanzadas. Así, podrán aplicar lo que aprendan en un enrono práctico y estarán mejor preparados para enfrentar los desafíos del mundo real en el desarrollo web. A medida que avancen en su viaje de aprendizaje, se fomentará la resolución de problemas y trabajo en equipo, habilidades fundamentales en la industria del desarrollo web.

2. BIBLIOGRAFÍA

La bibliografía básica debe contener los textos que son necesarios y obligatorios para el desarrollo de la asignatura, y que cubren los contenidos y competencias principales. La bibliografía complementaria debe contener los textos que son de apoyo y consulta recomendada, pero no obligatoria, y que aportan diversidad, profundidad y actualización al

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

tema de estudio. Se recomienda al menos 2 títulos para la bibliografía básica y al menos 4 títulos para la bibliografía complementaria.

2.1. Básica

- Recio García, J. A. (2016). *Html5, CSS3 y JQuery: cuso practico*: RA-MA Editorial.

Este texto ofrece una introducción accesible y práctica a las tecnologías web modernas. Está diseñado para ayudar a estudiantes y desarrolladores a aprender a crear sitios web interactivos y visualmente atractivos utilizando HTML5 para estructurar el contenido, CSS3 para definir el diseño y la apariencia, y JQuery para agregar funcionalidades avanzadas mediante JavaScript. Este texto abarca todos los conceptos básicos, para el desarrollo de páginas.

El enfoque del libro es aprender haciendo, por lo que cada capítulo incluye ejercicios prácticos y proyectos reales que permiten a los estudiantes aplicar lo aprendido inmediatamente. Además, se exploran buenas prácticas de desarrollo, como la compatibilidad entre navegadores y la optimización del rendimiento, ayudando a los lectores a desarrollar habilidades sólidas en la creación de sitios web modernos y profesionales.

- Gómez López, J. & Alcayde García, A. (2015). Construcción de páginas web.

El libro "Construcción de Páginas Web" de Gómez López y Alcayde García es una guía completa para aprender a crear y desarrollar sitios web efectivos. La obra está orientada a proporcionar a los lectores las herramientas y conocimientos necesarios para diseñar páginas web modernas y funcionales.

Una parte fundamental del texto está dedicada al aprendizaje de HTML y CSS, los lenguajes esenciales para estructurar y diseñar páginas web. Los autores explican cómo utilizar etiquetas HTML para estructurar el contenido y cómo aplicar estilos con CSS para mejorar la apariencia de las páginas.

2.2. Complementaria

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS

Valores y habilidades blandas: Amor: Comunicación asertiva y escucha activa

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Valores y habilidades blandas: Compromiso social, adaptabilidad.

Valores y habilidades blandas: cultura: creatividad.

Valores y habilidades blandas: gratitud: resiliencia.

Valores y habilidades blandas: justicia: resolución de problemas.

Valores y habilidades blandas: Lealtad y liderazgo.

Valores y habilidades blandas: Optimismo, planificación y gestión del tiempo

Valores y habilidades blandas: orgullo nacional: pensamiento crítico

Desarrolla aplicativos informáticos mediante el uso de plataformas actuales de programación.

4. OBJETIVO GENERAL

Proporcionar a los estudiantes una comprensión sólida y completa en los tres pilares fundamentales del desarrollo web: HTML5, CSS y JavaScript con una metodología de enseñanza centrada en la práctica, los estudiantes aprenderán como utilizar HTML para estructurar el contenido web, aplicar CSS para diseñar interfaces atractivas y emplear JavaScript para crear experiencias web interactivas. Al finalizar el curso, los estudiantes estarán equipados con habilidades necesarias para construir sitios web funcionales y estéticamente agradables, lo que les permitirá contribuir de manera efectiva al mundo digital en constante evolución y sentar las bases para futuras exploraciones y y desarrollos en el campo del diseño y programación web.

5. FORMACIÓN CIUDADANA, VALORES Y HABILIDADES BLANDAS

6. NORMAS DE CLASE

En relación con las normas de clases, es importante destaca que la evaluación de los componentes de gestión académica se compone de tres notas sumativas, cada un con una puntuación máxima de 6.60/6.60, así como como un proyecto práctico, como evaluación formativa que se valora con 3.40/3.40, que da un total de 10/10 para la calificación del módulo. Los parciales se califican en una escala de hasta 6.60 puntos, representando cada uno el 2.22 de la calificación total de 6.6 puntos. Para presentarse al proyecto final, el estudiante debe haber obtenido al menos 4.50 puntos sumando las tres primeras notas. En caso de no alcanzar este mínimo en el proyecto, se otorga una oportunidad de recuperación dentro de las 48 horas laborables siguientes, según el calendario académico oficial. La nota

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

mínima acumulada requerida para aprobar la asignatura es de 7/10, y es esencial mantener al menos un 70% de asistencia a las clases. Los docentes deben informar a los estudiantes sobre sus notas individuales antes de registrarlas en el sistema, y se espera que los alumnos confirmen su aceptación y conformidad con estas calificaciones. Además, los docentes deben entregar un reporte de notas y asistencia a través del SGA y notificado a la coordinación de carrera y registrar las calificaciones en el sistema en un plazo máximo de 5 días posteriores a la recepción del proyecto final. Los estudiantes deberán estar a la hora indicada en el horario si en el caso que el estudiante llegue 5 minutos tarde podrán ingresas a clase, pero en su registro de asistencia tendrá falta.

7. SISTEMA DE EVALUACIÓN

Tabla 1

Sistema de evaluación

INSTRUMENTO	PORCENTAJE	PUNTAJE
PARCIAL 1	22.0	2.2
PARCIAL 2	22.0	2.2
PARCIAL 3	22.0	2.2
PROYECTO	34.0	3.4
TOTAL	100.00	10.0

Nota: Tabla tomada el PEA sobre el instrumento de evaluación.

8. UNIDADES

Unidad 1: Terminología básica de la web y HTML

Unidad 2: Ambiente de desarrollo y hojas de estilo

Unidad 3: Programación interactiva con JavaScript

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

UNIDAD 1

TERMINOLOGÍA BÁSICA DE LA WEB Y HTML

Temas y Subtemas

- **Tema 1 Introducción a la programación web, revisión de conceptos página web estáticas y dinámicas**
- **Tema 2 DOM dentro del desarrollo web**
- **Tema 3 Sintaxis HTML**
- **Tema 4:**

Introducción a la programación web.

El desarrollo de una aplicación web implica el uso de diferentes tecnologías y herramientas que nos permite plasmar las ideas mediante líneas de código y poder verlas en la internet, las páginas web nos permite intercambiar y almacenar información en los servidores y luego poder visualizarlos en el navegador del usuario. La unión de estas tecnologías nos permitirá al final del desarrollo visualizar una página web llamativa, responsiva e intuitiva (Fernández Casado, 2023). Muchas de las tecnologías al principio serán un poco complicado de entender cómo (HTML, CSS, AJAX, JS, PHP …), pero al final todas estas tecnologías se complementarán en una sola y se podrán comprender mejor.

Antes de entrar con detalle de cómo desarrollar sitios web, vamos a dar un pequeño paseo por la historia que no permitirá entender cómo funciona la Web.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Historia del Desarrollo Web.

(Fontecha, 2022)Antes de entrar al mundo del desarrollo vamos a viajar al pasado y entender, cómo empezó este mundo fascinante del desarrollo de sitios web.

Los primeros pasos se remontan al año de 1989 donde el científico Tim-Berners-Lee, de CERN (Centro Europeo para la Investigación Nuclear), propone el proyecto WWW “, World Wide Web”, la cual facilitaría la compartición de archivos entre investigadores.

1990. Para inicio de los 90’s Tim-Berners-Lee el primer navegador y editor de código llamando **WorldWidwWeb**, que posteriormente se llamaría **Nexus.**

Figura 1.

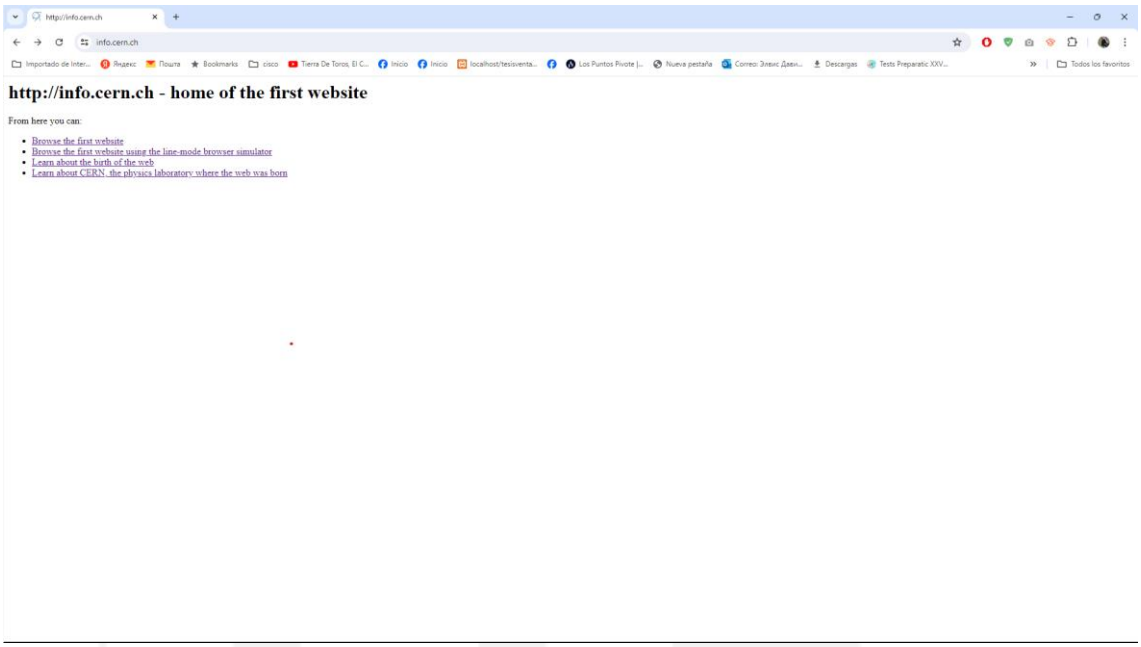
Primer Navegador

Nota: Netscape Navigator 4.04. Imagen de Andrew Turnbull

1991. A inicios de la década de los 90’s se lanza el primer sitio Web que fue desarrollado por Tim Berners-Lee, este sitio web proporcionaba información sobre el proyecto **World Wide Web,** (https://info.cern.ch).

Figura *2*

Primera página Web



Nota: Imagen de referencia tomada de https://info.cern.ch

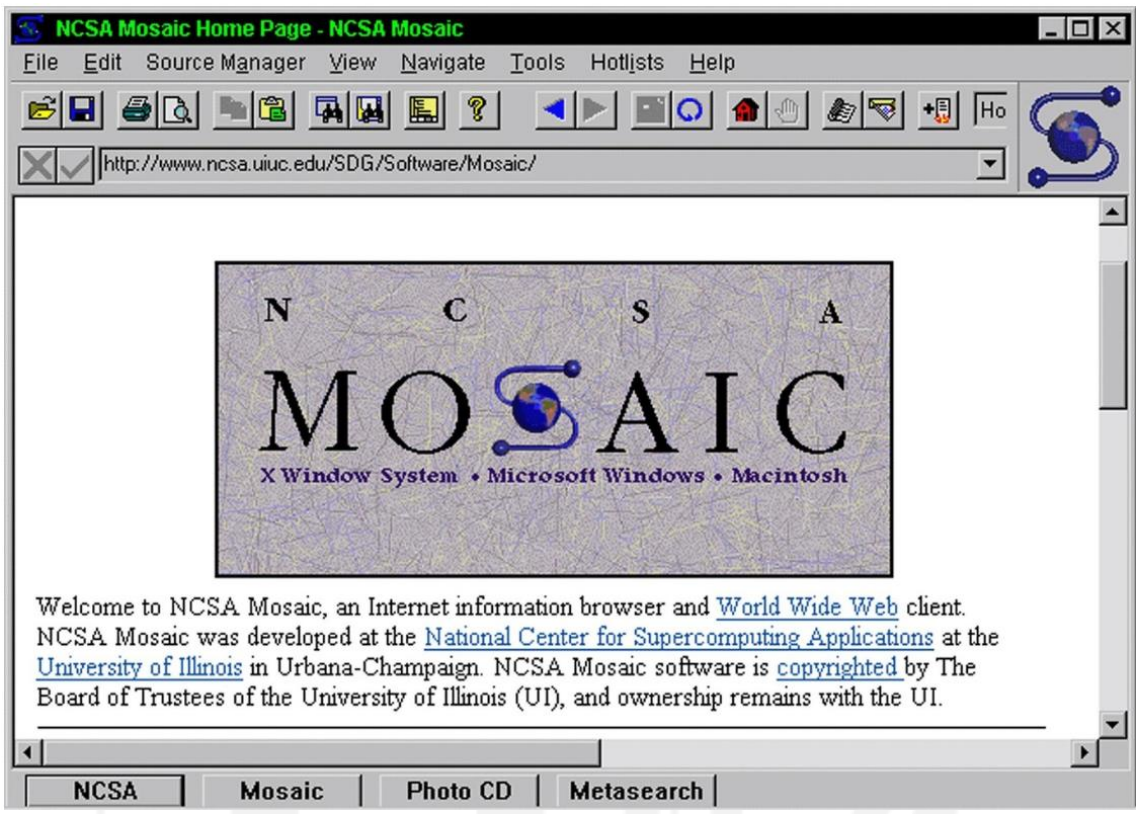
Como podemos ver la primera página web tiene un diseño plano eso quiere decir que solo esta desarrollado con HTML no tiene nada de estilos solo la información la cual intercambiaban entre los científicos de la CERN, hoy en día la página web está ejecutándose con toda la información que se subió al servidor (Gómez López, 2015).

1993. Se lanza Mosaic no es el primer navegador, pero si el primero que se uso masivamente desarrollado por Marc Andreessen y Eric Bina del NCSA, este navegador establece el estándar para los futuros navegadores.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 3.

Navegador Mosaic



Nota: Navegador Mosaic imagen tomada de https://www.xataka.com/historia-tecnologica/cuando-mosaic-dominaba-el-mundo-de-los-navegadores

1995. A mediados de la década de los 90 se lanza HTML 2.0 y el primer estándar oficial de HTML y HTML 2.0, también se lanza JavaScript que es un lenguaje de programación de lado del cliente que sirve para crear contenido dinámico en las páginas web, al mismo tiempo se lanzas PHP, que es un lenguaje de programación que se ejecuta del lado del servido el cual nos permite el manejo de datos en las aplicaciones web (Pére Rodríguez, 2018) (Celaya Luna, 2014).

1996. Para este año se introduce **CSS (Cascading Style Sheets),** que (Pére Rodríguez, 2018)son hojas de estilo que nos permite darle vida a nuestras páginas web, como colores y nos permite mejorar la presentación de nuestros sitios web.

El mismo año se lanza **Flash,** que es lanzado por la empresa Macromedia, que es una plataforma de multimedia que permite la creación de animaciones, juegos y aplicaciones interactivas.

1997. Para este año se lanza HTM 3.2, la cual introduce varias etiquetas y capacidades incluyendo tablas y scripts.

1998. Con el gran crecimiento de la Web en este año aparece HTML 4.0 el cual permite separar la estructura del contenido y la presentación, de la misa forma mejorando la accesibilidad e

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

internacionalización. EL mismo año se fundo **Google,** que se convierte en el motor de búsqueda más importante y utilizado.

1999. Se lanza XML (Extensible Markup Language), permite una mejor gestión y transmisión de datos estructura.

2000-2006. Empieza la era de la Web 2.0 donde se lanzan muchas tecnologías y plataformas como:

- XHTML
- WordPress, que es un sistema de gestión de contenido un (CMS)
- Se implementa AJAX (Asynchronus JavaScript and XML), que nos permite el desarrollo de aplicaciones web más dinámicas e interactivas.
- Se funda **Facebook** que se convierte y una de las mayores redes sociales.
- Se funda **Youtube,** la cual se transforma a la distribución de videos en la web.
- Se funda **Twitter,** popularizando la comunicación en tipo real a través de microblogging.

2007-2013. Empieza la era de la web móvil y la nube implementado nuevas y mejoradas tecnologías.

- Apple lanza el iPhone, revolucionando el acceso móvil a la web.
- Google lanza el navegador Chrome, que se enfoca en la velocidad y el rendimiento de datos.
- HTML 5 se presenta como un borrador, introduciendo nuevas APIs y elementos semánticos.
- Una de las tecnologías que a revolucionado el desarrollo web es lanzado como Node.js , código Java Script que se ejecuta del lado del servidor.
- Se lanza Angular Js por Google, popularizando los frameworks JavaScript para aplicaciones de una sola página.
- React es lanzado por Facebook, esto revoluciona la construcción de interfaces de usuario con un enfoque basado en componentes.

2014-Presente. La modernización y la Inteligencia Artificial.

- Se lanza Bootstrap 4, que es un framework CSS popular para el diseño responsivo.
- Se lanza CSS Grid Layout, mejorando significativamente las capacidades de diseño CSS.
- La privacidad y seguridad en la web se vuelve temas centrales, como iniciativas como HTPPS everywhere.
- La inteligencia artificial y el aprendizaje automático comienzan a integrarse más en aplicaciones web, mejorando la personalización y la experiencia del usuario.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Después de haber terminado con una breve historia de la evolución del Desarrollo Web explicaremos ciertos conceptos y tecnologías que son la base para el desarrollo de aplicaciones Web.

Arquitectura Cliente-Servidor

Te has puesto a pensar ¿Cómo se realiza el proceso de comunicación cuando solicitas la información de una página web? Aquí te lo explico.

Para realizar la comunicación y solicitar una página web intervienen dos actores un cliente que viene hacer el navegador y el servidor donde se guarda dicha información como lo explicamos en la figura 4.

1. El cliente escribe en el buscador url la dirección donde este alojado la aplicación en el servidor.

Figura 4.

Petición URL

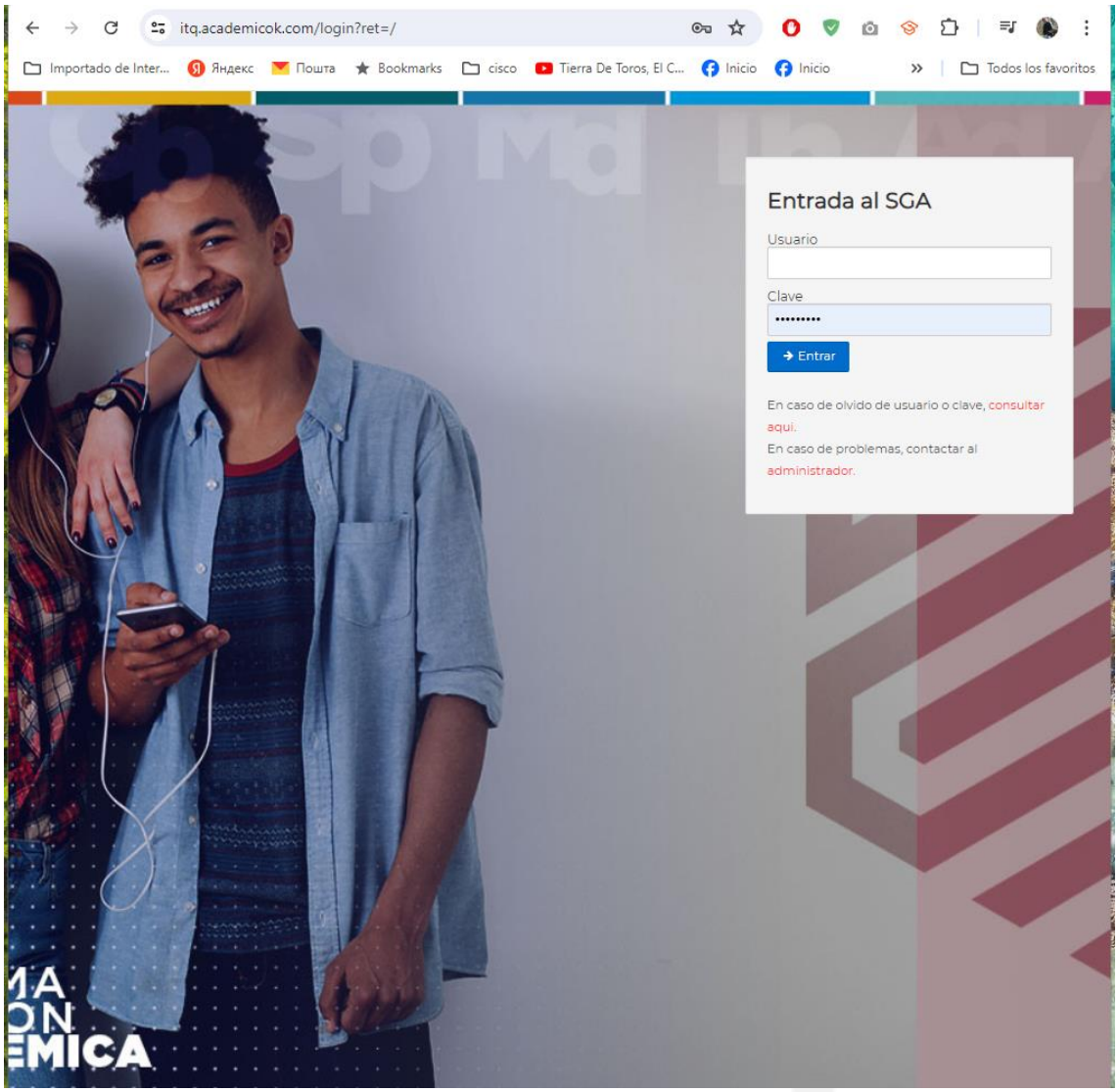
Nota: La imagen representa a la dirección URL donde se va a buscar la dirección solicitada

2. Al realizar la petición al servidor ya sea por el método **GET** o el método **POST,** el servidor busca la información deseada una vez que encuentra el servidor devuelve la la información deseada al usuario.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 5.

