Programación Web
Del Frontend al Backend

Celi Párraga Ricardo Javier
Boné Andrade Miguel Fabricio
Mora Olivero Aldo Patricio

EDITORIAL GRUPO AEA

Programación Web

Del Frontend al Backend

Autores:

Celi Párraga Ricardo Javier

Boné Andrade Miguel Fabricio

Mora Olivero Aldo Patricio

Título del libro:
Programación Web del Frontend al Backend.
Primera Edición, 2023
Editado en Santo Domingo, Ecuador, 2023
ISBN: 978-9942-7014-5-9

© Marzo, 2023
© Editorial Grupo AEA, Santo Domingo - Ecuador
© Celi Párraga Ricardo Javier, Boné Andrade Miguel Fabricio, Aldo Patricio Mora Olivero.
Editado y diseñado por Comité Editorial del Grupo AEA
Hecho e impreso en Santo Domingo - Ecuador

Cita.

Celi Párraga, R. J., Boné Andrade, M. F., Mora Olivero, A. P. (2023). Programación Web del Frontend al Backend. Editorial Grupo AEA.

Cada uno de los textos de la Editorial Grupo AEA han sido sometido a un proceso de evaluación por pares doble ciego externos (double-blindpaperreview) con base en la normativa del editorial.

Grupo AEA
Grupo de Asesoría Empresarial y Académica
www.grupo-aea.com
www.editorialgrupo-aea.com
Grupo de Asesoría Empresarial & Académica
Grupoaea.ecuador
Editorial Grupo AEA

Aviso Legal:

La información presentada, así como el contenido, fotografías, gráficos, cuadros, tablas y referencias de este manuscrito es de exclusiva responsabilidad del autor y no necesariamente reflejan el pensamiento de la Editorial Grupo AEA.

Derechos de autor ©

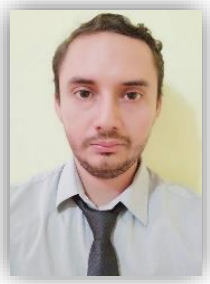
Este documento se publica bajo los términos y condiciones de la licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0).

El "copyright" y todos los derechos de propiedad intelectual y/o industrial sobre el contenido de esta edición son propiedad de la Editorial Grupo AEA y sus Autores. Se prohíbe rigurosamente, bajo las sanciones en las leyes, la producción o almacenamiento total y/o parcial de esta obra, ni su tratamiento informático de la presente publicación, incluyendo el diseño de la portada, así como la transmisión de la misma de ninguna forma o por cualquier medio, tanto si es electrónico, como químico, mecánico, óptico, de grabación o bien de fotocopia, sin la autorización de los titulares del copyright, salvo cuando se realice confines académicos o científicos y estrictamente no comerciales y gratuitos, debiendo citar en todo caso a la editorial.

Autores

Reseña de Autores

Ricardo Javier Celi Párraga



Universidad Técnica Luis Vargas Torres de Esmeraldas

✉ Correo: ricardo.celi@utelvt.edu.ec

Orcid: https://orcid.org/0000-0002-8525-5744

Ingeniero en sistemas informáticos, Máster en ingeniería del software y sistemas informáticos, Docente de La Universidad Técnica Luis Vargas Torres de Esmeraldas, Sede Santo Domingo de los Tsáchilas.

Miguel Fabricio Boné Andrade

Universidad Técnica Luis Vargas Torres de Esmeraldas

✉Correo: miguel.bone@utelvt.edu.ec

Orcid: https://orcid.org/0000-0002-8635-1869

Ingeniero de sistemas y computación, Magíster en sistemas de telecomunicaciones, Magíster en tecnologías de la información mención en seguridad de redes y comunicaciones, Docente de La Universidad Técnica Luis Vargas Torres de Esmeraldas, Sede Santo Domingo de los Tsáchilas.

Aldo Patricio Mora Olivero

Universidad Técnica Luis Vargas Torres de Esmeraldas

✉Correo: aldo.mora.olivero@utelvt.edu.ec

Orcid: https://orcid.org/0000-0002-4337-7452

Ingeniero de sistemas y computación, Magíster en tecnologías de la información, Docente de La Universidad Técnica Luis Vargas Torres de Esmeraldas, Sede Santo Domingo de los Tsáchilas.

Índice

Reseña de Autores III
Índice IV
Introducción V
Capítulo I: 1
Fundamentos de Programación Web 1
1.1. Introducción a la programación web 3
1.2. Lenguaje HTML 8
1.3. CSS 20
Capítulo II: 27
Framework CSS 27
2.1. Frameworks CSS 29
2.2. Bootstrap 31
2.2.1. Características de Bootstrap 32
2.2.2. Empezar a trabajar con Bootstrap 32
2.2.3. Grid System 33
2.2.4. Componentes 35
Capítulo III: 37
Programación Frontend 37
3.1. JavaScript 39
3.2. JQuery 46
3.2.1. Manipulación del DOM 49
Capítulo IV: 53
Programación Backend 53
4.1. Lenguajes del Backend 55
4.2. Servidores Web 57
4.3. PHP 58
4.3.1. Sintaxis básica de PHP 59
4.3.2. CRUD con PHP y MySQL 60
Referencias Bibliográficas 63

Introducción

La programación web es una disciplina que se enfoca en el desarrollo de aplicaciones y sitios web para ser utilizados en internet. Esta área de la programación se ha vuelto cada vez más importante en los últimos años debido al creciente uso de internet en nuestra vida cotidiana. Desde la creación de redes sociales hasta la gestión de grandes tiendas en línea, la programación web ha demostrado ser fundamental en el desarrollo de proyectos de gran envergadura. En este campo, se utilizan diferentes lenguajes de programación, como HTML, CSS, JavaScript y PHP, entre otros, para crear páginas web dinámicas y atractivas que proporcionen una experiencia de usuario única. En este contexto, la programación web se ha convertido en una habilidad esencial para los desarrolladores de software y los diseñadores web que desean crear sitios web modernos y efectivos.

La programación web se divide en dos áreas principales: el frontend y el backend. El frontend se refiere a la parte visible de una aplicación web, lo que los usuarios ven y con lo que interactúan. El frontend se compone principalmente de lenguajes de programación como HTML, CSS y JavaScript, que permiten la creación de páginas web dinámicas y atractivas. Por otro lado, el backend se encarga de la lógica detrás de una aplicación web. Esto incluye la gestión de la base de datos, la seguridad y la lógica de negocio de la aplicación. Los desarrolladores de backend utilizan lenguajes de programación como PHP, Ruby on Rails, Python y Java, entre otros, para crear aplicaciones web escalables y robustas que puedan manejar grandes volúmenes de tráfico y datos.

En este libro se estudia los fundamentos de la programación en el capítulo I, estudiando HTML y CSS. En el capítulo II, se estudia un Framework CSS que ayude a crear diseños atractivos como es el Framework Bootstrap. En el capítulo III, se estudia la programación Frontend con JavaScript y JQuery. En el capítulo IV, se estudia la programación Backend con PHP, llegando a crear aplicaciones conectadas a bases de datos.

Capítulo I:

Fundamentos de Programación Web

Fundamentos de Programación Web

1.1. Introducción a la programación web

En el desarrollo de proyectos Web, la generación del contenido de las páginas se lleva a cabo generalmente utilizando un conjunto de tecnologías en las que se integran tecnologías dedicadas, por un lado, a la generación del contenido estático, y por otro, tecnologías encaminadas a generar el contenido dinámico de las páginas Web.

Cuando se habla de contenido estático se hace referencia al contenido de las páginas Web, el cual se presenta de forma estática, invariante a las acciones que realiza el usuario, y cuyo objetivo primordial es presentar datos al usuario, los cuales normalmente no cambian con el paso del tiempo.

Por otro lado, cuando se habla de contenido dinámico se hace referencia a aquellas partes o elementos de las páginas Web que cambian con el tiempo, que sufren modificaciones y alteraciones por las acciones que realiza el usuario sobre el contenido.

La WEB 2.0

El término Web 2.0 está asociado a aplicaciones web que facilitan la labor de compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos que se han creado para ellos.

Frente a las tradicionales páginas web estáticas (Web 1.0) donde sus visitantes solo pueden leer los contenidos ofrecidos por su autor o editor, en la Web 2.0 todos los cibernautas pueden elaborar contenidos y compartirlos, opinar,

etiquetar/clasificar... Esto supone una democratización de las herramientas de acceso a la información y de elaboración de contenidos, aunque como no todos los que escriben en Internet son especialistas, se mezclarán los conocimientos científicos con las simples opiniones y las falsedades.

Conceptos y terminología

- Bookmark o favorito: en la World Wide Web, un bookmark es una dirección web que fue adicionada a una lista de favoritos.
- Browser o navegador: aplicación que permite al usuario navegar en la World Wide Web.
- Computación móvil: es un paradigma computacional derivado de la tecnología de redes sin cables que es utilizada por dispositivos móviles, tales como portátiles, tabletas, smartphones.
- Cookie: conjunto de información que es intercambiada entre el browser y el servidor web, y que es almacenada por el navegador del cliente.
- Download o descargar: transferir datos de otra máquina a la mía
- Gadgets: miniaplicaciones o bloques de código que son incorporados en las páginas web.
- Plugin: es un programa pequeño que sirve normalmente para adicionar funciones a otros programas mayores, adicionándoles alguna funcionalidad especial o muy específica.
- Post: mensaje publicado. Por ejemplo, entradas de texto cronológicas en blogs
- Dominio: Es un identificador de un conjunto de dispositivos que se encuentran publicados a través del internet.
- Hosting: También conocido como alojamiento, es el lugar físico en donde se encuentran los diferentes archivos (páginas web, imágenes) que serán visualizados a través de internet
- URL: Uniform Resource Locator, Localizador Universal de Recursos, es una cadena de caracteres la cual permite asignar una dirección única a cada recurso de información disponibles en el internet.

- Página web: Es un documento o archivo electrónico que es servido de forma remota o local y que es ejecutado o visualizado a través de un Browser por el cliente
- Sitios web: Es el conjunto de páginas web ordenadas que, relacionadas entre sí, brindan información específica sobre un tema, actividad o propósito específico.
- Portales web: Es la integración de diversos recursos o servicios sobre un tema, actividad o propósito especifico, que permiten al usuario interactuar con la información que se presenta de una forma fácil y amigable.

Tipos de dominio

1. Genéricos: Son dominios que no se rigen a un lugar en específico y pueden ser creados desde cualquier parte del mundo. Ejemplo:

- com: Usado para entidades comerciales
- net: Proveedores de redes
- org: Organizaciones no lucrativas
- edu: Instituciones Educativas
- gob: Entidades Gubernamentales

2. Territoriales: También llamados dominios geográficos ccTLD (Country Code Top Level Domain), las extensiones territoriales indican el país o la región de cada web. Son utilizados por las empresas para proteger la identidad de su marca en un país concreto, así como para tener más cercanía con los consumidores del país al que se va a dirigir. Ejemplo:

- mx: México
- dk: Dinamarca

El futuro con la Web 3.0

Web 3.0 es la web que facilita la accesibilidad de las personas a la información, sin depender de qué dispositivo use para el acceso a ella, una web con la que interactuar para conseguir resultados más allá del hecho de compartir "información", que esta información sea compartida por cada persona de una forma inteligible y de provecho para ella y sus necesidades en cada circunstancia.

Frontend

FrontEnd es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas, los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales… y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

Como hemos dicho, el desarrollador frontend se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Un desarrollador frontend debe conocer los siguientes lenguajes de programación: HTML5, CSS3, JavaScript, Jquery, Ajax.

Backend

Cuando hablamos de “Backend” nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina “el lado del servidor”.

El backend del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los

desarrolladores de Frontend y Backend suelen trabajar juntos para que todo funcione correctamente.

Un desarrollador Backend debe tener amplios conocimientos de los siguientes lenguajes: frameworks y los tipos de base de datos. No siendo necesario conocer todos los lenguajes, pero sí entender y saber trabajar con algunos de ellos: ASP.NET, PHP, Python, Ruby, Node.js, Java, MySQL, SQL Server, PostgreSQL, Oracle, MongoDB.

Programas más populares para la creación de páginas web

Los productos de software para la creación de páginas web permiten crear rápidamente sitios web propios con herramientas de diseño, plantillas, alojamiento de dominio y opciones de programación. También pueden incluir widgets para integración de redes sociales, procesamiento de tarjetas de crédito y funciones multimedia. Varias de estas aplicaciones también incluyen herramientas SEO integradas que mejoran el posicionamiento en los resultados de búsqueda.

- WordPress
- Joomla
- Dreamweaver
- Drupal
- Prestashop
- Google Sites
- Blogger
- Wix

CMS

Un CMS (Content Management System) o Sistema de Gestión de Contenidos es un sistema online que permite poner en marcha una página web de forma sencilla y rápida. Se trata de un software que te ayuda a administrar contenidos

dinámicos, por ejemplo, un blog, un ecommerce o cualquier tipo de página web. Especialmente es para aquellos que necesitan una actualización constante. Gracias a los CMS cualquier usuario sin ningún tipo de conocimiento en programación puede administrar a través de una interfaz gráfica todos los aspectos de una página web. Desde crear y editar contenido hasta agregar imágenes y vídeos.