Página Web ITQ

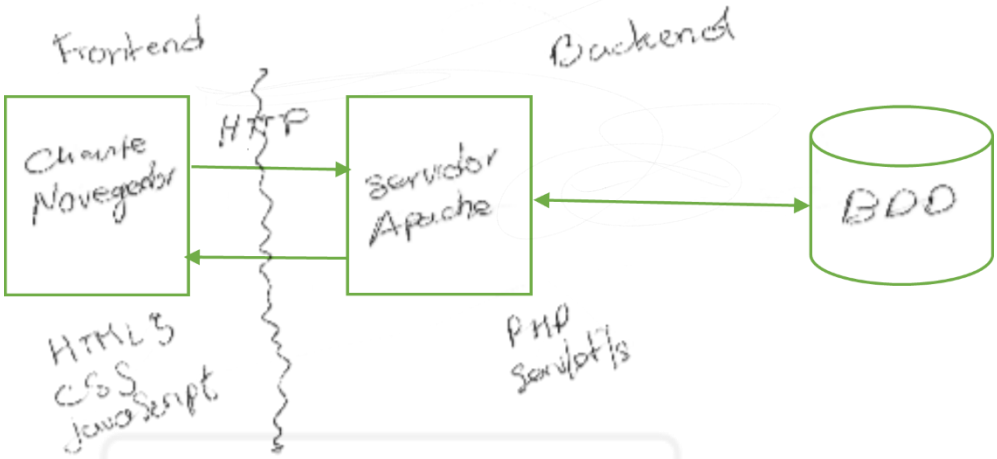


Nota: *Información devuelta realizando la petición del punto 1 el servidor busca y devuelve lo solicitado y puede ver el usuario la información.*

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 6

Petición de una página Web



Nota: La imagen hace referencia a como se realiza la petición de una página web a un servidor. Cuando desarrollamos aplicaciones web existen distintas tecnologías para cada una de las partes en el parte del cliente que es el **front-end.** Se utiliza numerosas tecnologías como las que ya hemos hablado en cambio para el lado del servidor utilizamos las tecnologías para el almacenamiento y procesamiento de datos. Para esta guía nos centraremos en las tecnologías de **front-end**.

TIPOS DE PETICIONES HTTP.

El Protocolo de Transferencia de hipertexto (HTTP), es la base de la comunicación en la web. Cada vez que accedes a una página web, envías una solicitud HTTP al servidor que aloja esa página. Los servidores responden a estas solicitudes con la información que has solicitado, como una página web, un archivo o datos. Las solicitudes HTTP se realizan utilizando distintos métodos, cada uno diseñado para una tarea especifica.

Métodos de solicitudes HTTP

Get: El método **GET** solicita la representación de un recurso especifico. Las solicitudes que usan GET solo deben recuperar datos, estas solicitudes viajen a de una forma visible eso quiere decir que el usuario puede ver que es lo que se esta solicitando por medio de la URL.

Figura 7.

Petición GET

Nota. La imagen representa una petición median Get al servidor donde, se observa los recursos solicitados al servidor.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

POST: El método **POST** se utiliza para enviar datos al servidor, como cuando se envia un formulario. Puede crear nuevos recursos o actualizar los existentes, esta tipa de petición envía la información por detrás de la aplicación eso quiere decir que el usuario no puede ver la información que se está enviando.

Figura 8.

Petición POST

Nota: En la imagen se puede enviar información por el método POST.

Existen otros métodos que se utilizan para realizar peticiones, pero en esta guía solo nos centraremos en estos dos y los otros no los abordaremos.

Tecnologías del Front-End

Para el desarrollo del front-end se utilizarán algunas tecnologías para el lado del cliente, es decir en el navegador el cual va a visualizar e interpretar los documentos que se obtiene del servidor. El navegador es en realidad una avanzada amalgama de tecnologías que permiten visualizar documentos HTML y ejecutar código que los modifica y permite interactuar con el usuario.

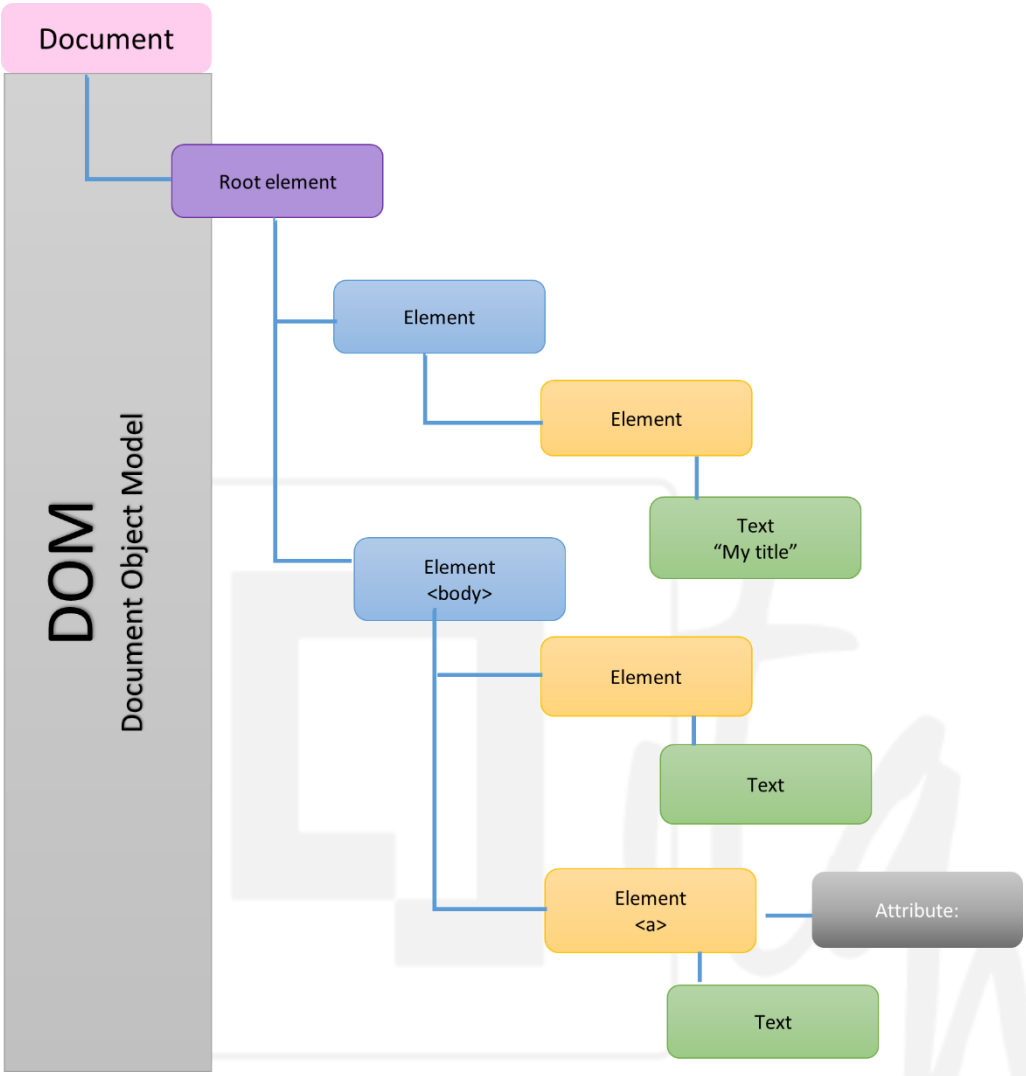
DOM en HTML

El Modelo del Objetos del Documento (DOM) es una interfaz de programación para documentos HTML y XML. Representa la estructura de un documento como un árbol de nodos, donde cada nodo corresponde a una parte del documento, como un elemento, un atributo o un texto. El DOM permite a los lenguajes de programación como JavaScript, acceder y manipular la estructura, el contenido y el estilo de los documentos web.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 9.

DOM



Nota: La imagen representa el DOM en HTML.

Introducción a HTML

El lenguaje HTML (hypertext markup language) es el lenguaje con él se define una página Web, HTML no es un lenguaje de programación, HTML es un lenguaje de marcado o etiquetado.

Nos permite describir el contenido de una página, incluyendo texto y otros elementos como imágenes, videos y otros elementos.

HTML, es una tecnología fundamental de la World Wide Web y ha ido evolucionando hasta nuestros días. En esta guía nos vamos a centrar en la última versión del lenguaje: HTML5, en el cual comenzaremos a describir cada uno de los conceptos fundamentales de este lenguaje.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Herramientas de desarrollo

Para escribir código HTML solo necesitas un editor de texto. Pero para no tener ningún inconveniente al escribir texto HTML se necesita un editor de código especifico, el cual nos va a facilitar la escritura con autocompletados, resaltando la sintaxis.

En esta sección vamos a ver algunos ejemplos de editores de código los cuales son de código abierto como.

- **SublimeText.** Este editor de código lo podemos descargar desde el siguiente link https://www.sublimetext.com/download
- **Atom.** Atom es un editor de código fácil de usar y nos permite autocompletar el código deseado además al igual que los otros editores me permite guardar bloques de códigos como snippeds. Este editor de códifo lo podemos descargar desde el siguiente link.
  https://atom-editor.cc/

Es importante establecer en el editor que codificación de caracteres queremos utilizar. La mejor opción para el desarrollo Web es utilizar la codificación UTF-8, esta codificación es la estándar que utiliza Internet y sistema Linux, Windows y MacOS.

Para este curso utilizaremos el editor de código ATOM, ustedes pueden buscar otros editores de código descargarlo y utilizarlo.

Servidores Web

Un servidor web es un software y hardware que utiliza HTTP (Hypertext Trasfer Protocol) y otros protocolos para responder a las solicitudes de las clientes realizadas a través de la www. SU función principal es servir contenido web, como páginas HTML, archivos multimedia y aplicaciones web, a los usuarios.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 10.

Servidor Web

Nota: La imagen representa un servidor web. Imagen tomada de https://i0.wp.com/somarsa.com/wp-content/uploads/2018/04/serveur-dedie2.jpg?resize=768%2C360&ssl=1

Un servidor web consta de dos partes.

- **Software del servidor Web.** Programas que manejan alas solicitudes HTTP y HTTPS y entregan el contenido solicitado.
- **Hardware del Servidor Web.** Computadora física donde se ejecuta el software del servidor donde se almacenan los datos del sitio web.

Servidor Web Locales

Un servidor web local es un software que permite a una computadora actuar como un servidor para alojar página web y aplicaciones durante el desarrollo. Esto te permite probar y desarrollar tu sitio web en tu máquina local antes de que se desplegada en un servidor remoto.

Un servidor web local imita las funciones de un servidor web real, pero a diferencia esta se ejecuta en tu propio computador, este entorno te permite.

- Probar y depurar tu sitio web sin necesidad de una conexión a Internet.
- Configurar y experimentar con tu servidor sin afectar un sitio web en producción
- Simular un entorno de producción lo más parecido posible.

Para poder emular el servidor local tenemos algunas opciones y cada uno necesita una configuración especial.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- **XAMPP.** Es una herramienta de desarrollo que nos permite probar el desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet. Xampp es una distribución de apache que incluye diferentes softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que los constituyen:

  Linux: Es el sistema operativo donde estará instalado nuestra aplicación.

  Apache: Es el servidor de código abierto más usado para la entrega de contenido web. Las aplicaciones del servidor son ofrecidas como software libre por la Apacha Software Foundation.

  MySQL/MariaDB: XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje de programación PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB.

  PHP: Es un lenguaje de programación de código de lado del servidor que permite crear aplicaciones web. Es independiente de la plataforma y soporta varios sistemas de base de datos.

  Perl: Es lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.
- **WampServer:** Similar a XAMPP, pero específico para Windows.
- **MAMP:** Paquete para macOS que incluye Apache, MySQL y PHP
- **Local by Flywhel:** Herramienta de desarrollo local para sitios WordPress
- **Node.js con Express:** Servidor web basado en JavaScript

Para nuestra guía usaremos el servidor local XAMPP para alojar los ejercicios que estaremos trabajando durante esta guía.

INTRODUCCIÓN A HTML5

Como habíamos dicho HTML es un lenguaje de marcado o etiquetado, eso quiere decir que consta de texto, que define los contenidos reales de la página web y de marcas especiales conocidas como etiquetas o tags, estas etiquetas permiten dar significado al texto o al contenido, así como indicar algún tratamiento especial sobre dicho texto.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Una de las ideas básicas del lenguaje de marcado es estructurar el contenido mediante dichos tags o etiquetas. Dichas etiquetas pueden tener o asociarse con distintos atributos. De esta manera podemos especificar características de formato, tipo de información y estas características puedan ser procesados por un programa en este caso por un navegador web. Vamos a realizar un ejemplo donde vamos a crear nuestro propio lenguaje de etiquetado, queda recalcar que el ejemplo no es un lenguaje HTML.

<oscuro>Esta sección se muestra en negrita </oscuro>y esta se muestra<importante>en cursiva</importante>

Salida de pantalla

"Esta sección se muestra en negrita" y esta se muestra en *cursiva.*

Lo que hace el estándar HTML5 es especificar qué marcas deben utilizarse para escribir un documento web y el significado de cada una de ellas. De esta forma el navegador podrá interpretar cada una de ellas.

Como sabemos, todo documento HTML debe ir delimitado por un par de etiquetas en este caso de inicio y cierre **<html></html>.** Dentro de estas etiquetas tenemos dos partes bien diferenciadas la cabecera la cual está 8delimitada por las siguientes etiquetas **<head></head>,** en esta sección vamos a insertar todos los metadatos de nuestra página web más adelante veremos a profundidad que tipo de elementos van dentro de esta sección, además no tenemos que olvidarnos que toda la información que este dentro de esta sección no se visualizara en ningún lugar de nuestra página web.

La otra sección es el cuerpo del documento HTML **<body></body>** y todos los elementos que estén dentro de estas dos etiquetas se podrán ver en el navegador. Estas son las etiquetas básicas que encontraremos dentro de un HTML.

Estructura Global

Los documentos HTML siempre se encuentran estrictamente organizados y se le de arriba hacia abajo. Cada parte del documento está muy bien diferenciado, declarado y determinada por etiquetas específicas, en esta sección de la guía vamos a ver como construir la estructura estándar de un documento HTML y las nuevas etiquetas semánticas que fueron incorporadas en HTML5.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

<!DOCTYPE>

La primera etiqueta que vamos a escribir es doctype esta etiqueta permite decirle al navegar el tipo documento que estamos creando esta etiqueta fue insertada en HTML5 e insertarlo es muy sencillo.

Tabla 2.

Doctype

<!DOCTYPE html>

Nota: En la tabla vemos como se implementa el comando doctype en HTML.

<html>

Luego de declarar el doctype o el tipo de documento que se va ejecutar, empezamos a estructurar el documento HTML. Cabe recalcar, que os documentos HTML siempre están estructurados tipo árbol teniendo como raíz el elemento**<html>.** Este elemento envolverá el resto del código de la página web.

Tabla 3.

Etiqueta html

<!DOCTYPE html> **<html lang="es" dir="ltr">** **</html>**

Nota: Usando el elemento <html>

El atributo **Lang** en la etiqueta de apertura de <html> es uno de los atributos que se necesita especificar en HTML5. Este atributo nos indica el idioma que vamos a usar en nuestro documento HTML, el siguiente atributo es **dir,** este atributo indica en qué dirección va a hacer escrito el documento HTML en este caso de izquierda a derecha.

<head>

Continuando con la implementación de nuestra plantilla, en secciones anteriores hablamos de que un documento HTML tiene dos partes, una cabecera y un cuerpo. En

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

esta sección se hablará del cabecero, esta sección se encierra dentro del elemento **<head>.**

El elemento **<head>**, traducido al español es la cabeza por eso es la etiqueta que va primero dentro del elemento **<html>** y de la misma forma como algunas de las etiquetas se utiliza una etiqueta de apertura y una de cierre.

Tabla 4.

head

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>

</head>

</html>
```

Nota: La tabla nos muestra la estructura head en HTML.

La etiqueta **<head>,** es uno de los elementos que no ha cambiado y se viene utilizando desde versiones anteriores y su propósito viene siendo el mismo. Dentro de las etiquetas definiremos el nombre de la página web, de la misma forma declararemos el tipo de caracteres que se va a utilizar dentro de la página web, también podemos implementar cierta información con metadatos como información general acerca del documento, incorporaremos los archivos externos con estilos, códigos JavaScript, incluso imágenes necesarias para crear la página web en la pantalla.

A excepto del título y algunos de los iconos, el resto de información incorporada de esta sección del documento no son visibles para el usuario.

<body>

La siguiente sección que es la parte principal de nuestro documento HTML, denominado cuerpo. Todo lo que se implemente dentro de esta etiqueta podrá ser visible para el

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

usuario. La etiqueta que se utiliza es **<body>** y esta no a cambiado en relación con las versiones anteriores.

Tabla 5.

Etiqueta body

```
<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
</head>
<body>

</body>

</html>
```

Nota: La tabla representa el uso de la etiqueta body.

Como podemos observar en la **tabla 5,** tenemos una estructura HTML simple, pero al mismo tiempo compleja. El código HTML no esta formado por un conjunto de instrucciones secuenciales.

HTML es un lenguaje de etiquetados, eso quiere decir que dentro vamos a tener un listado de elementos que usualmente se utilizan en pares y ale vez pueden ser anidados (contenidos uno dentro de otros). En la primera línea como lo explicamos indicamos el tipo de documento que estamos creando, inmediatamente abrimos la etiqueta **<html Lang=”es” dir=”ltr”>,** esta etiqueta y la de **</html>** al final de nuestro documento indica el comienzo y el final del código **HTML,** dentro de las etiquetas se inserta los dos elementos básicos:**<head>** para la cabecera y **<body>** para el cuerpo del documento. Como ya se explico estas dos etiquetas también se utilizan en pares.

<meta>

Después de haber construido el cuerpo del documento **HTML,** nos enfocaremos en la cabecera del documento, dentro de esta sección ira información como los **<meta>,** donde se definirá el juego de caracteres que se va a usar dentro de nuestra aplicación, esta etiqueta no se mostrara al usuario en ninguna parte.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Tabla 6.

Etiqueta meta

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>


  </body>
</html>
```

Nota: La tabla representa como se usa la etiqueta <meta>

En la versión 5 de **HTML,** se innovo alguno de los elementos como en la mayoría de los casos, es solo simplificación. La nueva etiqueta **<meta>** para la definición del tipo de caracteres es más corta y simple. Aquí podemos cambiar el tipo de codificación aquí se puede agregar distintas etiquetas **<meta>** como description o keywords en la cual se puede definir otros aspectos de la página web, como se muestra en la **tabla 7.**

Tabla 7.

Elementos <meta>

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="description" content="Ejemplo de HTML5">
    <meta name="keywords:" content="HTML5, CSS3, Javascript">


  </head>
  <body>


  </body>
</html>
```

Nota: EN la tabla se observa la implementación de más elementos **<meta>**

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Este elemento **<meta>** representa a los metadatos que no pueden ser representados por otros elementos relacionados como metadatos HTML, como son <base>, <link>, <script>, <style> o <title>.

Hay varios tipos de etiquetas **<meta>** las cuales pueden ser incluidas para declarar información general sobre el documento, y toda esta información que se está proporcionando no se mostrara en la ventana del navegador, esta información implementada es solo importante para los motores de búsqueda y dispositivos que necesitan hacer una vista previa del documento u obtener un sumario de la información que contiene.

Con la nueva versión de HTML no es necesario cerrar las etiquetas simples con una barra lateral al final de la sentencia, pero recomendamos utilizarlas por razones de compatibilidad con los navegadores el código de la **Tabla 7** no muestra como cerrar las etiquetas simples en la **Tabla 8.** Se mostrará como se debe implementar.

Tabla 8.

Cierre de etiquetas simples

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8"/>
    <meta name="description" content="Ejemplo de HTML5"/>
    <meta name="keywords:" content="HTML5, CSS3, Javascript”/>
  </head>
  <body>
  </body>
</html>
```

Nota: La tabla nos muestra cómo se debe cerrar una etiqueta simple.

<title>

La etiqueta **<title>,** lo único que hace es poner un título a nuestra página web, dentro de la etiqueta **<head>**, es lo único que se puede ver y la información se puede ver en la pestaña del navegador de usuario.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Tabla 9.