Los gestores de contenidos más populares en la actualidad son: WordPress y Joomla

Editores de código para crear páginas web

Los editores de código clásicos vienen a ser una solución intermedia entre los sencillos editores de texto y los sofisticados IDE (del inglés «Integrated Development Environment», o entorno de desarrollo integrado). Los primeros destacan por su amplio rango de funcionalidad, mientras que los segundos abarcan e integran varias herramientas para el desarrollo de software en una sola aplicación. Por lo general, los complejos entornos de desarrollo integrados también tienen un editor de código incorporado, aunque este solo es uno de sus muchos componentes.

- Visual Studio Code
- Brackets
- Sublime Text
- Atom Editor
- Notepad++
- También se puede utilizar el bloc de notas de Windows.

1.2. Lenguaje HTML

HTML son las siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de “etiquetas” o “marcas”, rodeadas por corchetes angulares (< >).

HTML fue desarrollado originalmente por Tim Berners-Lee mientras estaba en el CERN, y popularizado por el navegador Mosaic desarrollado en NCSA. Durante el transcurso de la década de 1990 proliferó con el crecimiento explosivo de la Web. Durante este tiempo, se añadieron etiquetas al lenguaje HTML. La web depende de los autores de páginas web y de que las compañías compartan las mismas convenciones de HTML. Esto ha motivado el trabajo conjunto sobre las especificaciones de HTML.

HTML no es propiamente un lenguaje de programación como puede serlo Java o C, sino que se basa en la utilización de un sistema de etiquetas restringido que es aplicado a un documento de texto. Otra característica es que HTML no necesita ser compilado, sino que es un lenguaje interpretado, es decir, es ejecutado a medida que se avanza por el documento HTML y su interprete es el navegador Web.

Estructura de un documento HTML

Toda página HTML debe incluir las etiquetas <HTML> y </HTML>, estas etiquetas nos están indicando que el código contenido entre ellas va a ser HTML.

Los documentos escritos en HTML están estructurados en dos partes diferenciadas:

1. **HEAD (cabecera):** Es la primera de las dos partes en que se estructura un documento HTML. En esta zona reside información acerca del documento, y generalmente no se ve cuando se navega por él. En la zona <HEAD> se pone el elemento <TITLE></TITLE> que es una breve descripción que identifica el documento. Es lo que veremos cómo título de la ventana en los navegadores que lo permitan. Es como se conocerá nuestra página en algunos buscadores y en la agenda de direcciones (bookmarks) de los usuarios. Por tanto, parece importante pensarnos bien

cómo llamarla. Este título aparecerá en la barra superior del navegador junto con el nombre de dicho navegador.

2. **BODY (cuerpo):** El cuerpo <BODY> y </BODY> es la segunda y última de las dos partes en que se estructura un documento HTML. Esta parte al contrario que <HEAD> es obligatoria, ya que es aquí donde reside el verdadero contenido de la página.

HTML5

Lo que hace el estándar HTML5 es especificar qué marcas deben utilizarse para escribir un documento web y qué significa cada una de ellas. De esta forma, cualquier navegador sabrá cómo interpretarlas. Por ejemplo, todo documento HTML ha de ir delimitado por la etiqueta de inicio y cierre <html>. Dentro de estas dos etiquetas tenemos dos partes bien diferenciadas: Cabecera (elemento <head>) y Cuerpo (elemento <body>).

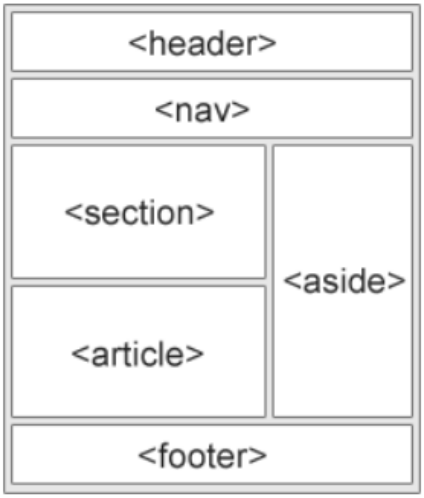
Estructura HTML

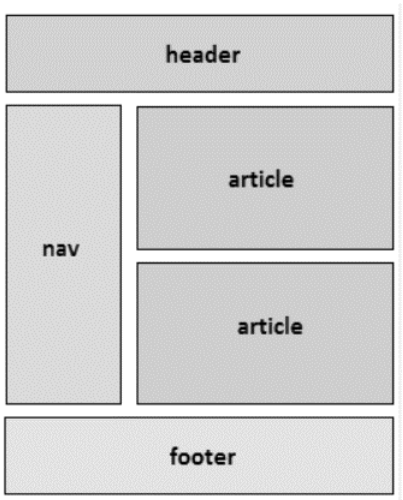
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <h1>Hola mundo</h1>
</body>
</html>
```

Estructura HTML5

Elementos como <header>, <nav>, <section>, <article>, <aside>y <footer> actúan más o menos como elementos . Agrupan otros elementos en secciones de página. Sin embargo, cuando una etiqueta puede contener cualquier tipo de información, es fácil identificar qué tipo de información iría en una región <header> semántica.

Ejemplos:





Etiquetas comunes

- <TITLE>: Establece el título de la página web.
- <DIV>: Crea un contenedor dentro de una página web.
- <TABLE>: Es la etiqueta principal para crear una tabla
- <TH>: Define una columna de encabezado dentro de una tabla.
- <TR>: Define una fila dentro de una tabla.
- <TD>: Define una columna sin encabezado dentro de una tabla.
- <P>: Define un párrafo dentro de una página web.
- <FORM>: Define un formulario dentro de una página web.
- <INPUT>: Define un campo de ingreso de datos dentro de una página web
- <LABEL>: Crea un rótulo de un input
- <IMG>: Permite la visualización de imágenes dentro de una página web.
- <LINK>: Crea un vínculo que hace referencia a otro recurso de la página web
- <A>: Crea un hiperenlace o vinculo que direcciona a otro recurso o página web
-
: Crea un salto de línea página web.
- <HR>: Crea una línea horizontal en la página web.

- <CENTER>: Permite centrar elementos en una página web.
- <H1,2,3,4,5,6>: Permite establecer texto de encabezado (Títulos) dentro de una página web.

La cabecera HTML

La cabecera está delimitada por el elemento <head> (siempre contenida dentro del elemento html) y en ella se describe información del documento y su configuración (título, configuración, scripts y estilos).

- La cabecera está delimitada por el elemento <head> (siempre contenida dentro del elemento html) y en ella se describe información del documento y su configuración (título, configuración, scripts y estilos).
- <TITLE>: Establece el título de la página web que se mostrará en la pestaña del navegador.
- Icono de la página web se puede agregar con la etiqueta link.

  <link rel="icon" href="http://example.com/favicon.png">

- Los metadatos (elementos <meta>) proporcionan metainformación sobre el documento (información procesable automáticamente por programas que analicen la página). Normalmente usan los atributos name (para definir un tipo de metadato) y content (para definir el valor),

```
<meta charset="utf-8">
<meta name="author" content="Juan Antonio Recio" />
<meta name="keywords" content="desarrollo web, html, css, ja-
vascript" />
<meta name="description" content="El mejor libro sobre desa-
rrollo web." />
```

- **charset** indica la codificación en la que se ha escrito el documento de la página. Es importante para que podamos ver correctamente las tildes en nuestra página. Como nuestro editor de texto guarda en codificación UTF-8 eso es lo que hemos indicado.
- **author** indica el autor de la página.

- **keywords y description** hacen referencia al contenido de la página y suelen ser usados por los motores de búsqueda como Google o Bing para indexar las páginas web. De esta forma, el buscador sabe qué términos de búsqueda son los que mejor describen a nuestra web para así mostrársela a sus usuarios.

El cuerpo del documento HTML

- Lo primero que debemos hacer al diseñar una página web es decidir cuál es su estructura en bloques. Normalmente toda web tiene una cabecera de título, algún tipo de barra lateral de navegación, un contenido principal y quizás un pie de página. Para hacer esta estructuración suele utilizarse el elemento o división.
- Body: Las etiquetas <body> y </body> rodean el contenido visible de la página. También puede llevar incluida información sobre las propiedades de la página, por ejemplo <body bgcolor="#RRGGBB"> define el color de fondo de la página.

Etiquetas HTML

- Comentarios: <!-- comentarios -->: Para escribir anotaciones que sirven para ayudar a la comprensión del código. Lo que se escribe dentro de esta etiqueta es ignorado por el navegador y no se muestra en la página.

Formato de Textos

- ... : Aplica negrita al texto incluido entre las etiquetas (es equivalente usar …)
- ...: Aplica cursiva al texto incluido entre las etiquetas (es equivalente usar …)
- <u> </u>: Aplica subrayado

- ...: Delimita un texto con un formato de fuente determinado definido por sus propiedades:
    - … : Define el color del texto, donde cada letra RRGGBB es un valor hexadecimal (de 0 a F) que indica el color.
    - … : Determina el tipo de fuente de texto, es decir, la tipografía. La etiqueta puede incluir los tres parámetros (tamaño, fuente y color): ……
- h1, h2, h3, h4, h5, h6: Indican 6 niveles de formato de encabezados, en los que <h1>...</h1> delimitaría el tipo de fuente de mayor tamaño.

Formato de Párrafos

-
: Introduce un salto de línea.
- ... : Delimita un párrafo de texto.
    - : Texto del párrafo con alineación centrada.
    - : Párrafo alineado a la izquierda.
    - : Párrafo alineado a la derecha. : Texto del párrafo con alineación justificada.
- <hr>: Inserta una línea horizontal.
- : Se utiliza para delimitar una sección dentro del documento, de forma que agrupe un número de elementos para luego añadirle un estilo determinado o realizar operaciones sobre ese bloque específico.

Creación de Listas

- **Lista no numerada:**

```
<ul> Inicio de la lista
    <li>primer elemento de la lista</li>
    <li>segundo elemento de la lista</li>
    <li>tercer elemento de la lista</li>
</ul> Cierra la lista
```

- **Lista numerada:**

```
<ol> Inicio de la lista
    <li>primer elemento de la lista</li>
    <li>segundo elemento de la lista</li>
    <li>tercer elemento de la lista</li>
</ol> cierra lista.
```

Creación de Enlaces

- Nombre del enlace: Cuando se pulsa sobre el texto "Nombre del enlace" en la web, se va al vínculo indicado en la dirección de href.
- Nombre del enlace: Cuando se pulsa sobre el texto "Nombre del enlace" en la web, se abre una ventana para enviar un correo electrónico a la dirección indicada.

Imágenes

- <img src="dirección de la imagen">: Inserta una imagen que se encuentra en la ruta indicada por "dirección de la imagen". También se pueden incluir una serie de propiedades:
- <img ... border="X">: Establece un borde de X pixels en torno a la imagen.
- <img ... height="XX" width="YY">: Establece un tamaño de la imagen, donde XX e YY son al altura y anchura en pixels respectivamente.
- <img ... title="texto explicativo">: Se muestra un texto al pasar el ratón sobre la imagen.
- <img ... align="bottom">: Alineación inferior de la imagen respecto al texto.
- <img ... align="middle">: Alineación de la imagen en medio del texto.
- <img ... align="top">: Alineación superior de la imagen respecto al texto.
- <img ... align="left">: Alineación izquierda de la imagen en el párrafo.
- <img ... align="right">: Alineación derecha de la imagen en el párrafo.
- <img ... hspace="x">: Espacio horizontal en pixeles entre la imagen y el texto.

- <img ... vspace="y">: Espacio vertical en pixeles entre la imagen y el texto.

Tablas

- <table>…</table>: Define dónde comienza y termina la tabla.
- <table width="X">: Determina la anchura de la tabla. Puede darse en píxeles o en porcentaje.
- <table height="X"> Determina la altura de la tabla en píxeles.
- <table border="X">: Establece el grosor en píxeles del borde de la tabla
- <table cellspacing="X">: Define el espacio en píxeles entre las celdas.
- <table cellpadding="X">: Define el espacio en píxeles entre el borde y el texto.
- <tr>…</tr>: Indica el comienzo y el fin de cada una de las filas de la tabla.
- <td>...</td>: Indica el comienzo y el fin de cada una de las columnas o celdas dentro de las filas.

Ejemplo**:**

```
<table>
    <tr>
        <th>Nombre</th>
        <th>Edad</th>
        <th>Teléfono</th>
    </tr>
    <tr>
        <td>Juan Perez</td>
        <td>32</td>
        <td>0997445142</td>
    </tr>
</table>
```

Archivos multimedia

HTML5 incluye en el propio lenguaje dos elementos que sirven para añadir audio y vídeo a nuestras páginas web:

- <video>: incrusta un vídeo en la página.
- <audio>: incrusta un elemento de audio.

Ejemplo:

```
<video src="video.mp4" controls></video>
```

```
<audio controls>
     <source src="music.mp3" type="audio/mp3">
</audio>
```

El elemento Inline Frame

<*iframe*> representa un contexto de navegación anidado, el cual permite incrustrar otra página HTML en la página actual:

```
<iframe src="default.asp" width="100%" height="300" ></iframe>
```

Formularios

Un formulario es un área especial de interactividad, dentro de la cual es posible insertar un conjunto de elementos que permiten recibir la respuesta a una o varias preguntas formuladas al usuario. Ejemplo de estos elementos son una caja de texto, un botón de opción, una lista desplegable, etc. La etiqueta necesaria para la creación de un formulario es <form></form>, que engloba el comienzo y el final del formulario. Por tanto, los elementos que reciban los datos deberán estar definidos dentro de la apertura y cierre de dicha etiqueta.

Procesamiento de formularios

La etiqueta necesita de dos atributos para que el formulario sea funcional. El primer atributo es action. Este atributo debe contener la URL a la que se redirigirán los datos del formulario. Es decir, es un hipervínculo a otra página que será la que extraiga los datos del formulario y los procese.

El segundo atributo importante es method, e indica cuál es el método por el cual el formulario va a enviar los datos. Las opciones de esta etiqueta son POST o

GET. En el caso de POST, los datos se mandan de forma oculta, mientras que en el método GET los datos son enviados de forma directa dentro de la URL.

Un formulario es una sección dentro de un documento HTML que sirve para que los usuarios introduzcan una serie de datos y los envíen a un servidor web o a un servidor de correo. Las etiquetas que se usan principalmente en los formularios HTML son:

- <form>...</form>: Indican el comienzo y el fin del formulario.
- <label>...</label>: Delimitan una etiqueta de texto dentro del formulario.
- <input>...</input>: Indican el principio y final de algún control de entrada de datos por parte del usuario. Pueden ser de varios tipos, según se defina su atributo type:
    - text: Entrada de texto.
    - password: Igual que text pero el texto introducido se presentan de tal modo que se oculten los caracteres, normalmente mediante un serie de asteriscos. Este tipo de control suele utilizarse para introducir contraseñas.
    - checkbox: Casilla de verificación o checkbox.
    - radio: Elemento de selección entre varias opciones de tipo radio.
    - submit: Botón para enviar el formulario.
    - image: Botón para enviar el formulario pero con una imagen de fondo.
    - reset: Botón para borrar todo el contenido de datos del formulario.
    - hidden: campo oculto, envía una información que no se muestra en el formulario.
    - file: Permite adjuntar un archivo desde el ordenador.
- <select>...</select>: Delimita el principio y el fin de una lista desplegable de opciones. Cada opción se indica mediante la etiqueta <option>.
- <textarea>...</textarea>: Permite introducir un texto largo.

Ejemplo de un formulario web

```
<form action="ejemplo.php" method="get">
    <p>Nombre: <input type="text" name="nombre" size="40"></p>
```

```
    <p>Año de nacimiento: <input type="number" name="nacido"
    min="1900"></p>
    <p>Sexo:
      <input type="radio" name="hm" value="h"> Hombre
      <input type="radio" name="hm" value="m"> Mujer
    </p>
    <p>
      <input type="submit" value="Enviar">
      <input type="reset" value="Borrar">
    </p>
</form>
```

Nuevos Tipos de Controles de Entradas

- <input type="email" />: Para introducir una dirección de correo electrónico. El sistema validará que la sintaxis es correcta.
- <input type="url" />: Para introducir una dirección web o url. Muestra un mensaje de error por ejemplo si se introducen caracteres no válidos.
- <input type="date" />: Permite introducir una fecha, incorpora un calendario desplegable para elegirla.
- <input type="number" />: Para introducir un número, se puede indicar un rango máximo y mínimo.
- <input type="range" />: Presenta una barra con un rango para arrastrar con el ratón.
- <input type="search" />: Para introducir un término de búsqueda en la web
- <input type="color" />: Para elegir un color. Despliega una carta de colores.

Nuevos Atributos para las Etiquetas

Entre los nuevos atributos para las etiquetas de tipo <input> se encuentran:

- autocomplete: La función de autocompletar el texto cuando se rellena un formulario es una característica del navegador. Sin embargo, HTML5 ofrece la posibilidad de activar o desactivar esta característica, dando el valor "on" u "off" a este atributo.

- autofocus: Indica si se colocará el cursor del ratón por defecto en el formulario al entrar en la página. min: Permite indicar un valor mínimo para un rango, por ejemplo en el input de tipo number.
- max: Permite indicar un valor máximo para un rango, por ejemplo en el input de tipo number.
- placeholder: Permite definir un texto que se desea que aparezca dentro del campo del formulario a modo de ayuda. Este texto se resalta en un tono diferente, de forma que, cuando se pone el cursor sobre el campo, el texto desaparece y el campo vuelve a su color habitual.
- required: Si se coloca este atributo dentro de una etiqueta input, el sistema comprobará que el campo ha sido rellenado antes incluso de pulsar el botón de envío. Si el usuario deja este campo en blanco, se mostrará un mensaje de error o simplemente se colocará el cursor de escritura sobre el primer campo vacío.

1.3. CSS

CSS (Cascading Style Sheets u Hojas de Estilo en Cascada) es un lenguaje de programación muy parecido a HTML que permite aplicar estilos a los distintos elementos de las páginas web, de modo que los títulos, listas y párrafos pueden verse igual en todas y cada una de las páginas.

Se puede hacer de tres formas diferentes, pero la más recomendada es la tercera.

1. **Mediante el atributo style:** El atributo style es de CSS pero se inserta dentro de las etiquetas HTML directamente. Por ejemplo, para aplicar el color rojo al fondo de la página, en el documento HTML se escribiría, dentro de la etiqueta body:

```
<html>
        <head>
            <title>Ejemplo</title>
        </head>
        <body style="background-color: #FF0000;">
            <p>Esta es una página con fondo rojo</p>
        </body>
</html>
```

2. **Mediante la etiqueta style:** También de esta forma se insertaría código CSS dentro del archivo HTML, en esta ocasión mediante una etiqueta HTML llamada style. Por ejemplo, para conseguir el mismo efecto que en el caso anterior, se escribiría:

```
<html>
        <head>
            <title>Ejemplo</title>

            <style type="text/css">
              body {background-color: #FF0000;}
            </style>

        </head>
        <body>
            <p>Esta es una página con fondo rojo</p>
        </body>
</html>
```

3. **Mediante un archivo CSS externo.**

```
<html>
    <head>
        <title>Mi documento</title>
        <link rel="stylesheet" type="text/css" href="estilos.css" />
    </head>
<body>
...
```

Agrupación de elementos con el atributo class

Se pueden definir estilos diferentes para un mismo tipo de etiqueta HTML, creando clases mediante el atributo class y luego indicando en CSS las propiedades de cada clase particular.