Etiqueta title

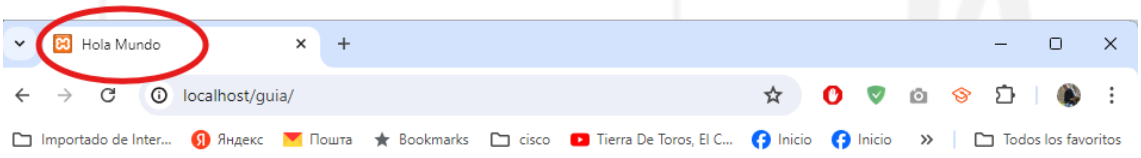
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="" content="">
    <title>Hola Mundo</title>
  </head>
  <body>


  </body>
</html>
```

Nota: La tabla muestra la implementación de la etiqueta **<title>**

El texto que está dentro de la etiqueta **<title>** es el título del documento que estamos creando. Esta información se muestra en la pestaña del navegador que se muestra en la **Figura11.**

Figura 11.

Etiqueta title.



Nota: La imagen muestra la salida en pantalla de la etiqueta **<title>**

<link>

Otro elemento importante dentro de la etiqueta **<head>** es la etiqueta **<link>.** Este tag es usado para incorporar estilos, código JavaScript, imágenes o iconos desde archivos externos. Una de las características más comunes para **<link>** es la implementación o incorporación de hojas de estilos CSS.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Tabla 10.

Elemento link

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="" content="">
    <title>Hola Mundo</title>
    <link rel="stylesheet" href="/css/estilos.css">
  </head>
  <body>
  </body>
</html>
```

Nota: En la tabla nos muestra cómo se usa el elemento link en la cabecera del documento web.

En la última versión de HTML5, ya no se debe especificar el tipo de estilos que estamos insertando, por lo que en esta versión el atributo **type** fue eliminado. Por consecuente solo se necesita dos atributos **rel y href:** el primer atributo **rel** significa la relación que hay entre el archivo HTML y el archivo que se está insertando en este caso **rel** tiene el valor de **stylesheet** eso quiere decir el **estilos.css** es un archivo CSS, el atributo **href**, nos indica la ruta o el lugar donde se encuentra el archivo **estilos.css.**

Con esta última implementación de código hemos terminado la estructura básica de la cabecera. A continuación, trabajaremos en la estructura del cuerpo.

Estructura del cuerpo.

La estructura que va dentro de las etiquetas **<body>**, es lo que se va a visualizar en el navegador Este código producirá la página web.

Para poder organizar y construir las páginas web HTML nos ofrece distintas formas para poder desarrollarlas las cual permite organizar toda la información que se va a mostrar en el navegador. Para esta organización se puede utilizar algunas etiquetas como **<table>.** Las tablas

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

permiten a los desarrolladores organizar de mejor manera el contenido de las páginas web, texto, imágenes y herramientas dentro de las filas y columnas.

Al principio con la implementación de las **tablas,** se pudo organizar la información de una manera en el cual el usuario tenía una mejor experiencia de usuario. Con la evolución de la tecnología y las nuevas versiones de HTML fueron apareciendo nuevas etiquetas que la remplazaron, ya que se obtiene el mismo resultado, pero con una menor cantidad de código, ayudando al desarrollo de las páginas web, permitiendo la portabilidad y ayudando al mantenimiento de los sitios web.

Con las nuevas versiones apareció la etiqueta ** **, esta etiqueta es una etiqueta contenedora quiere decir que esta puede contener otro tipo de etiquetas dentro de ella de esa manera se puede facilitar la organización de la información, además con la integración de CSS y JS la etiqueta ** ,** puede contener tablas, formularios, imágenes texto, esta etiqueta quiere decir división, y cada una de esas divisiones actúan como las celdas de una tabla.

Como ya se a dicho y se ha expuesto HTML5 incorpora nuevos elementos el cual van a ayudar a identificar cada sección del documento y no ayuda a organizar el cuerpo del documento.

Cómo usamos estos nuevos elementos depende de nosotros, pero las etiquetas otorgadas a cada uno de ellos nos van a ayudar a entender sus funciones. Normalmente un sitio web o aplicación web está dividida entre varias áreas visuales para tener una mejor experiencia de usuario y facilitar la interactividad. Las palabras claves que representan cada nuevo elemento de HTML5 están íntimamente relacionadas con estas áreas, como veremos pronto.

Organización.

La organización de los elementos de un sitio web depende mucho de usuario final y del desarrollador.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 12.

Representación visual de un diseño web clasico.

Nota: La imagen representa un diseño común encontrado en la mayoría de los sitios web. Imagen tomada de https://cursos.tienda/estructura-del-cuerpo-2/

Como se muestra en la Figura 12. En la parte superior de la estructura tenemos la **Cabecera,** en esta sección usualmente se coloca el logo, titulo, subtítulos y una corta descripción del sitio web.

En la parte inferior de la cabecera tenemos la **Barra de Navegación,** en esta sección la mayoría de los desarrolladores implementan un menú o una lista de enlaces el cual facilita la navegación por el sitio web. Mediante esta sección los usuarios pueden ser guiados desde esta barra hacia las diferentes páginas o documentos, normalmente implementadas en el mismo sitio web.

La parte principal de nuestra estructura está bien marcada, aquí implementaremos el contenido más relevante de una página web, muchas veces esta sección está dividida en diferentes secciones eso quiere decir en varias filas y columnas. En la figura 12 se observa que se ha utilizado dos secciones, la primera sección se va a implementar la **Información Principal** y en la segunda sección tenemos una **Barra Lateral,** como ya lo dijimos en esta sección y es adaptable a lo que el desarrollador lo implemente acorde a sus necesidades, el cual puede insertar varias columnas y filas teniendo varios bloques donde se puede implementar variedad de contenido. Ya dicho en esta sección es la parte principal del sitio web.

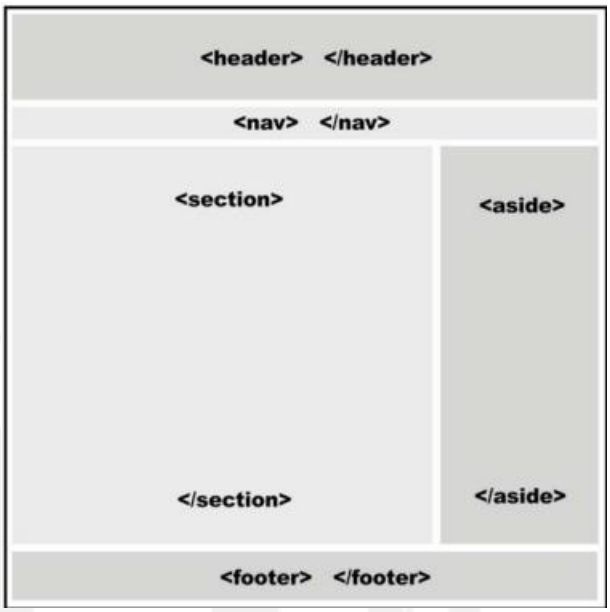
En la parte inferior del maquetado está el **<footer>** o pie de página, en esta sección va información sobre derechos reservados, términos y condiciones e información adicional que el desarrollador considere importante compartir.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Una vez que hemos visto cual es la maquetación de un sitio web estándar a continuación vamos a ver que etiquetas HTML vamos a utilizar para cada sección.

Figura 13.

Representación de un diseño web utilizando HTML



Nota: La imagen representa a un diseño web utilizando etiquetas HTML. Imagen tomada del sitio web https://cursos.tienda/estructura-del-cuerpo-2/

Etiquetas semánticas

HTML5 introdujo varias etiquetas semánticas para mejorar la estructura y accesibilidad de las páginas web. **De esta manera se estructura mejor cada página web** (Rubiales Gómez, Curso de desarrollo Web. HTML, CSS y JavaScript. , 2021)**.**

<header>

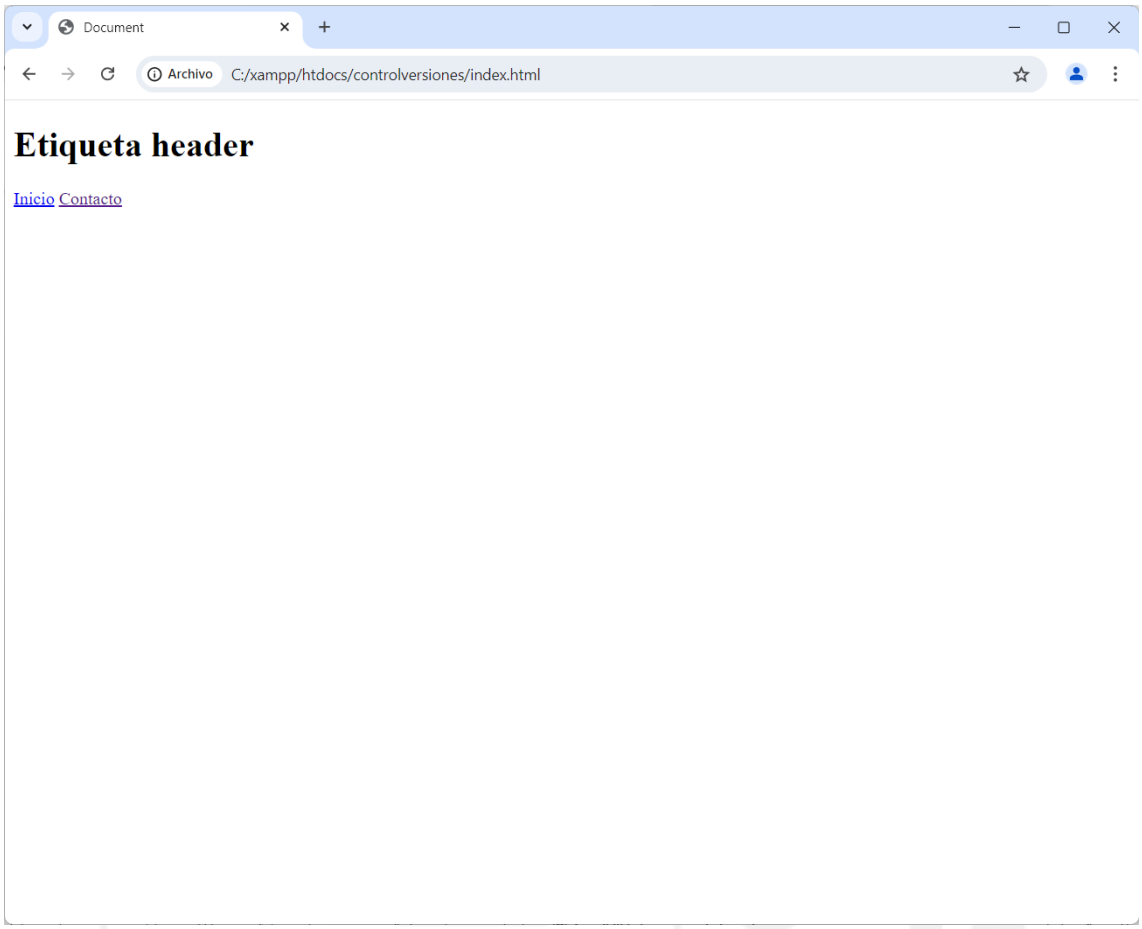
Utilizada para definir la cabecera de una página o sección. Normalmente contiene logotipos, menús de navegación, y otros elementos introductorios.

```
<header >
        <h1>Etiqueta div</h1>
        <nav>
            <a href="#">Inicio</a>
            <a href="">Contacto</a>
        </nav>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
</header>
```

Figura 14.
Etiqueta header



Nota: La imagen nos muestra la etiqueta header donde se puede colocar un menú

Etiqueta <nav>

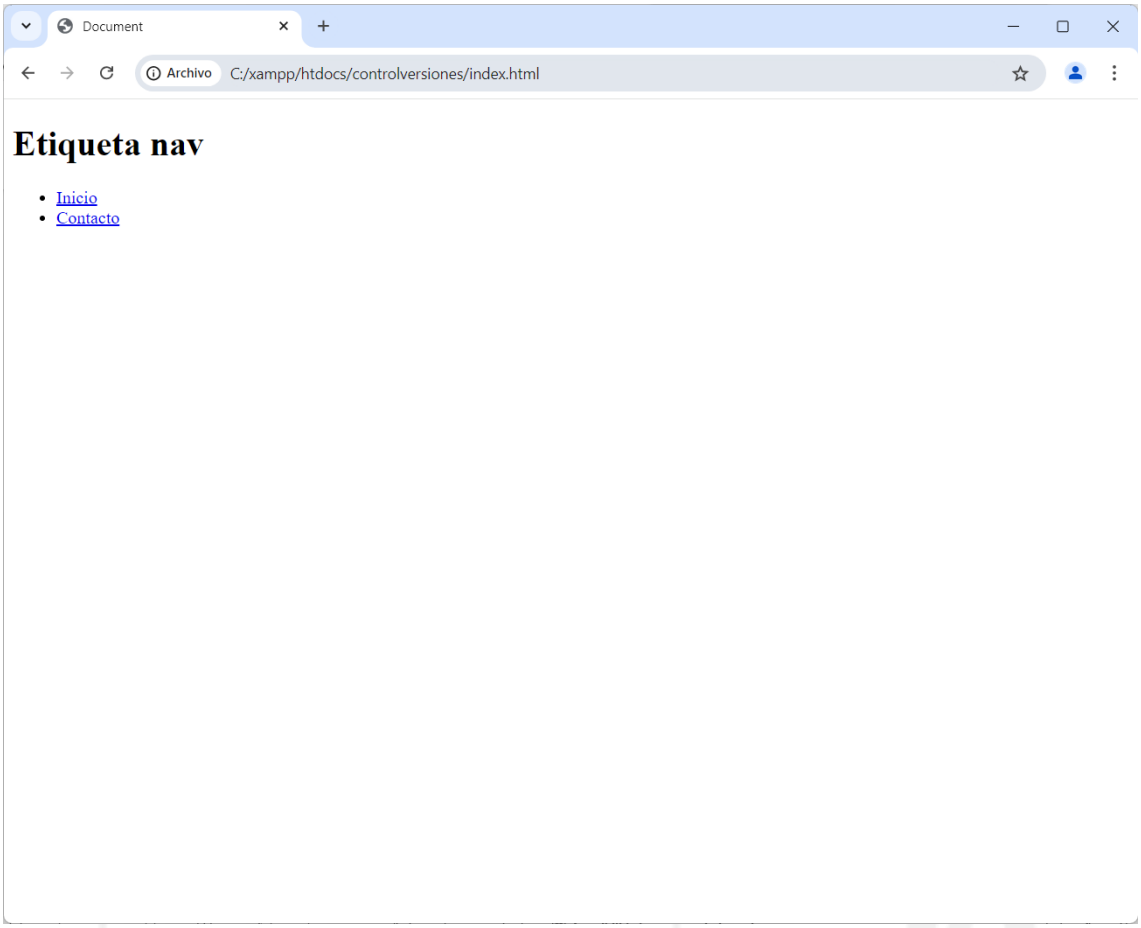
Representa un bloque de enlaces de navegación que dirigen a diferentes partes del sitio o a otros recursos relacionados.

```
<header >
        <h1>Etiqueta nav</h1>
        <nav>
            <ul>
                <li><a href="#">Inicio</a></li>
                <li> <a href="#">Contacto</a></li>
            </ul>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        </nav>
    </header>
```

Figura 15.
Etiqueta nav



Nota: La imagen me muestra la salida en pantalla de la etiqueta <nav>

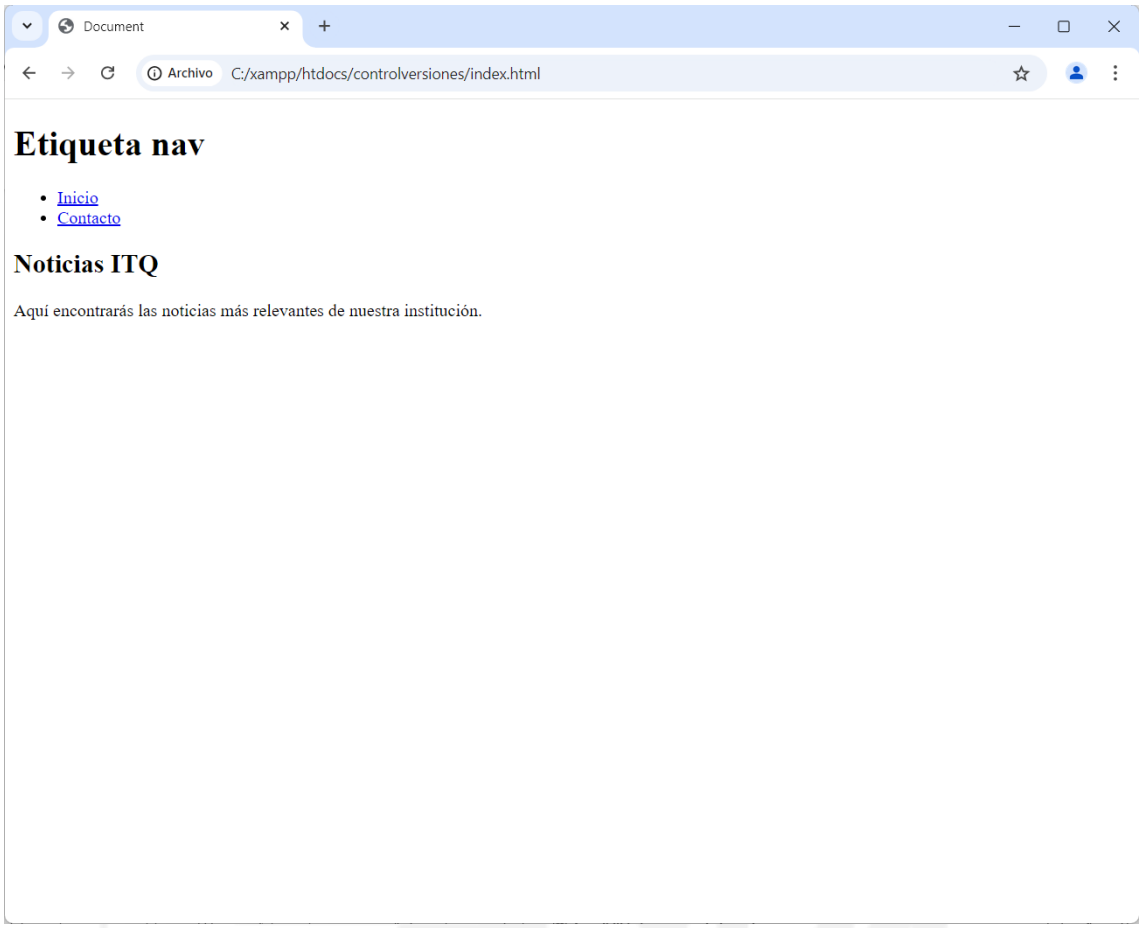
Etiqueta <section>

Esta etiqueta define una sección temática del contenido, normalmente con un título. Es ideal para separar partes distintas del contenido principal dentro de una misma página (López Sanz, 2016).

```
<section>
        <h2>Noticias ITQ</h2>
       <p>Aquí encontrarás las noticias más relevantes de nuestra institución. </p>
    </section>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 16.
Etiqueta <section>



Nota: La imagen nos muestra la salida de pantalla implementado la etiqueta <section> que es el contenido principal de nuestra página web, y esta desde Noticias ITQ.

Estructura div

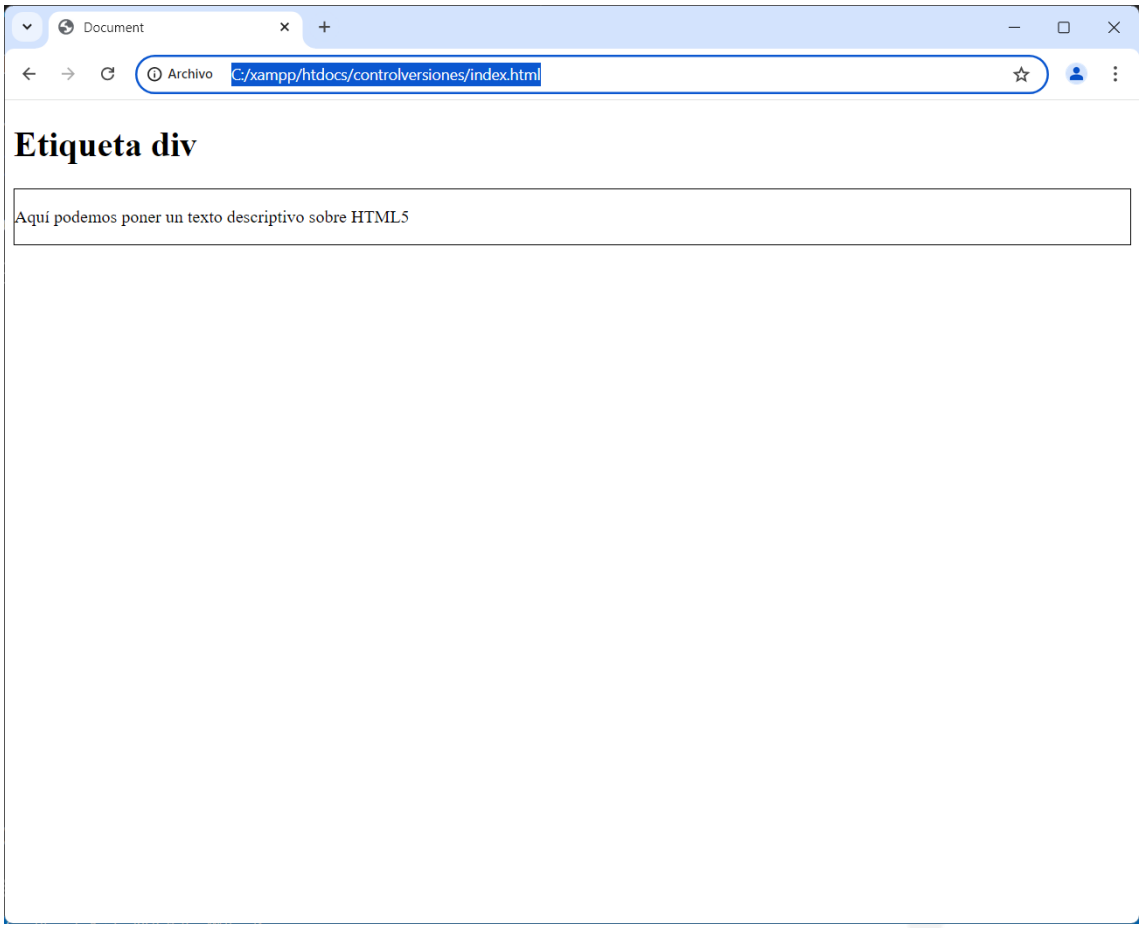
La etiqueta se utiliza para definir una sección o división de contenido en una página web. Aunque no hay restricciones específicas sobre cómo usar , lo ideal es aplicarla cuando se necesita separar contenido que no tiene un significado semántico claro. Es decir, debería emplearse solo cuando no se puede usar una etiqueta más significativa como <main> o <nav>, que aportan mayor claridad sobre la función del contenido (Arias, 2016).