```
<p>Ropa:</p>

        <ul>
        <li><a href="camisetas.htm" class="ropa">Camisetas</a></li>
        <li><a href="pantalones.htm" class="ropa">Pantalones</a></li>
        <li><a href="camisas.htm" class="ropa">Camisas</a></li>
        </ul>

<p>Complementos:</p>
        <ul>
        <li><a href="sombreros.htm" class="complementos">Sombreros</a></li>
        <li><a href="collares.htm" class="complementos">Collares</a></li>
        <li><a href="pendientes.htm" class="complementos">Pendientes</a></li>
        </ul>
```

Para asignar las propiedades a cada clase, se utiliza la sintaxis css:

```
<style>
    a{color: blue;}
    a.ropa{color: green;}
    a.complementos{color: red;}
</style>
```

Selectores

Los selectores sirven para identificar los elementos a los que se aplicará el formato definido por la lista de declaraciones.

Selector universal

Se define con * y se aplica a todos los elementos de la página:

```
* {
    margin: 0;
    font-family: Verdana;
}
```

Selector de etiqueta HTML

Indica el estilo que se va a un elemento o etiqueta concreto.

```
h1, h2, h3, h4, h5  {
    font-size: large;
}
```

Selector de clase

Desde el punto de vista de las CSS, podemos aplicar una regla a todos los elementos que sean de una determinada clase:

```
.cabecera {
    color: blue;
}
```

Selector de identificador

Permiten aplicar un estilo único al único elemento de la página que tiene un determinado identificador (especificado con el atributo id).

```
p#destacado {
    background-color:yellow;
}
```

Propiedades de texto

- color: indica el color de la fuente. Su valor es un color.
- font-size: tamaño de la fuente.
- font-style: para letras en cursiva: normal, italic.
- font-weight: para letras en negrita o letras delgadas: normal, bold, bolder, lighter…

- text-decoration: para añadir subrayados y tachados: none, underline, line-through, overline…
- text-transform: permite pasar a mayúsculas (uppercase) o a minúsculas (lowercase).
- letter-spacing y word-spacing: permiten indicar el espacio entre las letras y palabras.
- line-height: espacio entre las líneas.
- text-align: alineación horizontal del texto en un elemento que lo contiene: left, right, center, justify.
- text-indent: para añadir indentación a un bloque de texto. El valor es una unidad de medida.

Propiedades de fondo

Para establecer el fondo de un elemento utilizamos la propiedad background-color.

Otra opción muy utilizada es desactivar la repetición con background- repeat: no-repeat; y establecer una posición fija de la imagen con background- position. Por ejemplo, para situarla en la esquina inferior derecha utilizaríamos: background-position: right bottom;. Si además queremos que la imagen esté fija y no desaparezca al hacer scroll en la página, entonces utilizamos background-attachment: fixed;. Juntándolo todo quedaría así (pruébalo por ti mismo):

```
body {
    background-image: url("fondo.png");
    background-repeat: no-repeat;
    background-position: right bottom;
    background-attachment: fixed;
}
```

Dimensiones y bordes

Para indicar las dimensiones de un elemento utilizamos las propiedades width y height (ancho y alto).

Podemos asignar un margen externo para los cuatro lados con la propiedad margin. O también podemos indicar un margen externo particular para cada lado con margin-top, margin-bottom, margin-left y margin-right. De forma análoga, tenemos los márgenes internos padding-top, padding-bottom, padding-left y padding-right.

Ejemplo

```
#lateral {
        float:right;
        width:300px;
        background-color:#FFF599;
        padding:0px 20px 0px 20px;
}