```
<h1>Etiqueta div</h1>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
<div style="border: 1px; border-style: solid; border-color:
black;">
    <p>Aquí podemos poner un texto descriptivo sobre HTML5</p>
</div>
```

Figura 17.
Etiqueta div



Nota: La imagen nos muestra la salida de pantalla de la etiqueta donde se encierra en un marco de color negro.

Etiqueta <article>

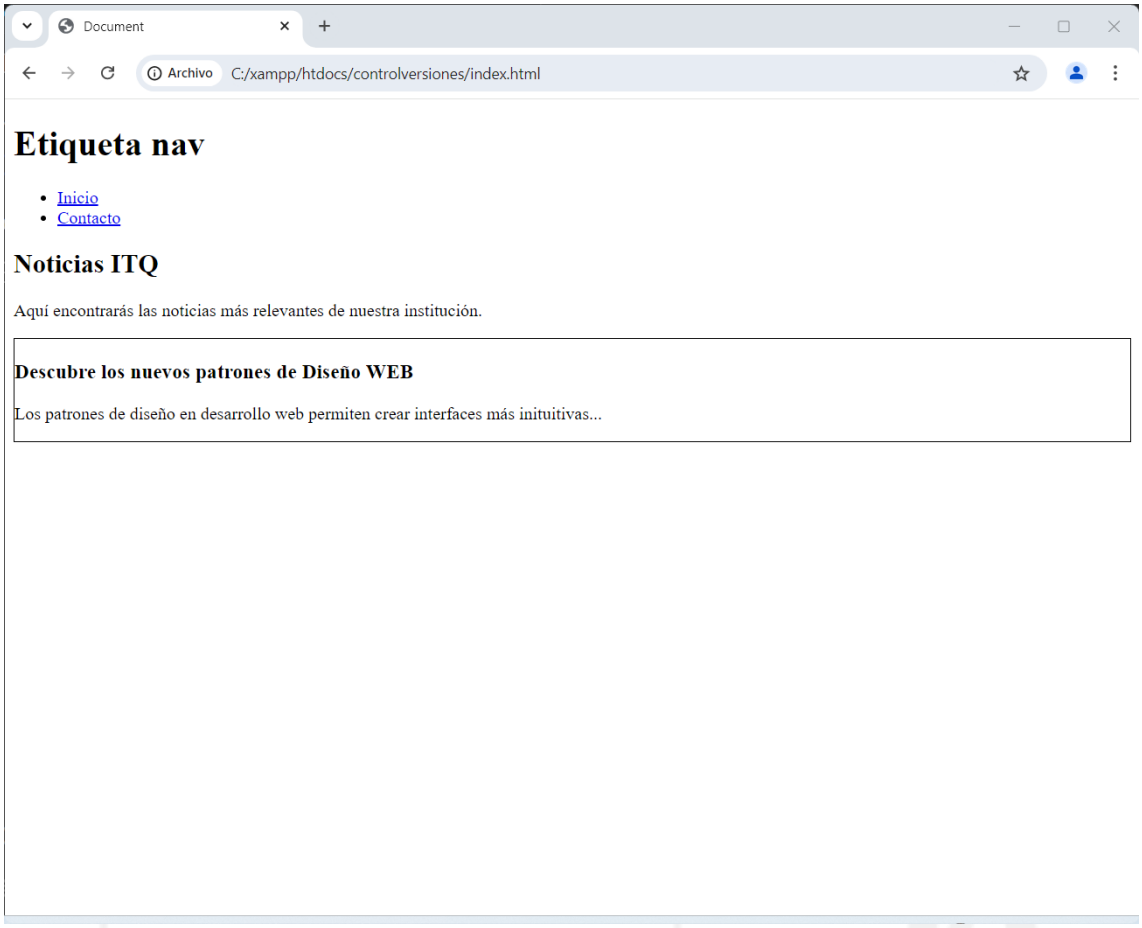
Esta etiqueta es utilizada para contenidos independientes que podrían ser distribuidos de forma separada, como artículos de blog, comentarios o entradas de foros.

```
<article>
            <h3>Descubre los nuevos patrones de Diseño WEB</h3>
            <p>Los patrones de diseño en desarrollo web permiten
crear interfaces más intuitivas...</p>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        </article>
```

Figura 18.
Etiqueta <article>



Nota: La imagen nos muestra la sección <article>, que esta encerrado dentro de una margen de color negro.

Etiqueta <footer>

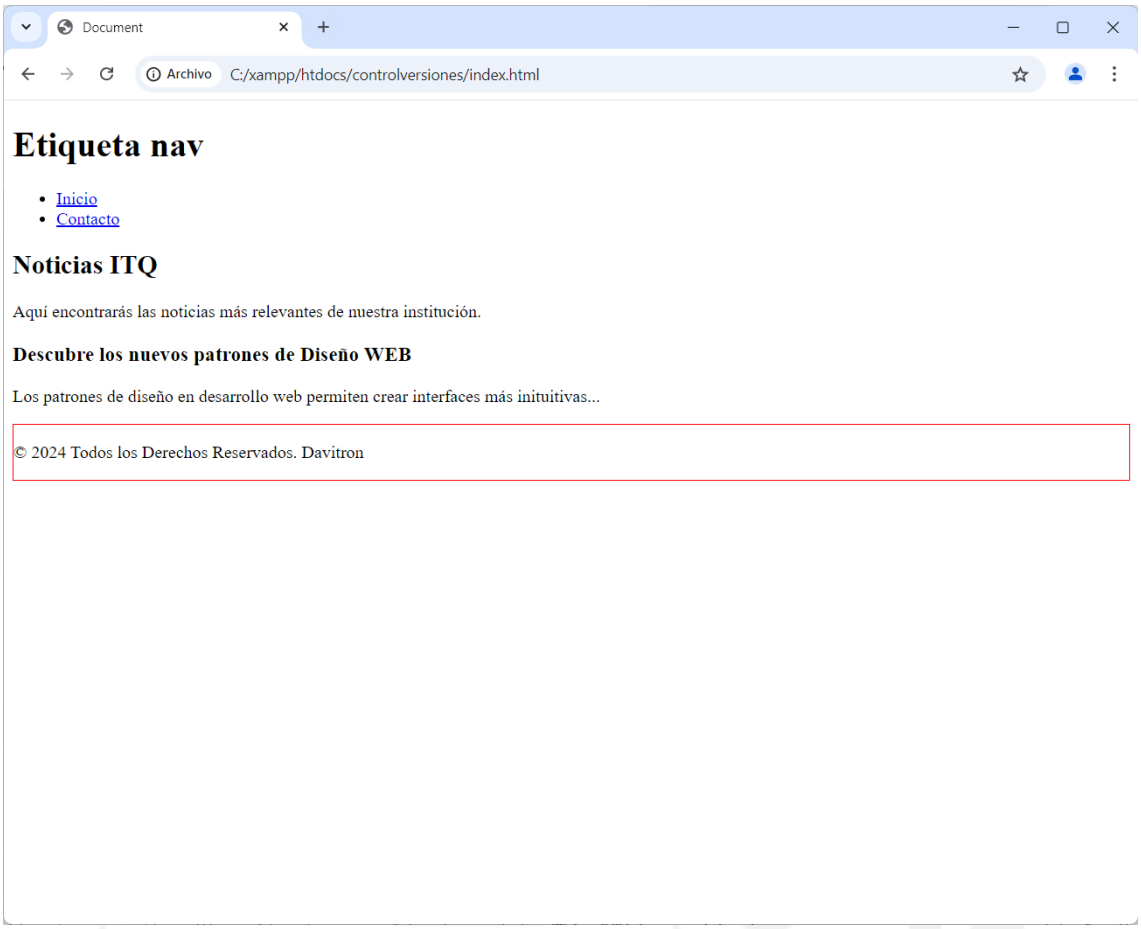
La etiqueta <footer> representa el pie de página de un documento o sección. Suele contener información sobre el autor, derechos de autor, enlaces de contacto, etc.

```
<section>
        <footer style="border: 1px; border-style: solid; border-color: red;">
            <p>&copy; 2024 Todos los Derechos Reservados. Davitron </p>
        </footer>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
</section>
```

Figura 19.
Etiqueta <footer>



Nota: La imagen nos muestra el pie de pagina de nuestro proyecto y esta encerrado dentro de un rectángulo rojo.

Etiquetas de Texto y estilo

En HTML5 nos proporciona etiquetas específicas para dar formato al texto y mejorar la claridad y presentación del contenido (Gómez Delgado, 2023).

Encabezados <h1> a <h6>

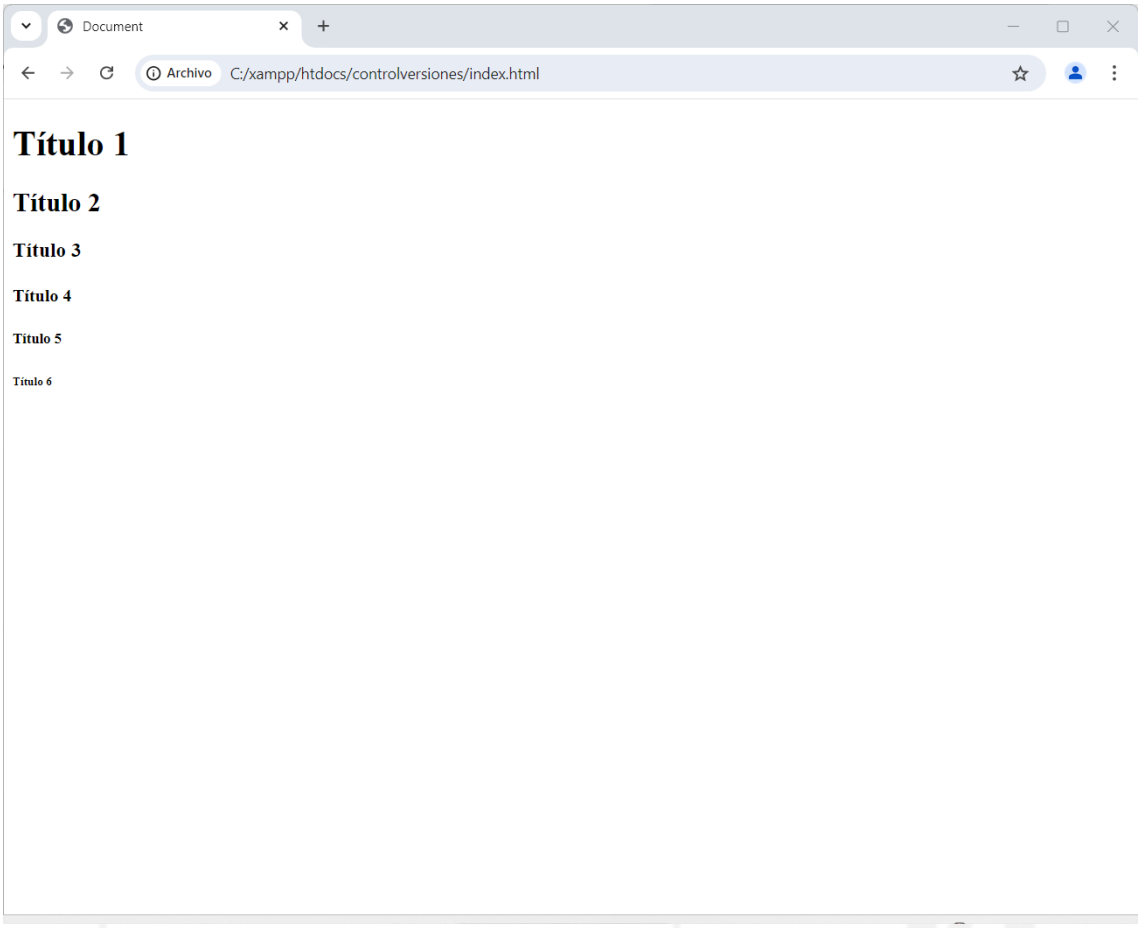
Estas etiquetas se utilizan para definir títulos y subtítulos de diferentes niveles, siendo la etiqueta <h1> la más importante.

```
<h1>Título 1</h1>
    <h2>Título 2</h2>
    <h3>Título 3</h3>
    <h4>Título 4</h4>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
<h5>Título 5</h5>

<h6>Título 6</h6>
```

Figura 20.
Etiqueta <h>



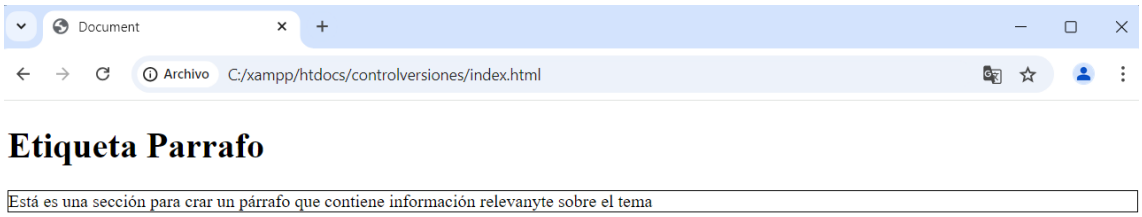
Nota: La imagen nos muestra las diferentes etiquetas de encabezado

Etiqueta párrafo <p>

Esta etiqueta define un bloque de texto, utilizando para agrupar frases y párrafos completos. Por defecto los navegadores le asignan un margen en la parte superior para separar un párrafo de otro.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 21.
Etiqueta <p>



Nota: La imagen nos muestra la sección de un párrafo implementado en html

Etiqueta enlace <a>

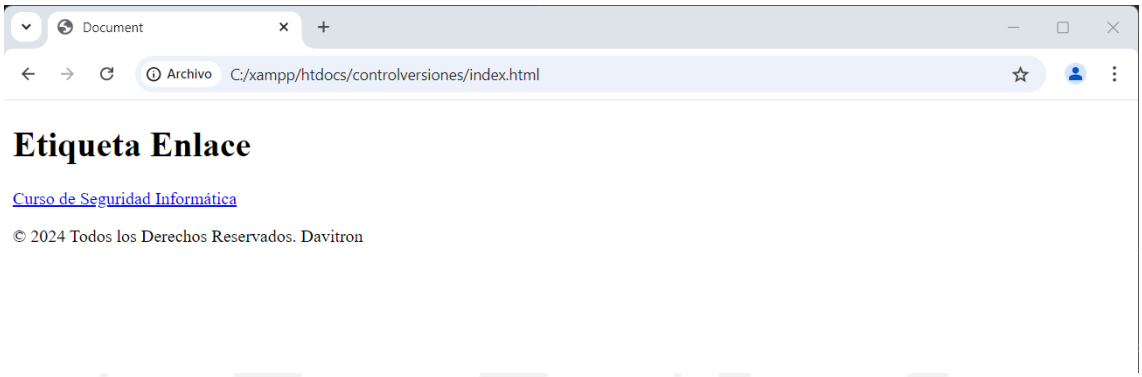
Esta etiqueta nos permite crear un enlace a otra página o recurso utilizando el atributo href. Este elemento por defecto se muestra en letras de color azul y subrayado en la parte inferior del texto (Hernández, 2014).

```
<!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>
<h1>Etiqueta Enlace</h1>
    <a href="https://youtu.be/ZDvxqqSoyZs?si=FNtCu-qbBSSyc-ex">Curso de Seguridad Informática</a>
    <section>
        <footer style="border: 1px; border-style: solid; border-color: red;">
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
            <p>&copy; 2024 Todos los Derechos Reservados.
Davitron </p>

        </footer>

    </section>

</body>

</html>
```

Figura 22.
Etiqueta <a>

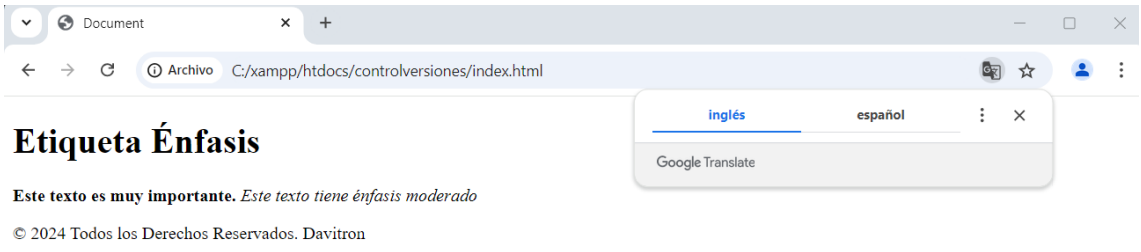


Nota: La imagen nos muestra la salida de pantalla de la implementación de la etiqueta <a> el elemento se encuentra en letras color azul.

Etiqueta énfasis ,

Estas etiquetas se usan para enfatizar un texto, la etiqueta agrega énfasis fuerte lo que nos muestra un texto en negrita, mientras que la etiqueta aplica un énfasis la letra de tipo en cursiva.

Figura 23.
Etiqueta <strong,>



Nota: La imagen nos muestra la implementación de las dos etiquetas de énfasis. y

Listados

A menudo, la información se debe representar como una lista de ítems. Por ejemplo, muchos de los sitios incluyen listados de libros, películas o términos y descripciones. Para crear estos listados, HTML ofrece los siguientes elementos.

**
-** Este elemento crea una lista de ítems sin orden. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems ** y ** y trabaja junto con el elemento.

<li>-Esta etiqueta nos permite definir cada uno de los ítems de la lista.

<ol>- Este elemento crea una lista ordenada de ítems. Este elemento puede incluir los atributos **reversed** para invertir el orden de los indicadores, **start** para determinar el valor desde el cual los indicadores tienen que comenzar a contar y **type** para determinar el tipo de indicador que queremos usar. Los valores disponibles para el atributo **type** son **1, a, A, i e I.**

<dl>- Este elemento crea una lista de términos y descripciones. Este elemento trabaja junto con los elementos **<dt> y <dd>** para definir los ítems de la lista. El elemento **<dl>** define la lista, el elemento **<dt>** define los términos y el elemento **<dd>** define las descripciones.

Para crear una lista depende de las características del contenido.

```
<!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>
     <header >
        <h1>Campeonato Ecuatoriano de Futbol</h1>
    </header>
    <section>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        <ul>
            <li>1. Barcelona SC</li>
            <li>2. Liga de Quito</li>
            <li>3. Emelec</li>
            <li>4. Independiente del Valle</li>
            <li>5. Deportivo Cuenca</li>
            <li>6. Orense</li>
            <li>7. Cumbaya FC</li>
            <li>8. Libertad SC</li>
            <li>9. El Nacional</li>
            <li>10. Aucas</li>
            <li>11. Macara</li>
            <li>12. Técnico Universitario</li>
            <li>13. Imbabura FC</li>
        </ul>
    </section>
    <section>
        <footer >
            <p>&copy; 2024 Todos los Derechos Reservados.
Davitron </p>
        </footer>
    </section>
</body>
</html>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 24.
*Lista desordenad *

Nota: La imagen nos muestra la implementación de una lista desordenada.

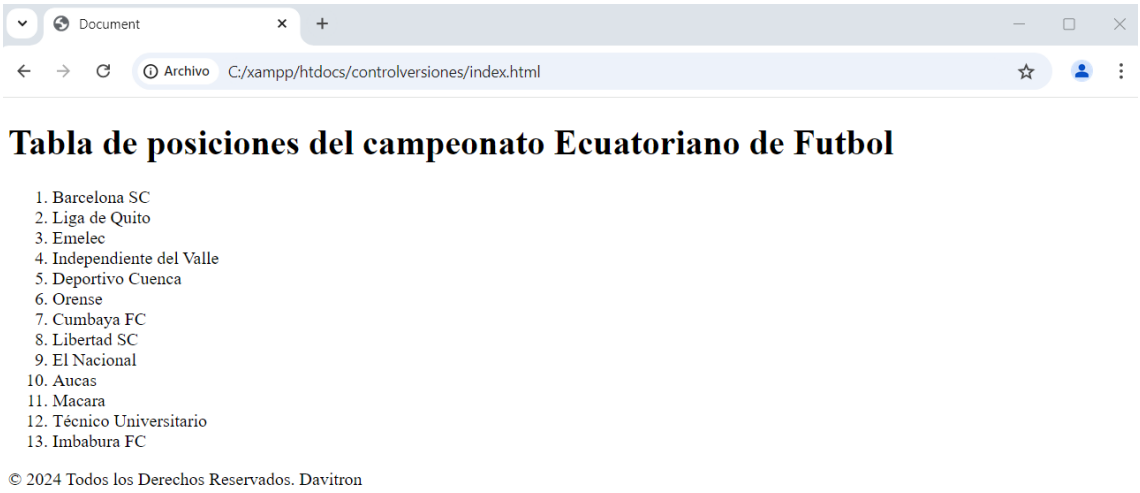
Si necesitamos declarar la posición de cada ítem, podemos crear una lista con la etiqueta **<ol>.** Esta etiqueta crea una lista de ítems en ele orden en el que se han declarado en el código, pero en lugar de usar puntos para identificarlos, les asigna un valor. Por defecto los indicadores se crean con números, pero podemos cambiarlos con el atributo **type**.

```
<ol>
        <li>Barcelona SC</li>
        <li>Liga de Quito</li>
        <li>Emelec</li>
        <li>Independiente del Valle</li>
        <li>Deportivo Cuenca</li>
        <li>Orense</li>
        <li>Cumbaya FC</li>
        <li>Libertad SC</li>
        <li>El Nacional</li>
        <li>Aucas</li>
        <li>Macara</li>
        <li>Técnico Universitario</li>
        <li>Imbabura FC</li>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        </ol>
```

Figura 25.
*Lista Ordenada *



Nota: La imagen muestra la tabla de posiciones del campeonato ecuatoriano de futbol.

Etiqueta citas <blockquote>

Esta etiqueta representa una cita en bloque de otra fuente. Puede incluir un atributo **cite** para especificar la fuente.

```
 <section>
        <blockquote cite="https://www.netapp.com/es/artificial-
intelligence/what-is-artificial-intelligence/">
            <p>La inteligencia artificial (IA) es la base a
partir de la cual se imitan los procesos de inteligencia humana
mediante la creación y la aplicación de algoritmos creados en un
entorno dinámico de computación.
                O bien, dicho de forma sencilla, la IA consiste
en intentar que los ordenadores piensen y actúen como los
humanos. </p>
        </blockquote>
    </section>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Etiqueta <table>

La etiqueta <table> en HTML se utiliza para crear tablas de datos. Las tablas permiten organizar información en filas y columnas, lo que es útil para presentar datos tabulados de manera estructurada y fácil de entender. A pesar de que en los diseños modernos se prefieren las tecnologías como CSS para el diseño y la estructura, las tablas siguen siendo muy importantes para mostrar información tabular.

Una tabla en HTML se compone de varias etiquetas que trabajan en conjunto para definir su estructura y contenido. A continuación, se describen cada uno de estos componentes con ejemplos para ilustrar su uso:

<table>- La etiqueta <table> es la etiqueta principal que envuelve todo el contenido de la tabla. Dentro de esta etiqueta, se colocan las demás etiquetas que definen las filas, columnas, encabezados, y el cuerpo de la tabla.

<tr>(Table Row- Fila de Tabla)-La etiqueta **<tr>** define una fila dentro de la tabla. Cada fila contiene una o más celdas de datos **<td>** o celdas de encabezado **<th>**.

<td> (Table data- Celda de tabla)- La etiqueta **<td>** se utiliza para definir una celda de datos dentro de una fila de la tabla. Estas celdas contienen el contenido que se desea mostrar en la tabla, como texto, imágenes, enlaces, etc.

<th> (Table header – Encabezado de Table)- La etiqueta **<th>** define una celda de encabezado en una tabla. Por lo general, se utiliza en la primera fila de una tabla para indicar los títulos de cada columna. El contenido de **<th>** suele estar en negrita y centrado.

<thead> (Table Head – Cabeza de la Tabla)- La etiqueta **<thead>** se utiliza para agrupar las filas de encabezado en una tabla. Esto es especialmente útil cuando se trabaja con tablas complejas que tienen múltiples filas de encabezado.

<tbody> (Table body – Cuerpo de la tabla)- La etiqueta **<tbody>** agrupa las filas que contienen los datos principales de la tabla. En combinación con **<thead>** y **<tfoot>**, ayuda a organizar y estructurar mejor la tabla.

<tfoot> (Table footer – Pie de la tabla)- La etiqueta **<tfoot>** se utiliza para agrupar las filas que contienen los resúmenes o totales al final de la tabla. Esto es útil cuando se necesita calcular totales o mostrar información agregada.

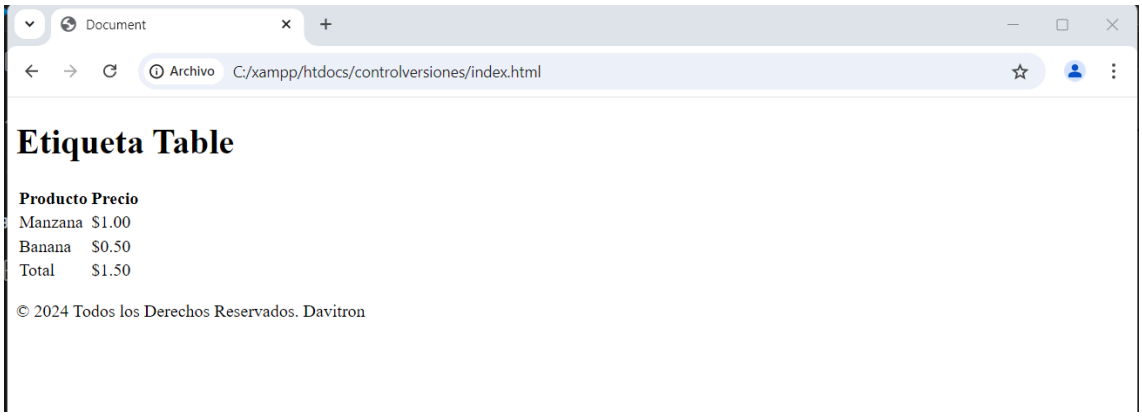
```
<section>
        <table>
            <thead>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
            <tr>
                <th>Producto</th>
                <th>Precio</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Manzana</td>
                <td>$1.00</td>
            </tr>
            <tr>
                <td>Banana</td>
                <td>$0.50</td>
            </tr>
        </tbody>
        <tfoot>
            <tr>
                <td>Total</td>
                <td>$1.50</td>
            </tr>
        </tfoot>
    </table>
</section>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 26.
Etiqueta <table>



Nota: La imagen nos muestra la ejecución del código donde se implementa una tabla, de producto y precio.

Atributos comunes de la etiqueta <table> y sus componentes

Border: El atributo border define el grosor del borde alrededor de la tabla y las celdas.

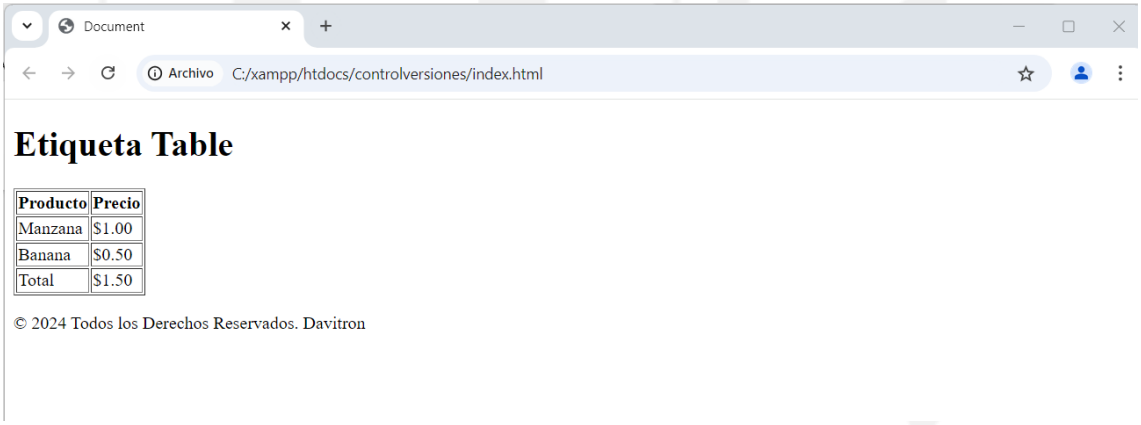
```
<section>
        <table border="1px">
            <thead>
                <tr>
                    <th>Producto</th>
                    <th>Precio</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Manzana</td>
                    <td>$1.00</td>
                </tr>
                <tr>
                    <td>Banana</td>
                    <td>$0.50</td>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
            </tr>
        </tbody>
        <tfoot>
            <tr>
                <td>Total</td>
                <td>$1.50</td>
            </tr>
        </tfoot>
    </table>
</section>
```

Figura 27.

Tabla HTML



Nota: La imagen nos muestra cómo se implementa una tabla con bordes.

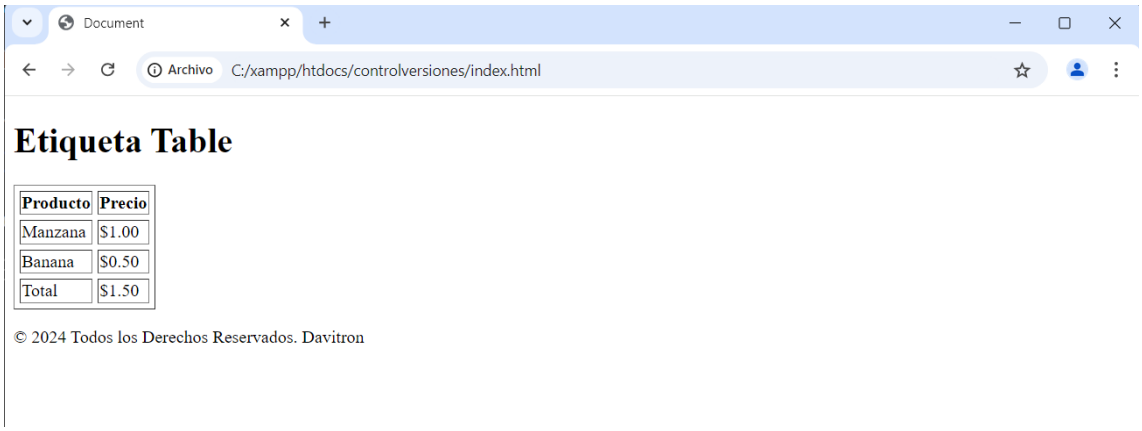
Cellpadding y cellspacing

Define el espacio interior dentro de cada celda, es decir, la distancia entre el contenido de la celda y sus bordes.

La etiqueta cellspacing define el espacio entre las celdas de la tabla

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 28.
Tabla con espacio



Nota: La imagen nos muestra cómo se implementa espacios con los argumentos dados en el código.

Colspan

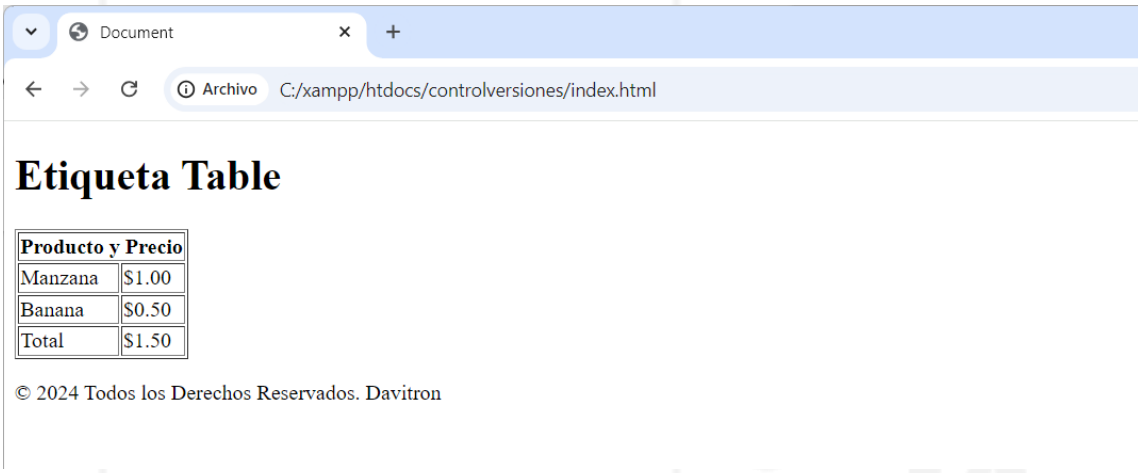
Este argumento permite que una celda se extienda sobre varias columnas.

```
<table border="1px">
            <thead>
                <tr>
                    <th colspan="2">Producto y Precio</th>

                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Manzana</td>
                    <td>$1.00</td>
                </tr>
                <tr>
                    <td>Banana</td>
                    <td>$0.50</td>
                </tr>
            </tbody>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        <tfoot>
            <tr>
                <td>Total</td>
                <td>$1.50</td>
            </tr>
        </tfoot>
    </table>
```

Figura 29.
Combinación de columnas



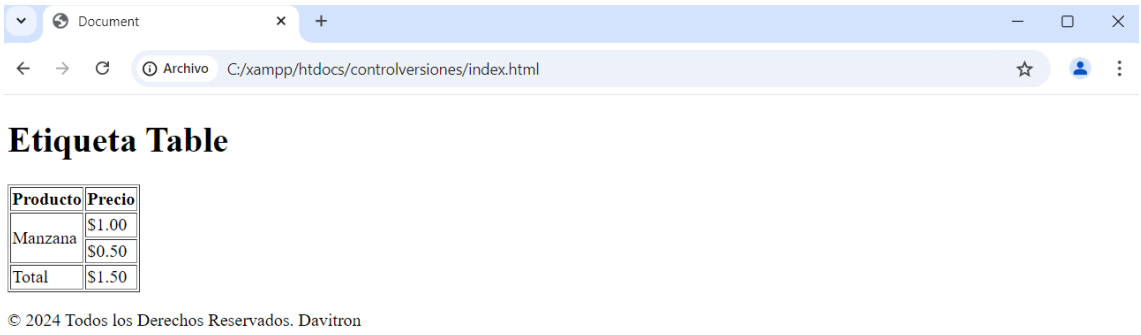
Nota: La imagen nos muestra como combinar dos columnas.

Rowspan

Este argumento permite que una celda se extienda sobre varias filas

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 30.
Argumento Rowspan



Nota: La imagen nos muestra como combinar dos filas en una tabla

Etiqueta <img>

La etiqueta <img> en HTML es fundamental para insertar imágenes en una página web, lo que permite enriquecer el contenido visual y mejorar la experiencia del usuario. A continuación, se proporciona una explicación detallada de la etiqueta <img>, sus atributos más importantes, y ejemplos prácticos para ilustrar su uso.

La etiqueta <img> es una de las más utilizadas en HTML, ya que permite la inserción de imágenes en una página web. A diferencia de otras etiquetas, <img> es una etiqueta "vacía", lo que significa que no tiene una etiqueta de cierre y todo su contenido está dentro de la misma etiqueta de apertura.

Para utilizar correctamente la etiqueta <img>, es importante comprender los atributos que se pueden emplear para controlar su comportamiento y presentación. Aquí te detallo los atributos más relevantes:

src (Source - Fuente)

El atributo src es quizás el más importante de la etiqueta <img>, ya que especifica la URL de la imagen que se desea mostrar. Esta URL puede ser relativa o absoluta:

URL relativa:

Se refiere a la ubicación de la imagen en relación con la ubicación del archivo HTML.

URL absoluta:

Es una dirección completa, incluyendo el protocolo (http, https) y el dominio.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

alt (Alternate Text - Texto Alternativo)

El atributo alt proporciona un texto alternativo para la imagen. Este texto se muestra en caso de que la imagen no se pueda cargar, y es esencial para la accesibilidad, ya que los lectores de pantalla lo utilizan para describir la imagen a usuarios con discapacidades visuales.

width y height (Ancho y Alto)

Los atributos width y height se utilizan para definir las dimensiones de la imagen en píxeles. Es importante especificar ambos atributos para evitar que el navegador tenga que recalcular el diseño de la página cuando se carga la imagen.

title (Título de la Imagen)

El atributo title permite agregar un título descriptivo a la imagen. Este título aparece como un "tooltip" cuando el usuario coloca el cursor sobre la imagen.

loading (Carga de Imágenes)

El atributo loading es una característica relativamente nueva que permite diferir la carga de imágenes fuera de la vista inicial, lo que puede mejorar el rendimiento de la página. Los valores posibles son lazy (carga diferida) y eager (carga inmediata).