#principal {
        float:left;
        width:600px;
        background-color:#BFCFFF;
        padding:0px 20px 0px 20px;
}
```

Otras propiedades

- visibility: indica si el elemento está o no visible: visible, hidden, collapse (usado solo para elementos de tablas para no visualizar filas o columnas).
- display: cambia el tipo de caja del elemento: block, inline o none. Como veremos en la próxima sección, esto afectará a la forma en la que se posiciona un elemento en dentro de la página. El valor none suele utilizarse para hacer desaparecer elementos.

Capítulo II:

Framework CSS

Framework CSS

2.1. Frameworks CSS

Un framework CSS es una biblioteca de estilos visuales genéricos que se emplean uso para desarrollar una página web. Además de estos estilos visuales los frameworks CSS suelen añadir una serie de utilidades, como componentes para hacer cuadros de diálogo, tablas, carrusel de imágenes, etcétera.

A continuación, se menciona algunos de los frameworks CSS más usados.

Tailwind CSS

Este es un framework CSS que ofrece un enfoque diferente que otros como Bootstrap. Tailwind CSS en realidad no tiene muchos componentes, sino clases de utilidad que aplicar directamente sobre el CSS. Estos frameworks se llaman también "utility first" y ofrecen estilos CSS atómicos. Aunque también permite crear componentes, lo deja más del lado del desarrollador, que los podrá personalizar a su gusto.

Tailwind CSS tiene la característica de se muy maleable y adaptarse muy bien a lo que el desarrollador necesite. Con el framework puedes hacer builds de clases CSS totalmente personalizadas, que se parezcan o no a las que se ofrecen de manera predeterminada.

Bootstrap

Es el framework CSS más popular, en 2020, usado en infinidad de proyectos de todo tipo. Cuando apareció creó una tendencia de frameworks basados en componentes, capaces de implementar temas de diseño completos y complejos, aportando mucha sencillez y agilidad al desarrollo CSS y dotando a los

programadores de herramientas para crear diseños consistentes con poco esfuerzo.

Durante sus años de existencia ha evolucionado mucho, incorporando novedades del estándar de CSS con rapidez y eliminando dependencias pesadas como jQuery que hoy en día es innecesario en la mayoría de los proyectos.

Materialize CSS

Materialize es un framework CSS que implementa el tema de diseño "Material Design". Ofrece componetes material listos para usar, que se pueden integrar de una manera cómoda en los sitios web, consiguiendo un diseño guiado por las directrices de aplicaciones y sitios de Google.

Crea diseños consistentes que son fáciles de entender y familiares para los usuarios en general. Tiene muchos componentes que también implementan comportamientos dinámicos con Javascript y recientemente se actualizó para eliminar su dependencia con jQuery.

El framework es sencillo de usar, relativamente ligero, permite optimización y los componentes están altamente personalizados en su diseño.

Foundation

Con Bootstrap y Materialize, Foundation es uno de los frameworks CSS basados en componentes más importantes del momento. Tiene muchos seguidores y mantiene una base de código que se actualiza con frecuencia para incorporar las novedades de CSS.

A diferencia de otros frameworks CSS ofrece un subset de sus herramientas orientado a la creación de emails, que tienen características de desarrollo distintos que los sitios web. Ofrece un set de herramientas para poder optimizarlo

correctamente y construir un build que incluya solamente los componentes que se desean.

Bulma

Este es un framework CSS basado en componentes que usa Flexbox de manera precisa para simplicidad del código.

Los componentes de Bulma son CSS puros, es decir, no tiene el set de componentes que requieren Javascript para funcionar. No obstante, tiene un montón de extensiones que sí incluyen componentes de todo tipo, dinámicos y con Javascript, usando varios frameworks, creados por terceros. Es sencillo de usar y de aprender y permite un desarrollo moderno.

2.2. Bootstrap

En los últimos años y con la aparición de la denominada web 2.0, Internet ha sufrido un importante cambio para ir adaptándose a todas las necesidades de los usuarios, y por consiguiente, los sitios web también han tenido que cambiar mucho. Uno de los cambios más significativos, fue la llegada del concepto responsive, es decir, diseños que se adaptan al tamaño de pantalla de cualquier dispositivo capaz de navegar por la red.

Esto es posible gracias al uso de técnicas avanzadas de CSS, o bien, mediante el uso de frameworks desarrollados por terceras personas o empresas. Es el caso de Bootstrap creado por Twitter. A lo largo de este artículo vamos a hablar sobre Bootstrap, uno de los frameworks CSS más famosos y utilizados hoy en día por los desarrolladores de portales web.

¿Qué es Bootstrap?

El framework Bootstrap vio la luz en el año 2011. En un principio fue desarrollado por Twitter aunque posteriormente fue liberado bajo licencia MIT. Hoy en día

continúa su desarrollo en un repositorio de GitHub. Se trata de un framework que ofrece la posibilidad de crear un sitio web totalmente responsive mediante el uso de librerías CSS. En estas librerías, nos podemos encontrar un gran número elementos ya desarrollados y listos para ser utilizados como pueden ser botones, menús, cuadros e incluso un amplio listado de tipografías.

Desde que vio la luz, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño.

2.2.1. Características de Bootstrap

- **Fácil de usar**: cualquiera con conocimientos básicos de HTML y CSS puede comenzar a usar Bootstrap
- **Funciones de respuesta**: el CSS responsivo de Bootstrap se ajusta a teléfonos, tabletas y computadoras de escritorio
- **Enfoque móvil primero:** en Bootstrap, los estilos primero para dispositivos móviles son parte del marco central
- **Compatibilidad con el navegador:** Bootstrap es compatible con todos los navegadores modernos (Chrome, Firefox, Internet Explorer 10+, Safari y Opera

2.2.2. Empezar a trabajar con Bootstrap

Hay dos formas de comenzar a utilizar Bootstrapen su propio sitio web. Usted puede:

1. Incluye Bootstrap desde un CDN
2. Descargue Bootstrap desde getbootstrap.com

Bootstrap CDN

La gente de StackPath amablemente proporciona soporte CDN para CSS y JavaScript de Bootstrap. Revisar https://getbootstrap.com/

Descarga

Si desea descargar y alojar Bootstrap usted mismo, vaya a https://getbootstrap.com y siga las instrucciones allí.

Una vez descargado, descomprima la carpeta comprimida para ver la estructura de Bootstrap (compilado). Verás algo como esto:

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

2.2.3. Grid System

Sin duda, una de las características más importantes y útiles que ofrece Bootstrap es su sistema de rejilla donde se pueden adaptar todos los elementos al ancho de 12 columnas que varían su tamaño dependiendo del dispositivo desde el cual se visualiza la web. Este sistema está pensado para ayudarnos en la disposición de los componentes de nuestra web y su adaptación a los diferentes tamaños de las pantallas.

El funcionamiento de este sistema se puede resumir en los siguientes puntos.

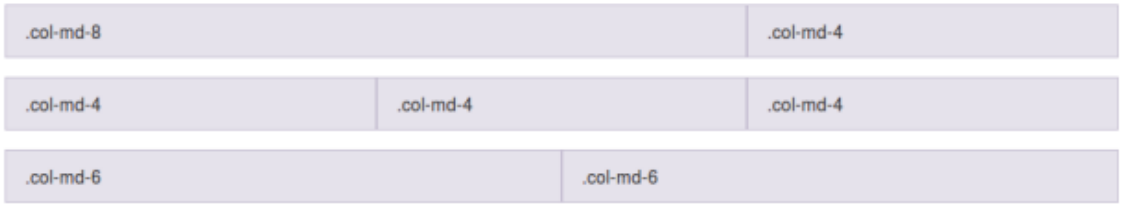
- Las columnas de la rejilla irán agrupadas dentro de las filas, que se indicarán con la clase ".row". Las filas se deben colocar dentro de una etiqueta contenedora que se indicará con la clase ".container" para ancho fijo o ".container-fluid" para ocupar todo el ancho de la pantalla.

- El contenido se debe disponer dentro de columnas, las cuales serán los únicos hijos de las filas, y que a su vez serán los únicos hijos del contenedor
- Cada fila se puede dividir hasta un máximo de 12 columnas, pero seremos nosotros los que tendremos que definir el número de columnas en el que queremos dividir cada fila. En cualquier caso, la suma del tamaño de esas columnas no puede superar el valor de 12. Es decir, podemos crear una columna que ocupe 6 espacios, y otras dos que ocupen 3 cada una de ellas.
- El tamaño de las columnas se especificará con clases CSS que ofrece Bootstrap y que define para cada tamaño de pantalla. Por ejemplo, col-md-xx, donde xx podrá tomar un valor comprendido entre 1 y 12. Las posibles clases que podemos utilizar son:
    - **.col-xs-:** Pensado para pantallas cuyo tamaño sea menor de 768 px.
    - **.col-sm-:** Para pantallas mayor o igual a 768 px.
    - **.col-md-:** Para pantallas mayor o igual a 992 px.
    - **.col-lg-:** Para pantallas mayor o igual a 1.200 px.

A continuación, os dejamos un ejemplo de uso de este sistema de rejilla.

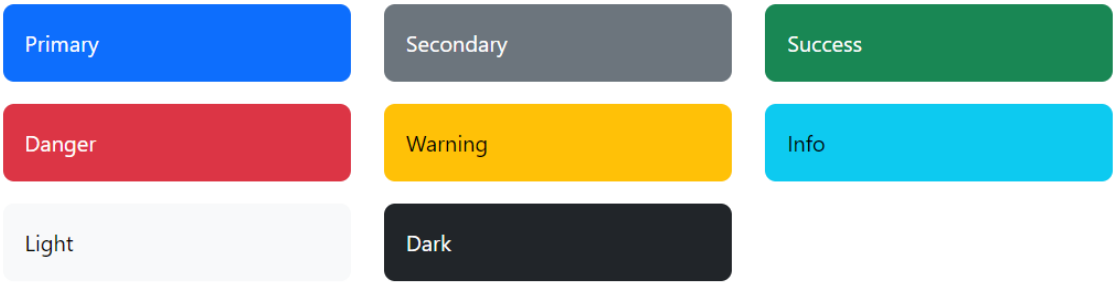
```
<div class="row">
      <div class="col-md-8">.col-md-8</div>
      <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
      <div class="col-md-6">.col-md-6</div>
      <div class="col-md-6">.col-md-6</div>
</div>
```

El resultado de ese ejemplo, es lo que se puede ver en la siguiente imagen.



2.2.4. Componentes

Las clases de Bootstrap se basan en los siguientes colores:



Ejemplo de botones:

Ejemplo de implementación de Bootstrap en un proyecto HTML.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
     alpha2/dist/css/bootstrap.min.css" rel="stylesheet"
     integrity="sha384-
     aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
     crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
     alpha2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
     qKXV1j0HvMUeCBQ+QVp7JcfGl760yU08IQ+GpUo5hlbpg51QRiuqHAJz8+BrxE/N"
     crossorigin="anonymous"></script>
  </body>
</html>
```