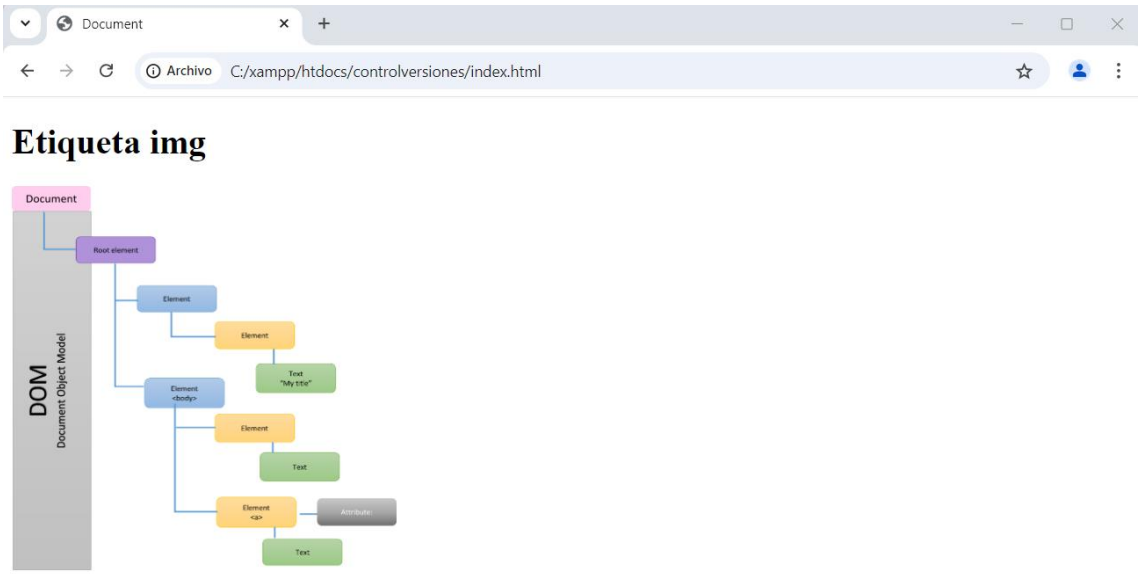
```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>


<body>
     <header >
        <h1>Etiqueta img</h1>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
    </header>


    <section>
        <div>
            <img src="dom.png" alt="domJS" title="DOM JSP"
width="350" height="350">
        </div>


    </section>


    <section>
        <footer >
            <p>&copy; 2024 Todos los Derechos Reservados.
Davitron </p>
        </footer>
    </section>


</body>
</html>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 31.
Etiqueta <img>



Nota: La imagen nos muestra como incrustar una imagen en html5

Autoevaluación 1

1. ¿Quién es conocido como el creador de HTML?

   Tim Berners-Lee es reconocido como el creador de HTML. Desarrolló el lenguaje en 1991 mientras trabajaba en el CERN para facilitar la compartición de documentos en la web.

2. ¿En qué año se lanzó la primera versión de HTML?

   La primera versión de HTML se lanzó en 1991.

3. ¿Qué organización se encarga de desarrollar y estandarizar HTML?

   El World Wide Web Consortium (W3C) es la organización que desarrolla y estandariza HTML.

4. ¿Qué versión de HTML introdujo las etiquetas semánticas como <header> y <footer>?

   HTML5 introdujo las etiquetas semánticas como <header> y <footer>.

5. ¿Qué cambio importante trajo HTML5 respecto a las versiones anteriores?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

HTML5 trajo una estructura más semántica, eliminó la necesidad de complementos externos para videos y audio, e introdujo nuevas APIs para gráficos y almacenamiento local.

6. ¿Qué es un cliente en la arquitectura cliente-servidor?

   Un cliente es cualquier dispositivo o aplicación que solicita servicios o recursos de un servidor, como un navegador web que solicita una página HTML.

7. ¿Qué es una solicitud HTTP?

   Una solicitud HTTP es un mensaje enviado desde el cliente al servidor solicitando un recurso, como un archivo HTML, CSS, imagen o datos de una base de datos.

8. ¿Qué etiqueta se utiliza para definir el título de una página web en HTML?

   La etiqueta <title> se utiliza para definir el título de la página, que aparece en la pestaña del navegador.

9. ¿Cuál es la diferencia entre <ol> y ?

    crea una lista ordenada (numerada), mientras que <ul> crea una lista no ordenada (con viñetas).

10. ¿Para qué se utiliza la etiqueta <table> en HTML?

    La etiqueta <table> se usa para crear tablas, organizando datos en filas y columnas.

Resumen de la Unidad 1

En esta primera unidad, los estudiantes se adentran en los cimientos del desarrollo web mediante el aprendizaje del Lenguaje de Marcado de Hipertexto, conocido comúnmente como HTML. Esta unidad es crucial, ya que HTML es la base sobre la cual se construyen todas las páginas web, proporcionando la estructura necesaria para organizar y presentar contenido en la web.

Comenzamos con una comprensión clara de qué es HTML y por qué es fundamental en el mundo del desarrollo web. Se introduce a los estudiantes al concepto de un "lenguaje de marcado", explicando cómo HTML utiliza etiquetas para definir los diferentes elementos de una página web. Esta sección enfatiza la importancia de HTML como el esqueleto de cualquier sitio web, sobre el cual se añadirán estilos y funcionalidades más adelante.

Los estudiantes aprenden a crear un documento HTML desde cero, ¡comenzando con la estructura básica que incluye las etiquetas esenciales como <!DOCTYPE html>, <html>, <head>,

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

<title>, y <body>. Se destaca cómo cada una de estas etiquetas juega un papel crucial en la organización del contenido y en cómo los navegadores interpretan y muestran las páginas web.

- **<!DOCTYPE html>**: Se explica como la declaración que informa al navegador que el documento sigue las especificaciones de HTML5, la versión más moderna y ampliamente soportada.
- **<html>**: Se presenta como la etiqueta raíz que contiene todo el contenido del documento.
- **<head>**: Se introduce como la sección donde se coloca la meta-información del documento, como el conjunto de caracteres (<meta charset="UTF-8">), el título de la página (<title>), y los enlaces a hojas de estilo o scripts.
- **<body>**: Se describe como el contenedor del contenido visible de la página, incluyendo texto, imágenes, videos y enlaces.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

UNIDAD 2 HOJAS DE ESTILOS EN CASCADA CSS

Temas y Subtemas

Introducción a CSS

- Importancia de CSS
- Sintaxis y estructura básica de las reglas CSS
- Selección de elementos y aplicaciones de estilos básicos

Modelos de caja en CSS

1. Estilos avanzados
2. Diseño de páginas utilizando propiedades de posicionamiento

Introducción a FlexBox y GridLayout

1. Creación de diseños flexibles y cuadricula.
2. Control de la laineación y el ordén de los elementos

Estilos Avanzados pseudo clases y pseudo-elements

1. Animaciones y transformaciones
2. Uso de fuentes personalizadas con CSS

Introducción al diseño web adaptable

1. Media Queries

INTRODUCCIÓN A CSS

CSS, o Hojas de Estilo en Cascada (Cascading Style Sheets), es una tecnología fundamental en el desarrollo web moderno. Mientras que HTML se encarga de la estructura y el contenido de una página web, CSS es responsable de la presentación visual. Es decir, CSS controla cómo se ve el contenido en la pantalla, permitiendo a los desarrolladores y diseñadores web aplicar estilos y diseñar páginas atractivas, coherentes y accesibles (Domínguez Mínguez, 2023).

Desde ajustar colores y fuentes hasta diseñar el layout (disposición) de una página, CSS ofrece un control completo sobre la apariencia de un sitio web. Esto significa que el mismo contenido HTML puede presentarse de diferentes maneras dependiendo de la hoja de estilo aplicada, lo que facilita el mantenimiento y mejora la experiencia del usuario.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 32. HTML y CSS

Nota: La imagen nos muestra cómo se integra CSS a HTML.

IMPORTANCIA DE CSS

CSS no solo es importante porque embellece las páginas web, sino porque también aporta organización, consistencia y eficiencia al desarrollo web. Sin CSS, los sitios web estarían limitados a la presentación básica de HTML, lo que resultaría en páginas poco atractivas y difíciles de navegar (Cucaro, 2022).

BENEFICIOS DEL USO DE CSS

1. **Separación de Contenido y Presentación**: Al usar CSS, el contenido (HTML) se separa de la presentación visual. Esto permite a los desarrolladores cambiar el diseño de un sitio

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

web sin tener que alterar el contenido, lo que facilita la actualización y el mantenimiento.

2. **Reutilización y Consistencia**: Con CSS, un conjunto de reglas puede aplicarse a múltiples páginas web, asegurando un diseño consistente en todo el sitio. Si se necesita un cambio, solo se modifica la hoja de estilo, y el cambio se refleja en todo el sitio.
3. **Accesibilidad**: CSS mejora la accesibilidad al permitir diseños responsivos, que se adaptan a diferentes tamaños de pantalla y dispositivos. Esto asegura que todos los usuarios, independientemente de su dispositivo, puedan acceder al contenido de manera óptima.
4. **Rendimiento**: Al mantener el diseño separado del contenido, CSS reduce el tamaño de los archivos HTML, lo que puede mejorar el rendimiento y la velocidad de carga del sitio.

SINTAXIS Y ESTRUCTURA BÁSICA DE LAS REGLAS CSS

Para utilizar CSS de manera efectiva, es esencial entender su sintaxis y cómo se estructuran las reglas de estilo. Una regla CSS se compone de un selector y una declaración. La declaración a su vez está formada por propiedades y valores.

COMPONENTES DE UNA REGLA CSS

- **Selector**: Indica a qué elementos HTML se aplicará la regla. Puede ser un elemento específico, un grupo de elementos, una clase, un ID, o incluso una combinación de estos.
- **Declaración**: Define cómo se debe estilizar el elemento seleccionado. La declaración se coloca entre llaves {} y está compuesta por una propiedad y un valor.
- **Propiedad**: Especifica el aspecto del elemento que se va a modificar, como el color, tamaño de fuente, márgenes, etc.
- **Valor**: Define cómo se va a aplicar la propiedad, por ejemplo, el color rojo (red), un tamaño de fuente de 16 píxeles (16px), o un margen de 10 píxeles (10px).

EJEMPLO DE REGLAS CSS

Para empezar con la practica vamos a integrar CSS dentro de nuestro archivo HTML, esta practica no es la más recomendada ya que si se lo hace de esta manera el archivo HTML se llena de información, que luego hace imposible leer el código con claridad. Para esto dentro de la etiqueta **<head>** integraremos los estilos que van dentro de la etiqueta **<style>** y ahí codificaremos todas las reglas de estilo para el documento **HTML.**

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <!--Creando estilos para HTML-->
    <style>
        h1{
            color: blue;
            font-size: 24px;
            text-align: center;
        }
    </style>
</head>
<body>
     <header >
        <h1>Añadiendo Estilos CSS</h1>
    </header>
    <section>
        <footer >
            <p>&copy; 2024 Todos los Derechos Reservados. Davitron </p>
        </footer>
    </section>
</body>
</html>
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Explicación del código

Selector: El selector serán todas las etiquetas HTML en las que se aplicarán estos estilos en el caso de nuestro ejemplo el estilo se aplicara a todo lo que este dentro de la etiqueta **h1**

Declaraciones: La declaraciones o argumentos serán todas las características que tomara esta etiqueta.

- **Color: blue;** estable el color de texto azul.
- **Font-size: 24px;** define el tamaño de la fuente del texto en 24px
- **Text-align: cebter;** centra el texto del contenedor.

Salida de Pantalla

Figura 33.
Integrando CSS

Nota: La imagen muestra implementando estilos a la etiqueta **h1**

SELECCIÓN DE ELEMENTOS Y APLICAION DE ESTILOS BASICOS

Una de las potentes características de CSS es la capacidad de seleccionar y aplicar estilos a elementos específicos de una página web. Esto se logra utilizando diferentes tipos de selectores, cada uno diseñado para apuntar a un conjunto específico de elementos.

TIPOS DE SELECTORES EN CSS

1. **Selectores de Elemento**: Aplican estilos a todos los elementos de un tipo específico.

Ejemplo:

```
h1{
        color: blue;
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
        font-size: 24px;
        text-align: center;
       }
```

Cambia el estilo de todas las etiquetas h1

2. **Selectores de Clase**: Se usan para aplicar estilos a elementos con una clase específica. Se definen con un punto (.) seguido del nombre de la clase. Esta clase se puede usar en cualquier parte de la aplicación.

Ejemplo:

```
<!--Creando estilos para HTML-->
    <style>
        .destacado{
            color: green;
        }
    </style>

<section>
        <!--Llamando a la clase-->
          <p class="destacado">En esta sección se aplica los estilos de la clase destacado</p>
    </section>
```

Aplica el color verde a la sección donde se este llamando a la clase

3. **Selectores ID.** Aplican estilos a un único elemento con un identificador específico. Se definen con una almohadilla **#** seguida del ID. Este tipo de selector se puede utilizar una sola vez.

```
<!--Creando estilos para HTML-->
    <style>
        #encabezado{
            color: red;
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
            text-align: center;
        }
    </style>
<header >
        <!--Llamando al identificador-->
        <h1 id="encabezado">Añadiendo Estilos CSS mediante un
</h1>
    </header>
```

Figura 34.
Selector #id

Nota: La imagen nos muestra la salida de pantalla aplicando mediante el selector id.

MODLEOS DE CAJAS CSS Y DISEÑO DE PAGINAS CON POSICIONAMIENTO

Introducción al Modelo de Caja CSS

El modelo de caja Box Model es un concepto fundamental en CSS que describe cómo se estructuran y se representan visualmente los elementos HTML en una página web. Según este modelo, cada elemento de una página se representa como una caja rectangular que consta de varios componentes: contenido, relleno padding, borde border y margen margin. Comprender cómo funciona este modelo es esencial para cualquier diseñador o desarrollador web, ya que influye directamente en el diseño, el espaciamiento y la disposición de los elementos en una página.

Componentes del modelo de Caja

Contenido (Content): Es el área donde se coloca el contenido real del elemento, como texto, imágenes o videos. El tamaño del contenido puede ajustarse con propiedades como width (ancho) y height (alto).

Relleno (Padding): Es el espacio entre el contenido y el borde del elemento. El relleno es transparente y se puede ajustar individualmente para cada lado (superior, inferior, izquierdo, derecho) utilizando propiedades como padding-top, padding-right, padding-bottom, y padding-left.

Borde (Border): Es el contorno que rodea el contenido y el relleno. Se puede personalizar en cuanto a grosor, estilo y color. Las propiedades más comunes para ajustar el borde incluyen border-width, border-style, y border-color.

Margen (Margin): Es el espacio exterior que separa un elemento de otros elementos que lo rodean. Al igual que el relleno, los márgenes se pueden ajustar de manera individual para cada lado.

Aplicando los conceptos

Crear un documento HTML donde se cree un contendor con los siguientes estilos ancho de 350px, una altura de 200px, un padding de 20px borde de 5px, solido y de color azul, y un margen de 15 px. Los estilos deben estar en un archivo externo al documento 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>
    <link rel="stylesheet" href="estilos.css">
</head>
<body>
     <header >
         <!--Llamando al identificador-->
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
            <h1 id="encabezado">Añadiendo Estilos CSS mediante un
</h1>
    </header>
    <section class="box">
         <p >En esta sección se aplica los estilos de la clase
destacado</p>
    </section>
    <section>
        <footer >
           <p>&copy; 2024 Todos los Derechos Reservados. Davitron
</p>
        </footer>
    </section>
</body>
</html>
```

Figura 35.
Cajas con CSS

Nota: La imagen nos muestra el posicionamiento de cajas y sus distintos argumentos.

ESTILOS AVANZADOS EN EL MODELO DE CAJA

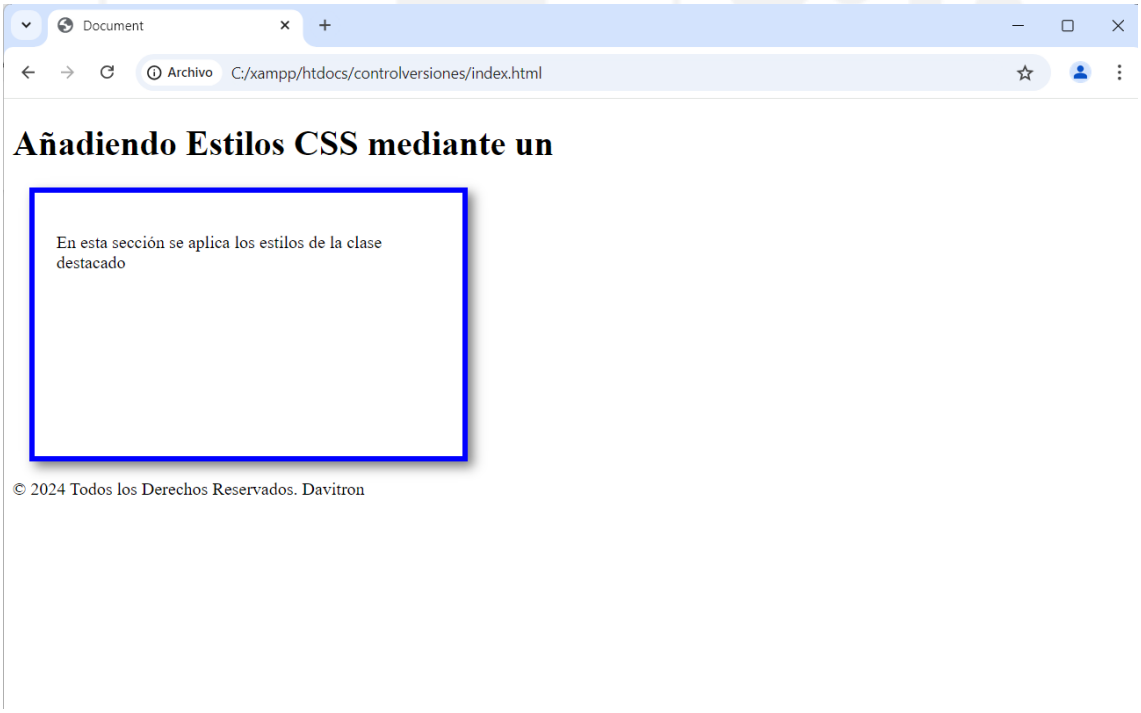
Una vez comprendido el concepto básico del modelo de caja, podemos explorar cómo CSS permite aplicar estilos avanzados para crear diseños más sofisticados. Aquí se cubren técnicas como sombras, bordes redondeados, y cómo ajustar el comportamiento del modelo de caja.

SOMBRA DE CAJA

La propiedad box-shadow en CSS permite agregar sombras a los elementos, lo que añade profundidad y ayuda a resaltar ciertos elementos en la página.

```
.box{
    width: 350px;
    height: 200px;
    padding: 20px;
    border: 5px solid blue;
    margin: 15px;
    box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.5);
}
```

Figura 36.
Sobra en el marco



Nota: La imagen no muestra una sombra en la caja

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

DISEÑO DE PAGINAS UTILIZANDO PROPIEDADES DE POSICIONAMIENTO

Además del modelo de caja, CSS proporciona varias propiedades de posicionamiento que permiten colocar los elementos de manera precisa en una página. Estas propiedades son cruciales para el diseño de layouts más complejos y responsivos.

Propiedades de Posicionamiento básico

- **position: static;** Es el valor por defecto. Los elementos se colocan según el flujo normal del documento.
- **position: relative;** Posiciona un elemento relativo a su posición original. Se pueden utilizar propiedades como top, right, bottom, y left para ajustar su ubicación.

```
.relative-box{
    position: relative;
    border: 5px solid blue;
    top: 20px;
    left: 10px;
}
```

Figura 37.
Posición Relativa

Nota: La imagen nos muestra al contenedor con una posición relativa.

- **position: absolute;** Posiciona un elemento en relación a su contenedor más cercano que no tenga position: static;. Se elimina del flujo normal del documento.

- **position: fixed;** Fija un elemento en la ventana del navegador, lo que significa que permanece visible incluso cuando el usuario hace scroll.
- **position: sticky;**: Combina el comportamiento de relative y fixed. Un elemento con position: sticky se comporta como relative hasta que se alcanza un cierto punto en la página, momento en el que se fija en su lugar.

DISEÑO CON FLEXBOX Y GRID

Para diseños de página más avanzados y responsivos, CSS ofrece las tecnologías Flexbox y Grid, que permiten una mayor flexibilidad y control sobre la disposición de los elementos.

- **Flexbox**: Es ideal para diseños unidimensionales, donde se requiere alinear elementos a lo largo de una fila o columna. Con display: flex;, los elementos hijos dentro de .flex-container se alinearán automáticamente y se distribuirán equitativamente.
- **Grid**: Es más adecuado para diseños bidimensionales, donde se necesita controlar tanto filas como columnas.

MEDIA QUERIES EN CSS: ADAPTANDO EL DISEÑO A TODOS LOS DISPOSITIVOS

¿Qué Son las Media Queries?

Las **medias queries** son una poderosa herramienta en CSS que permite adaptar el diseño de una página web a diferentes dispositivos y tamaños de pantalla. En esencia, una media query es una condición que se utiliza para aplicar estilos CSS solo cuando se cumplen ciertos criterios, como el ancho de la pantalla, la resolución o la orientación del dispositivo (horizontal o vertical).

En un mundo donde los usuarios acceden a la web desde una variedad de dispositivos—teléfonos móviles, tabletas, laptops, y monitores grandes—, es crucial que los sitios web sean responsivos. Esto significa que deben ajustarse automáticamente a cualquier tamaño de pantalla para ofrecer una experiencia de usuario óptima. Aquí es donde entran en juego las media queries.

¿Por Qué Son Importantes las Media Queries?

Las medias queries permiten a los desarrolladores web crear sitios que funcionen y se vean bien en cualquier dispositivo. Sin ellas, un diseño que se vea bien en una pantalla grande podría no ser legible o funcional en una pantalla más pequeña, como la de un teléfono móvil.

Ventajas Clave de Usar Media Queries:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- **Responsividad**: Hacen que el sitio web sea responsivo, adaptándose a diferentes tamaños de pantalla y resoluciones.
- **Mejora de la Experiencia del Usuario**: Garantizan que los usuarios tengan una experiencia óptima sin importar el dispositivo que utilicen.
- **Mantenimiento Eficiente**: Permiten mantener un solo archivo CSS que cubra múltiples escenarios, en lugar de crear versiones separadas del sitio para diferentes dispositivos.

ESTRUCTURA BÁSICA DE UNA MEDIA QUERY

Las media queries siguen una estructura simple pero efectiva. Se pueden incluir directamente en la hoja de estilo CSS o dentro de una etiqueta <style> en el documento HTML.

Sintaxis

```
@media (condición) {
 /* Reglas CSS que se aplican cuando se cumple la condición */
}
```

La **condición** dentro de la media query especifica los criterios que deben cumplirse para que las reglas CSS dentro de ella se apliquen.

Ejemplo Básico

```
@media (max-width: 600px) {
    body {
        background-color: lightblue;
    }
}
```

En este ejemplo, la regla CSS dentro de la media query solo se aplicará si el ancho de la pantalla del dispositivo es de 600 píxeles o menos. Si se cumple esta condición, el color de fondo del cuerpo body cambiará a azul claro.

TIPOS COMUNES DE MEDIA QUERIES

Las media queries pueden basarse en varios tipos de condiciones que reflejan las características del dispositivo. A continuación, se describen algunas de las más comunes:

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Ancho y alto de la pantalla (width y height)

Estas condiciones se utilizan para adaptar el diseño según el ancho o el alto de la pantalla del dispositivo.

max-width: Aplica las reglas cuando el ancho de la pantalla es menor o igual al valor especificado.

```
@media (max-width: 768px) {
    .menu {
        display: none;
    }
}
```

En este ejemplo, el menú se ocultará cuando el ancho de la pantalla sea de 768 píxeles o menos.

min-width: Aplica las reglas cuando el ancho de la pantalla es mayor o igual al valor especificado.

```
@media (min-width: 1024px) {
    .sidebar {
        width: 300px;
    }
}
```

Aquí, la barra lateral tendrá un ancho de 300 píxeles en pantallas de 1024 píxeles de ancho o más.

Orientación del Dispositivo (Orientation)

Esta condición se utiliza para detectar si la pantalla está en modo horizontal o vertical retrato.

orientation: landscape: Aplica las reglas cuando el dispositivo está en modo horizontal.

```
@media (orientation: landscape) {
    body {
        font-size: 18px;
    }
}
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

En este ejemplo, el tamaño de la fuente aumentará cuando el dispositivo esté en modo horizontal.

orientation: portrait: Aplica las reglas cuando el dispositivo está en modo vertical.

```
@media (orientation: portrait) {
    body {
        font-size: 14px;
    }
}
```

Aquí, el tamaño de la fuente se reducirá cuando el dispositivo esté en modo vertical.

Resolución de pantalla

Se utiliza para adaptar el diseño a diferentes resoluciones de pantalla, midiendo la densidad de píxeles.

min-resolution y **max-resolution**: Estas condiciones se usan para aplicar estilos en dispositivos con resoluciones específicas, como pantallas de alta densidad (por ejemplo, Retina Display).

```
@media (min-resolution: 2dppx) {
    img {
        width: 50%;
    }
}
```

Este ejemplo ajusta el tamaño de las imágenes en dispositivos con una densidad de píxeles de 2dppx (puntos por píxel) o más.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Autoevaluación 2

1. ¿Qué es CSS y cuál es su propósito principal en el desarrollo web?

   CSS (Cascading Style Sheets) es un lenguaje de estilos que se utiliza para describir la presentación visual de un documento HTML. Su propósito principal es separar la estructura y el contenido (HTML) del diseño y la apariencia (CSS), permitiendo a los desarrolladores controlar cómo se ve y se comporta una página web en diferentes dispositivos y pantallas.

2. ¿Cómo se estructura una regla CSS?

   Una regla CSS se estructura en dos partes principales: el selector y la declaración. El **selector** identifica qué elementos HTML se van a estilizar, y la **declaración** define cómo se deben estilizar, utilizando pares de propiedad y valor dentro de llaves {}. Por ejemplo:

```
h1 {
    color: blue;
    font-size: 24px;
}
```

3. ¿Qué es un selector en CSS y cuáles son los tipos más comunes?

   Un selector en CSS es una parte de la regla que especifica qué elementos HTML se van a estilizar. Los tipos más comunes de selectores incluyen:

   - **Selectores de elementos:** Aplica estilos a todos los elementos de un tipo específico (por ejemplo, p para párrafos).
   - **Selectores de clase:** Aplica estilos a elementos con una clase específica, utilizando un punto (.) antes del nombre de la clase (por ejemplo, .destacado).
   - **Selectores de ID:** Aplica estilos a un elemento único con un ID específico, utilizando una almohadilla (#) antes del nombre del ID (por ejemplo, #encabezado).

4. ¿Cuál es la diferencia entre un selector de clase y un selector de ID en CSS?

   Un selector de **clase** (.nombre-de-clase) se utiliza para aplicar estilos a múltiples elementos que comparten la misma clase, mientras que un selector de **ID** (#nombre-de-id) se aplica a un único elemento que tiene un ID específico. En HTML, un ID debe ser

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

único dentro de la página, mientras que una clase puede ser utilizada por varios elementos.

5. ¿Cómo se puede aplicar un color de fondo a un elemento utilizando CSS?

   Se puede aplicar un color de fondo a un elemento utilizando la propiedad background-color en la declaración CSS

```
div {
    background-color: lightblue;
}
```

6. ¿Qué significa la propiedad margin en CSS y cómo se utiliza?

   La propiedad margin en CSS define el espacio exterior que rodea un elemento, separándolo de otros elementos adyacentes. Se puede establecer un margen individual para cada lado (superior, derecho, inferior, izquierdo) o un margen uniforme para todos los lados.

```
p {
    margin: 20px;
}
```

7. ¿Qué es la propiedad padding en CSS y en qué se diferencia de margin?

   La propiedad padding en CSS define el espacio interior entre el contenido de un elemento y su borde. A diferencia de margin, que crea espacio fuera del borde del elemento, padding agrega espacio dentro del borde del elemento.

```
.box {
    padding: 10px;
}
```

8. ¿Cómo se puede centrar un texto dentro de un elemento usando CSS?

   Se puede centrar un texto dentro de un elemento utilizando la propiedad `text-align`.

```
h2 {
    text-align: center;
}
```

9. ¿Qué función cumple la propiedad border en CSS?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

La propiedad border en CSS se utiliza para definir el borde alrededor de un elemento. Se pueden especificar el grosor, el estilo y el color del borde.

```
img {
    border: 2px solid black;
}
```

10. ¿Cómo se puede cambiar el tamaño del texto en CSS?

    El tamaño del texto en CSS se puede cambiar utilizando la propiedad font-size. Esta propiedad acepta valores en unidades como píxeles (px), em, rem, o porcentajes.

```
p {
    font-size: 16px;
}
```

Resumen de la Unidad 2

En esta segunda unidad, exploramos cómo el uso de CSS (Cascading Style Sheets) revoluciona la forma en que los desarrolladores diseñan y presentan páginas web. Mientras que HTML se encarga de la estructura básica y el contenido de la página, CSS permite darle vida a esa estructura, aplicando estilos visuales que hacen que las páginas sean más atractivas, coherentes y usables.

CSS juega un papel crucial al separar la presentación del contenido, lo que facilita no solo la creación de diseños flexibles y responsivos, sino también el mantenimiento y actualización del sitio web a lo largo del tiempo. A través de esta unidad, los estudiantes aprenderán a dominar las herramientas necesarias para transformar el aspecto de cualquier página web, adaptándola a diferentes dispositivos y necesidades de los usuarios.

Una de las primeras habilidades que los estudiantes deben adquirir al trabajar con CSS es comprender su sintaxis y cómo estructurar correctamente una regla de estilo.

Cada regla CSS se compone de dos partes fundamentales:

- **El Selector:** Indica qué elementos HTML serán afectados por la regla. Pueden ser etiquetas específicas como <h1>, clases (.mi-clase), o IDs (#mi-id). El selector es el punto de inicio para aplicar estilos, y entender cómo elegir el selector correcto es clave para un diseño eficiente.
- **La Declaración:** Dentro de llaves {}, las declaraciones definen cómo se debe aplicar el estilo al elemento seleccionado. Una declaración se compone de propiedades y valores,

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

como color: blue; que establece el color de un texto, o margin: 10px; que define un margen alrededor de un elemento.

Comprender esta estructura es esencial para crear estilos que se apliquen de manera efectiva y precisa a los elementos de una página web.

Esta unidad también ofrece una inmersión profunda en el uso de CSS junto con HTML, proporcionando a los estudiantes las habilidades necesarias para transformar una página web básica en una experiencia visualmente atractiva y funcional. Al comprender y aplicar correctamente los selectores, propiedades y técnicas avanzadas de CSS, los estudiantes estarán bien preparados para diseñar y desarrollar sitios web que no solo se vean bien, sino que también proporcionen una excelente experiencia de usuario en cualquier dispositivo.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

UNIDAD 3 PROGRAMACIÓN INTERACTIVA CON JavaScript

Temas y Subtemas

Introducción a la programación de JavaScript
- Sintaxis básica
- Tipos de datos
- Operadores
- Uso de consola del navegador

Condicionales
1. if, else
2. bucles
3. Funciones

Arreglos
1. Creación y manipulación de objetos JS
2. Iteración a través de arreglos y objetos

Exploración de eventos clave en JavaScript
1. click
2. teclado
3. formulario

INTRODUCCIÓN A JAVASCRIPT.

JavaScript es un lenguaje de programación que permite a los desarrolladores agregar interactividad y dinamismo a las páginas web. A diferencia de HTML y CSS, que se encargan de la estructura y el diseño, JavaScript da vida a los sitios web, permitiendo crear desde animaciones sencillas hasta aplicaciones web completas. Es un lenguaje que se ejecuta del lado del cliente, lo que significa que se ejecuta directamente en el navegador del usuario, sin necesidad de contactar con el servidor (Pérez Rodríguez, 2012).

SINTAXIS BÁSICA DE JavaScript

La sintaxis de JavaScript es la forma en que se escribe y organiza el código. Como cualquier lenguaje de programación, JavaScript tiene un conjunto de reglas y convenciones que deben seguirse para que el navegador pueda interpretar y ejecutar el código correctamente.

INICIANDO CON UN SCRIPT

En una página HTML, (Gauchat, 2012) JavaScript se puede incluir de dos maneras principales: directamente dentro de la etiqueta <script> en el archivo HTML, o en un archivo JavaScript

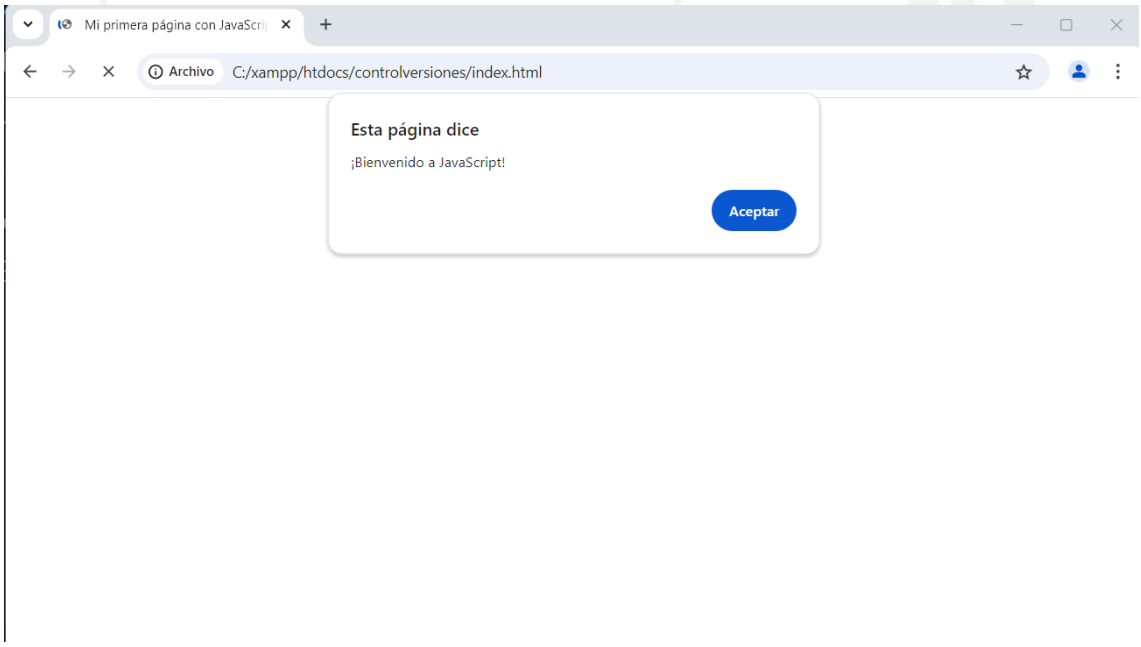
(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

externo que se enlaza al HTML. Aquí un ejemplo básico de cómo se integra JavaScript en una página web:

Html.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Mi primera página con JavaScript</title>
</head>
<body>
    <h1>¡Hola, Mundo!</h1>
    <script>
        alert('¡Bienvenido a JavaScript!');
    </script>
</body>
</html>
```

Figura 38.
Insertando JS



Nota: La imagen muestra la implementación de una alerta de JavaScript en el documento HTML.

Comentarios en JavaScript

Los comentarios son esenciales para explicar el código y hacerlo más fácil de entender. JavaScript permite dos tipos de comentarios:

Comentarios de una sola línea: Se inician con // y el resto de la línea es ignorado por el navegador.

```
// Este es un comentario de una sola línea

console.log('Hola, Mundo');
```

Comentarios de múltiples líneas: Comienzan con /* y terminan con */, y se pueden utilizar para comentarios más largos o para desactivar temporalmente partes del código.

```
/*
Este es un comentario de múltiples líneas.
Puedes escribir varias líneas aquí.
*/
console.log('JavaScript es genial');
```

TIPOS DE VARIABLES EN JavaScript

En JavaScript, las variables son contenedores que almacenan datos. Podemos pensar en una variable como una caja que guarda información que podemos utilizar y manipular más tarde en nuestro código (Luna, 2019).

Declaración de variables

Hay tres maneras de declarar variables en JavaScript:

var: Es la forma más antigua de declarar variables. Su alcance puede ser global o local, dependiendo de dónde se declare.

```
var nombre = 'Juan';
```

let: Introducido en ECMAScript 6 (ES6), let es la forma más moderna y recomendable para declarar variables. Tiene un alcance de bloque, lo que significa que solo existe dentro del bloque de código donde se define.

```
let edad = 25;
```

const: También introducido en ES6, const se utiliza para declarar constantes, es decir, valores que no cambiarán a lo largo de la ejecución del programa. Como let, tiene un alcance de bloque.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
const PI = 3.1416;
```

TIPOS DE DATOS EN JavaScript

JavaScript trabaja con diferentes tipos de datos, y es fundamental entenderlos para utilizar variables de manera efectiva (Pérez, 2019).

Tipos de datos primitivos

Números (number): Representan valores numéricos, incluyendo enteros y decimales.

```
let edad = 30;
```

```
let temperatura = 36.5;
```

Cadenas de Texto (string): Se utilizan para representar texto. Las cadenas se definen usando comillas simples (') o dobles (").

```
let nombre = 'Ana';
```

```
let saludo = "¡Hola, Mundo!";
```

Booleanos (boolean): Solo tienen dos valores posibles: true (verdadero) o false (falso). Se usan para representar valores lógicos.

```
let esMayorDeEdad = true;
```

```
let tieneLicencia = false;
```

Null: Representa la ausencia intencionada de un valor, es decir, un valor vacío.

```
let resultado = null;
```

Undefined: Significa que una variable ha sido declarada pero no se le ha asignado ningún valor.

```
let valorIndefinido;
```

```
console.log(valorIndefinido); // Esto mostrará "undefined"
```

Tipos de Datos Complejos

Objetos (object): Son estructuras más complejas que pueden contener múltiples valores de diferentes tipos, organizados en pares clave-valor.

```
let persona = {
    nombre: 'Carlos',
    edad: 28,
    esEstudiante: true
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
};
```

Arreglos (array): Son listas ordenadas de valores, que pueden ser de cualquier tipo de dato.

```
let colores = ['rojo', 'verde', 'azul'];
```

OPERADORES EN JavaScript

Los operadores en JavaScript permiten realizar operaciones con variables y valores. Son fundamentales para construir expresiones y tomar decisiones en el código.

Operadores aritméticos

Estos operadores realizan operaciones matemáticas básicas:

Suma (+): Suma dos números o concatena cadenas.

```
let suma = 5 + 3; // Resultado: 8
let concatenacion = 'Hola, ' + 'Mundo'; // Resultado: "Hola, Mundo"
```

Resta (-): Resta un número de otro.

```
let resta = 10 - 6; // Resultado: 4
```

Multiplicación (*): Multiplica dos números.

```
let multiplicacion = 4 * 7; // Resultado: 28
```

División (/): Divide un número entre otro.

```
let division = 20 / 5; // Resultado: 4
```

Módulo (%) : Devuelve el residuo de una división.

```
let modulo = 10 % 3; // Resultado: 1
```

Operadores de Asignación

Asignan un valor a una variable. El operador más común es el de igual (=), pero hay otros combinados con operadores aritméticos:

Asignación simple (=):

```
let x = 10;
```

Asignación con suma (+=):

```
let x = 5;
x += 3; // x ahora es 8
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Asignación con resta (-=):

```
let x = 5;

x -= 2; // x ahora es 3
```

Operadores de comparación

Se utilizan para comparar valores y siempre devuelven un valor booleano (true o false).

Igualdad (==): Compara dos valores sin tener en cuenta su tipo.

```
let comparacion = 5 == '5'; // Resultado: true
```

Igualdad estricta (===): Compara dos valores y su tipo.

```
let comparacion = 5 === '5'; // Resultado: false
```

Diferencia (!=): Compara si dos valores son diferentes.

```
let diferencia = 5 != '5'; // Resultado: false
```

Diferencia estricta (!==): Compara si dos valores y su tipo son diferentes.

```
let diferencia = 5 !== '5'; // Resultado: true
```

USO DE LA CONSOLA EN EL NAVEGADOR

La consola del navegador es una herramienta invaluable para los desarrolladores, ya que permite depurar y probar código JavaScript directamente en el navegador. Se puede acceder a la consola en la mayoría de los navegadores modernos, como Chrome y Firefox, presionando F12 o haciendo clic derecho en la página y seleccionando "Inspeccionar".

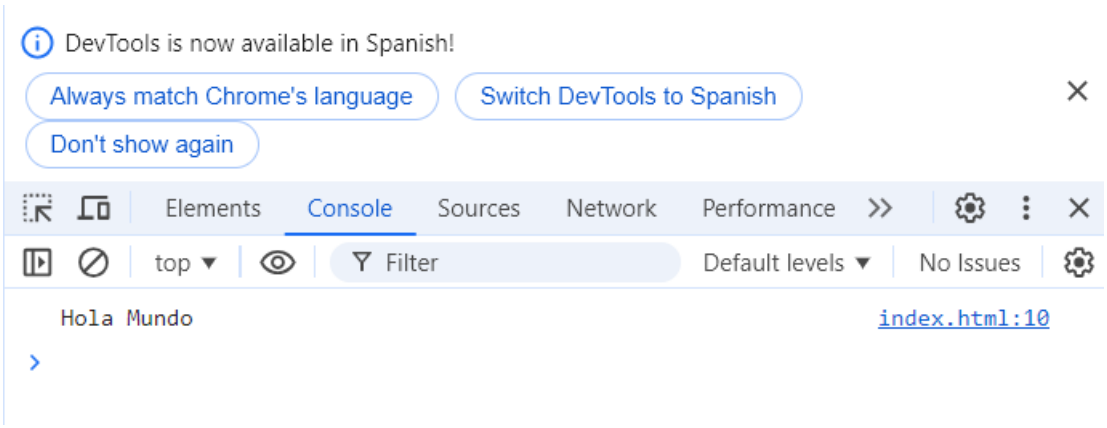
Métodos de la consola

console.log(): El método más común, se usa para imprimir mensajes o valores en la consola.

```
console.log('Hola, Mundo');
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 39.
Ejecución JS

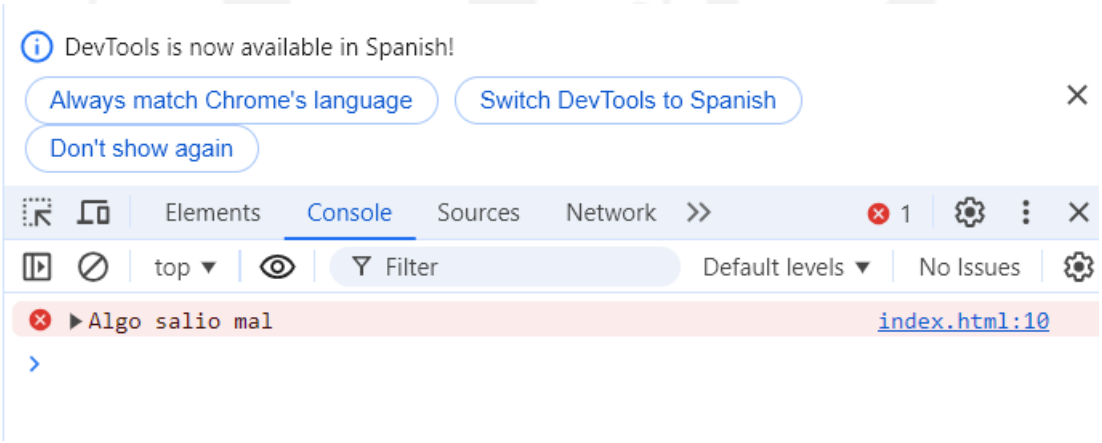


Nota: En consola nos muestra el mensaje de JS.

console.error(): Se utiliza para mostrar mensajes de error en la consola.

```
console.error('Algo salió mal');
```

Figura 40.
Mensaje de error



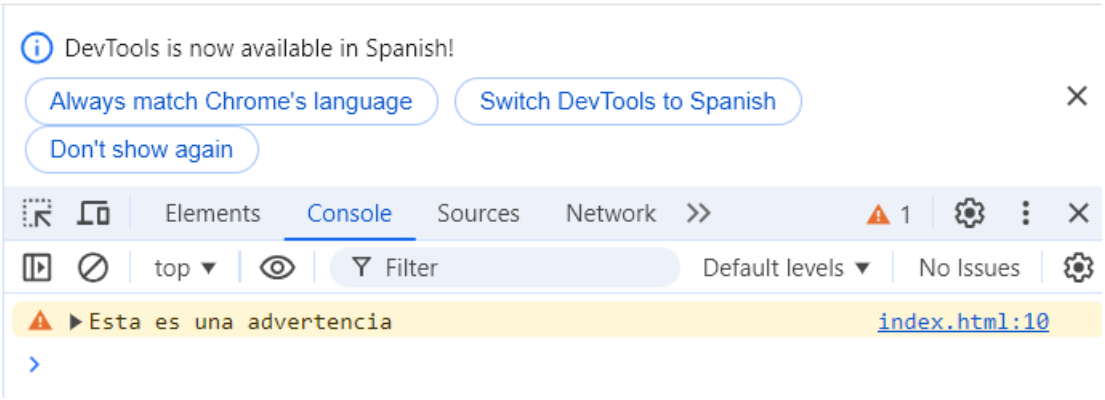
Nota: La imagen muestra un mensaje de error en consola

console.warn(): Muestra advertencias en la consola.

```
console.warn('Esta es una advertencia'); // Imprime una advertencia
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Figura 41.
Mensaje de advertencia

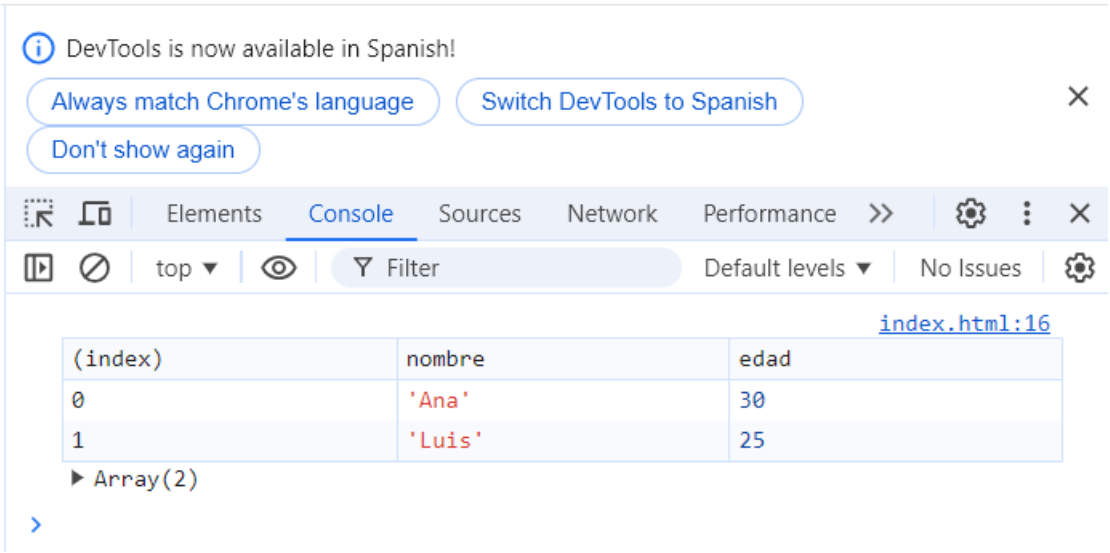


Nota: La imagen nos muestra un mensaje de advertencia codificado en JS

console.table(): Muestra datos en formato de tabla, lo que es útil para visualizar objetos y arrays complejos.

```
let personas = [
    { nombre: 'Ana', edad: 30 },
    { nombre: 'Luis', edad: 25 }
];
console.table(personas);
```

Figura 42.
Muestra el contenido de un arreglo



Nota: La imagen nos muestra en consola la ejecución de JS mostrando una tabla.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

CONDICIONALES EN JavaScript

Las estructuras condicionales en JavaScript son una de las herramientas más poderosas y esenciales en la programación. Estas permiten que el código tome decisiones basadas en ciertas condiciones. Dicho de otra manera, las condicionales le dicen a tu programa qué hacer cuando se cumple o no una condición específica.

Por ejemplo, si estás creando una aplicación que debe verificar si un usuario ha iniciado sesión, puedes usar una condicional para ejecutar una acción específica si el usuario está autenticado, y otra si no lo está. Esto hace que el programa sea más dinámico y capaz de responder a diferentes situaciones en tiempo real.

La Estructura if...else

La estructura condicional más básica y utilizada en JavaScript es if...else. Esta estructura evalúa una condición; si la condición se cumple (es true), ejecuta un bloque de código. Si la condición no se cumple (es false), puede ejecutar un bloque de código alternativo definido en else.

Sintaxis básica

```
if (condicion) {
    // Código a ejecutar si la condición es verdadera
}
```

- **if:** Palabra clave que indica el inicio de la estructura condicional.
- **condición**: Una expresión que es evaluada como verdadera (true) o falsa (false).
- **Bloque de código**: Es el conjunto de instrucciones que se ejecutarán si la condición es verdadera. Se coloca entre llaves {}.

Ejemplo condicional if

```
let edad = 18;


if (edad >= 18) {
    console.log('Eres mayor de edad.');
}
```

En este ejemplo, la condición edad >= 18 verifica si la variable edad es mayor o igual a 18. Si es así, se imprime el mensaje "Eres mayor de edad." en la consola.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Sintaxis if else

```
if (condicion) {
    // Código a ejecutar si la condición es verdadera
} else {
    // Código a ejecutar si la condición es falsa
}
```

else: Es la parte opcional que se ejecuta si la condición del if no se cumple.

Ejemplo

```
let edad = 16;


if (edad >= 18) {
    console.log('Eres mayor de edad.');
} else {
    console.log('Eres menor de edad.');
}
```

Aquí, si la condición edad >= 18 no se cumple, se ejecutará el bloque de código dentro de else, mostrando "Eres menor de edad."

La estructura if else if else

A veces, necesitas evaluar múltiples condiciones en secuencia. Para esto, puedes usar else if, que te permite verificar una segunda condición si la primera no se cumple.

Sintaxis if else is else

```
if (condicion1) {
    // Código a ejecutar si la condicion1 es verdadera
} else if (condicion2) {
    // Código a ejecutar si la condicion1 es falsa y la
condicion2 es verdadera
} else {
    // Código a ejecutar si ninguna de las condiciones
anteriores es verdadera
}
```

Ejemplo

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
let nota = 85;


if (nota >= 90) {
    console.log('Tienes una A');
} else if (nota >= 80) {
    console.log('Tienes una B');
} else if (nota >= 70) {
    console.log('Tienes una C');
} else {
    console.log('Necesitas mejorar.');
}
```

En este ejemplo, la variable nota se compara con varias condiciones. Dependiendo de su valor, se imprimirá un mensaje diferente. Si nota es 85, se imprimirá "Tienes una B".

CONDICIONALES ANIDADAS

Las condiciones anidadas son estructuras if...else dentro de otras estructuras if...else. Son útiles cuando necesitas evaluar una condición solo después de que otra se haya cumplido.

Ejemplo de condicionales anidadas

```
let usuario = 'admin';
let contrasena = '1234';


if (usuario === 'admin') {
    if (contrasena === '1234') {
        console.log('Bienvenido, administrador.');
    } else {
        console.log('Contraseña incorrecta.');
    }
} else {
    console.log('Usuario no reconocido.');
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
}
```

Aquí, primero se verifica si el usuario es "admin". Si es verdadero, entonces se verifica la contrasena. Si ambas condiciones se cumplen, se le da la bienvenida al administrador. Si alguna de las condiciones falla, se muestra un mensaje de error adecuado.

La Estructura switch

Cuando se tiene una variable que puede tomar múltiples valores posibles, usar switch puede ser más limpio y fácil de leer que múltiples if...else if. El switch evalúa el valor de una expresión y ejecuta el código correspondiente al caso que coincida.

Sintaxis del switch

```
switch (expresion) {
    case valor1:
        // Código a ejecutar si expresion === valor1
        break;
    case valor2:
        // Código a ejecutar si expresion === valor2
        break;
    default:
        // Código a ejecutar si ninguno de los casos coincide
}
```

- **expresion**: La variable o expresión que se evalúa.
- **case**: Define un bloque de código que se ejecutará si el valor de expresión coincide con valor1, valor2, etc.
- **break**: Detiene la ejecución dentro del switch y evita que se ejecuten otros casos innecesariamente.
- **default**: Es opcional y se ejecuta si ninguno de los casos coincide.

Ejemplo de switch

```
let dia = 3;

let diaSemana;
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
switch (dia) {
    case 1:
        diaSemana = 'Lunes';
        break;
    case 2:
        diaSemana = 'Martes';
        break;
    case 3:
        diaSemana = 'Miércoles';
        break;
    case 4:
        diaSemana = 'Jueves';
        break;
    case 5:
        diaSemana = 'Viernes';
        break;
    case 6:
        diaSemana = 'Sábado';
        break;
    case 7:
        diaSemana = 'Domingo';
        break;
    default:
        diaSemana = 'Día no válido';
}

console.log(`Hoy es ${diaSemana}.`);
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

En este ejemplo, día es 3, por lo que el switch asigna "Miércoles" a la variable diaSemana. Luego, se imprime "Hoy es Miércoles." en la consola.

OPERADORES LÓGICOS EN CONDICIONALES

Para construir condiciones más complejas, puedes combinar varias condiciones usando operadores lógicos como && (AND), || (OR), y ! (NOT).

Operador AND (&&)

Este operador devuelve true solo si ambas condiciones son verdaderas.

```
let edad = 20;
let tieneLicencia = true;

if (edad >= 18 && tieneLicencia) {
    console.log('Puedes conducir.');
} else {
    console.log('No puedes conducir.');
}
```

Aquí, solo se permite conducir si la persona tiene 18 años o más y tiene licencia.

Operador OR (||)

Este operador devuelve true si al menos una de las condiciones es verdadera.

```
let tieneEntrada = true;
let esInvitado = false;

if (tieneEntrada || esInvitado) {
    console.log('Puedes entrar a la fiesta.');
} else {
    console.log('No puedes entrar.');
}
```

En este ejemplo, se permite la entrada si la persona tiene una entrada o si es un invitado, es decir, si al menos una de las condiciones es verdadera.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Operador NOT (!)

Este operador invierte el valor lógico de una condición.

```
let esEstudiante = false;


if (!esEstudiante) {
    console.log('Necesitas registrarte como estudiante.');
}
```

Aquí, !esEstudiante es true porque esEstudiante es false, por lo que se muestra el mensaje de registro.

BUCLES Y FUNCIONES EN JavaScript

En programación, los **bucles** son estructuras que permiten ejecutar un bloque de código repetidamente, lo cual es útil cuando necesitamos realizar una acción varias veces sin tener que escribir el mismo código repetidamente. Los bucles son fundamentales para automatizar tareas repetitivas y manejar grandes volúmenes de datos de manera eficiente.

JavaScript ofrece varios tipos de bucles, cada uno diseñado para situaciones específicas. Los más comunes son for, while, y do...while.

BUCLE FOR

El bucle for es probablemente el más utilizado en JavaScript. Se utiliza cuando se conoce de antemano el número de veces que se desea repetir un bloque de código. La estructura básica del bucle for incluye una declaración inicial, una condición para continuar, y una expresión que se ejecuta al final de cada iteración.

Sintaxis básica de for

```
for (inicializacion; condicion; incremento) {
    // Código a ejecutar en cada iteración
}
```

- **Inicialización**: Es la declaración de la variable de control del bucle y se ejecuta una vez al comienzo.
- **Condición**: Es una expresión que se evalúa antes de cada iteración. Si es true, el bucle continúa; si es false, el bucle termina.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

- **Incremento**: Es una expresión que se ejecuta al final de cada iteración, generalmente para actualizar la variable de control.

Ejercicio ciclo for

```
for (let i = 0; i < 5; i++) {
    console.log('Iteración número ' + i);
}
```

En este ejemplo, el bucle comienza con i = 0, y se repite mientras i sea menor que 5. Después de cada iteración, i aumenta en 1. El bucle imprimirá "Iteración número 0", "Iteración número 1", y así hasta "Iteración número 4".

BUCLE WHILE

El bucle while se utiliza cuando no se sabe de antemano cuántas veces se debe repetir el bloque de código. El bucle continúa ejecutándose mientras la condición especificada sea true.

Sintaxis básica ciclo while

```
while (condicion) {
    // Código a ejecutar mientras la condición sea verdadera
}
```

Ejemplo ciclo while

```
let contador = 0;

while (contador < 5) {
    console.log('Contador: ' + contador);
    contador++;
}
```

En este caso, el bucle while continuará ejecutándose mientras contador sea menor que 5. Dentro del bucle, contador se incrementa en 1 en cada iteración. El resultado será similar al bucle for anterior.

Bucle do while

El bucle do...while es similar al while, pero con una diferencia clave: el bloque de código se ejecuta al menos una vez, ya que la condición se evalúa después de la primera ejecución.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Sintaxis básica de do while

```
do {
    // Código a ejecutar al menos una vez
} while (condicion);
```

Ejemplo de do while

```
let contador = 0;


do {
    console.log('Contador: ' + contador);
    contador++;
} while (contador < 5);
```

Aquí, el bucle se ejecutará de manera similar al while, pero incluso si la condición fuera false desde el principio, el código dentro del bloque do se ejecutaría una vez.

INTRODUCCIÓN A FUNCIONES

Las **funciones** en JavaScript son bloques de código reutilizables que realizan una tarea específica. Las funciones son fundamentales porque permiten dividir un programa en piezas manejables, reduciendo la repetición de código y facilitando el mantenimiento y la comprensión del programa (Rubiales Gómez, Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021, 2021).

Hay diferentes maneras de declarar una función en JavaScript, pero la más común es usando la palabra clave function seguida del nombre de la función, paréntesis (), y un bloque de código entre llaves {}.

Sintaxis básica de una función

```
function nombreDeLaFuncion() {
    // Código a ejecutar cuando se llama a la función
}
```

Ejemplo

```
function saludar() {
    console.log('¡Hola, Mundo!');
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

```
}
```

Aquí, se declara una función llamada saludar que, cuando se llama, imprime "¡Hola, Mundo!" en la consola.

Después de declarar una función, se puede "llamar" o "invocar" en cualquier parte del código para ejecutar el bloque de código que contiene.

```
saludar(); // Esto imprimirá "¡Hola, Mundo!" en la consola
```

Parámetros y argumentos en funciones

Las funciones pueden aceptar **parámetros**, que son variables que se utilizan para pasar información a la función. Los valores reales que se pasan cuando se llama a la función se llaman argumentos (Meloni, 2018).

Sintaxis de una función con parámetros.

```
function sumar(a, b) {
    console.log(a + b);
}
```

En este ejemplo, a y b son parámetros que se usan dentro de la función para sumar dos números.

Llamada a una función con argumentos:

```
sumar(5, 10); // Esto imprimirá 15 en la consola
```

Aquí, 5 y 10 son los argumentos que se pasan a la función sumar.

RETORNO DE VALORES DESDE UNA FUNCIÓN

Las funciones en JavaScript pueden devolver un valor usando la palabra clave return. Esto es útil cuando se necesita que la función realice un cálculo o procese datos y luego proporcione un resultado que se pueda utilizar en otra parte del programa (Lancker, 2013).

Ejemplo

```
function multiplicar(a, b) {
    return a * b;
}

let resultado = multiplicar(3, 4);
console.log(resultado); // Esto imprimirá 12
```


(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

En este caso, la función multiplicar devuelve el producto de a y b. Luego, este valor se almacena en la variable resultado, que se imprime en la consola.

VECTORES Y MATRICES EN JavaScript

En JavaScript, los **vectores** (también conocidos como arrays) y las **matrices** (arrays multidimensionales) son estructuras de datos que permiten almacenar y gestionar conjuntos de datos de forma ordenada. Estas estructuras son fundamentales para manejar colecciones de elementos, como listas de números, cadenas de texto, objetos, e incluso otras arrays (Martin, 2017).

VECTORES

Un **vector** o **array** en JavaScript es una lista ordenada de elementos, donde cada elemento puede ser de cualquier tipo de dato: números, cadenas, objetos, o incluso otros arrays. Los elementos de un array están indexados, lo que significa que cada uno tiene una posición específica dentro de la lista, comenzando desde el índice 0 (Aubry, 2017).

Crear un Vector

Hay varias maneras de crear un array en JavaScript, pero la forma más común es utilizando corchetes [].

Ejemplo

```
let numeros = [10, 20, 30, 40, 50];
```

En este ejemplo, numeros es un array que contiene cinco elementos: 10, 20, 30, 40 y 50.

Acceso a los elementos de un Vector

Para acceder a un elemento específico de un array, se utiliza el índice correspondiente entre corchetes [].

```
let primerNumero = numeros[0]; // Accede al primer elemento: 10
let tercerNumero = numeros[2]; // Accede al tercer elemento: 30
```

Modificación de Elementos en un Vector

Puedes modificar los elementos de un array asignando un nuevo valor a una posición específica.

```
numeros[1] = 25; // Cambia el segundo elemento a 25
```

Después de esta modificación, el array numeros se verá así: [10, 25, 30, 40, 50].

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

MATRICES

Una **matriz** es un array que contiene otros arrays como sus elementos. En JavaScript, las matrices se utilizan para representar tablas de datos o estructuras más complejas, como coordenadas en un espacio bidimensional.

Para crear una matriz, simplemente se crea un array donde cada elemento es otro array.

```
let matriz = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
];
```

En este ejemplo, matriz es un array de 3x3 que contiene números organizados en tres filas y tres columnas.

Acceso a los Elementos de una matriz

Acceder a los elementos de una matriz requiere dos índices: el primero para la fila y el segundo para la columna.

```
let elemento = matriz[1][2]; // Accede al elemento en la segunda fila, tercera columna: 6
```

Modificación de Elementos en una Matriz

Puedes modificar un elemento específico de una matriz utilizando sus índices.

```
matriz[2][0] = 10; // Cambia el elemento en la tercera fila, primera columna a 10
console.log(matriz); // Imprime [[1, 2, 3], [4, 5, 6], [10, 8, 9]]
```

Autoevaluación 3

1. ¿Qué es JavaScript?

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos que se utiliza principalmente para desarrollar aplicaciones web interactivas y dinámicas. Se ejecuta en el navegador del cliente y puede manipular el contenido del documento HTML y CSS.

2. ¿Cómo se declara una variable en JavaScript?

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Las variables en JavaScript se pueden declarar utilizando var, let o const

```
var nombre = 'Juan';

let edad = 30;

const PI = 3.14;
```

3. ¿Cuál es la diferencia entre let y const?

let permite declarar variables cuyo valor puede cambiarse posteriormente, mientras que const se utiliza para declarar constantes cuyo valor no puede cambiar una vez asignado.

4. ¿Qué es una función en JavaScript y cómo se declara?

Una función en JavaScript es un bloque de código diseñado para realizar una tarea específica. Se declara de la siguiente manera:

5. ¿Qué son las funciones flecha (arrow functions)?

Las funciones flecha son una sintaxis más concisa para declarar funciones en JavaScript.

6. ¿Cómo se crea un objeto en JavaScript?

Un objeto se puede crear utilizando llaves {}

7. ¿Qué es el DOM y cómo se accede a él en JavaScript?

El DOM (Document Object Model) es una representación estructural de los documentos HTML o XML. En JavaScript, se puede acceder al DOM mediante métodos como document.getElementById, document.querySelector

8. ¿Cómo se maneja un evento en JavaScript?

Los eventos en JavaScript se manejan añadiendo un "listener" al elemento que desencadena el evento.

9. ¿Qué es una promesa (Promise) en JavaScript?

Una promesa es un objeto que representa la eventual finalización o fracaso de una operación asíncrona. Permite manejar operaciones asíncronas de manera más elegante.

10. ¿Qué es el hoisting en JavaScript?

El hoisting es el comportamiento de JavaScript de elevar las declaraciones de variables y funciones al inicio de su contexto de ejecución. Sin embargo, las inicializaciones no se elevan.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Resumen de la Unidad 3

JavaScript es un lenguaje de programación fundamental en el desarrollo web, conocido por su capacidad para crear páginas interactivas y dinámicas. Este lenguaje se ejecuta en el navegador del usuario, lo que le permite manipular el contenido y el estilo de las páginas web en tiempo real.

Para comenzar a trabajar con JavaScript, es esencial entender cómo se manejan las variables. Puedes declarar variables usando var, let o const, siendo let más moderno y adecuado para variables que cambian de valor, mientras que const se usa para valores que deben permanecer constantes.

Las funciones en JavaScript son bloques de código que puedes reutilizar para realizar tareas específicas. Existen dos formas comunes de definir funciones: la tradicional, usando la palabra clave function, y la más reciente, las funciones flecha (arrow functions), que ofrecen una sintaxis más compacta.

Un concepto clave en JavaScript es el manejo del DOM (Document Object Model), que es la representación estructural de una página web. JavaScript permite interactuar con el DOM para cambiar elementos, agregar eventos y actualizar el contenido dinámicamente.

Cuando trabajas con eventos, puedes añadir "listeners" a elementos HTML para responder a acciones del usuario, como clics o teclas presionadas. Esto te permite crear experiencias interactivas y reactivas en tu sitio web.

Además, JavaScript maneja operaciones asíncronas mediante promesas. Las promesas te permiten gestionar tareas que tardan un tiempo en completarse, como la carga de datos desde un servidor, de una manera más organizada y menos propensa a errores.

Es importante conocer el concepto de hoisting, que es el comportamiento de JavaScript de elevar las declaraciones de variables y funciones al inicio del contexto de ejecución. Aunque esto puede parecer confuso al principio, entenderlo te ayudará a evitar errores comunes en la programación.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

9. REFERENCIAS

Arias, Á. (2016). *Curso de Desarrollo Web: 2a Edición. (n.p.).* Curso de Desarrollo Web: 2a Edición. (n.p.).

Aubry, C. (2017). *HTML5 y CSS3: revolucione el diseño de sus sitios web.* HTML5 y CSS3: revolucione el diseño de sus sitios web.

Celaya Luna, A. (2014). *Creación de páginas web: HTML 5: (ed.).* ICB.

Cucaro, O. (2022). *HTML, CSS, Bootstrap, Php, Javascript y MySql: Todo lo que necesitas saber para crear un sitio dinámico. .* ResearchFreelance.

Domínguez Mínguez, T. (2023). *HTML y CSS como nunca antes se lo habían contado.* Marcombo.

Fernández Casado, P. E. (2023). *Construcción y diseño de páginas web con HTML, CSS y JavaScript:.* RA-MA.

Fontecha, J. S. (2022). *MERN: guía práctica de aplicaciones web.* RA-MA.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript.* Marcombo.

Gómez Delgado, J. (2023). *El desarrollo web desde el entorno cliente: Una visión Full Stack Developer.* ESIC Editorial.

Gómez López, J. &. (2015). *Construcción de páginas web: (ed.).* RA-MA Editorial.

Hernández, J. (2014). *Análisis y Desarrollo Web.*

Lancker, L. (2013). *Los API JavaScript de HTML5.* Ediciones ENI.

López Sanz, M. F. (2016). *MF0491_3 Programación web en el Entorno Cliente.* RA-MA Editorial.

Luna, F. (2019). *avaScript-Aprende a programar en el lenguaje de la web.* RedUsers.

Martin, S. (2017). *HTML Básico y HTML 5. .* Independently Published.

Meloni, J. (2018). *HTML, CSS, and JavaScript All in One.* Pearson Education.

Pére Rodríguez, M. D. (2018). *HTML 4.0: (2 ed.).* Editorial ICB.

Pérez Rodríguez, M. D. (2012). *JavaScript: (2 ed.).* ICB.

Pérez, J. E. (2019). *introduccion a JavaScript.*

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC

Recio García, J. A. (s.f.). *HTML5, CSS3 y JQuery: curso práctico:.* RA-MA Editorial.

Rubiales Gómez, M. (2021). *Curso de desarrollo Web. HTML, CSS y JavaScript. .* ANAYA MULTIMEDIA.

Rubiales Gómez, M. (2021). *Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021.* ANAYA MULTIMEDIA.

(+593) 96 356 1961 admisiones@itq.edu.ec Antonio de Ulloa N28-30 y Diego de Atienza (Esq). WWW.ITQ.EDU.EC