Todos los componentes de Bootstrap se encuentran detallados en la página web de Bootstrap: https://getbootstrap.com/docs/

Capítulo III: Programación Frontend

Programación Frontend

3.1. JavaScript

Hasta ahora hemos estudiado las dos tecnologías básicas para crear un sitio web —HTML, que nos permite definir el contenido; y CSS, que nos permite darle formato—, solo nos falta la tercera tecnología fundamental para el desarrollo web del lado del cliente: JavaScript. Esta tecnología posibilita la interacción con el usuario. La estandarización de JavaScript con HTML5 en todos los navegadores nos permite crear innumerables tipos de aplicaciones web que pueden ejecutarse en cualquier plataforma (tanto de escritorio como móvil).

Incluir JavaScript en un documento HTML

De forma similar a como hacíamos con los ficheros CSS, para incluir JavaScript dentro de nuestro código tenemos las siguientes opciones:

1. Incluir el código dentro de los elementos HTML, generalmente para manejar eventos o para escribir código dentro de la página (no recomendado)

```
<input type="button" value="Pulse este botón"
onclick="alert('¡Has pulsado el botón!');">
```

2. Incluir el código con elementos <script> en cualquier parte del documento. Aunque lo hemos hecho en los ejemplos anteriores, no se recomienda utilizar esta opción.

```
<script type="text/javascript">
  //Código JavaScript
</script>
```

3. La opción más recomendable es incluir en cualquier parte del documento una referencia a un archivo externo con extensión .js

```
<script type="text/javascript" src="./js/miJavascript.
js"></script>
```

Conceptos básicos de JavaScript

Un programa en JavaScript es una secuencia de sentencias u órdenes separadas por punto y coma (;). Aunque no es obligatorio, es muy recomendable ser estrictos con esto ya que si no escribimos nosotros los puntos y comas (;) será el intérprete quien los escriba (y puede cometer errores). JavaScript distingue entre mayúsculas y minúsculas, por lo que las palabras while y WHILE no representan lo mismo dentro del lenguaje (la primera es una palabra reservada, la otra no). JavaScript ignora espacios en blanco, tabuladores y saltos de línea entre palabras reservadas pero es conveniente utilizar sangrado para que los scripts sean más legibles. Por último, se pueden poner comentarios que ocupan una línea (comenzando por // hasta el final de línea) o varias líneas (entre /* y */).

Variables y expresiones

Las variables son posiciones de memoria en las que guardamos información que puede ser utilizada dentro del script. Las variables se pueden declarar explícitamente con la palabra reservada var y su nombre puede contener letras o dígitos y empezar por guión bajo (_). Para asignar un valor a una variable se usa el operador de igualdad (=). Se puede utilizar tanto al declarar la variable (esto se llama inicializar una variable) como en cualquier otro momento:

```
var x,y;  // x e y están sin inicializar
var cadena = "miCadena";
cadena = "otra cadena";
var a = 0;
```

Todas las variables en JavaScript se crean de la misma forma utilizando la palabra reservada *var* sin embargo y a diferencia de otros lenguajes de

programación, el tipo de las variables es dinámico, de manera que el programador puede utilizar la misma variable para almacenar diferentes tipos de datos.

Además de la expresión de asignación anterior, existen otras muchas expresiones creadas mediante combinaciones de operadores:

```
var a = 3+2;              // a vale 5
a+=6;                     // a vale 11 (5+6)
var b = a * 7;            // b vale 77 (11*7)
a < b;                    // true (verdadero)
a >= b;                   // false (falso)
```

Entrada y salida del usuario con el navegador

- **var x = prompt("Texto a escribir"):** muestra una ventana en la que el usuario puede escribir un texto. Tras pulsar en Aceptar, la cadena escrita se guarda en x (podemos poner cualquier nombre a la variable).
- **console.log("Mensaje"):** escribe en la consola de JavaScript el mensaje indicado. También puede ser el contenido de una variable o el resultado de una expresión.
- **alert("Mensaje"):** muestra una ventana emergente con el mensaje indicado. Al igual que antes, también puede ser el contenido de una variable o el resultado de una expresión.

  Ejemplo

```
var nombre = prompt("Dime tu nombre");
console.log("Tu nombre es: "+ nombre );
alert("Hola "+nombre);
```

Instrucciones condicionales

```
if ( /* Condición */) {
    // Instrucciones a ejecutar si la condición
    // se evalúa a true
}
else {
    // Instrucciones a ejecutar si la condición
    // se evalúa a true
}
```

```
switch(/* expresión /) {
    // La expresión devuelve un número, una cadena o un valor
lógico
    case valor1:
        // Instrucciones valor1
        break;
    case valor2:
        // Instrucciones valor2
        break;
    default:
        // Opcional: Instrucciones que se hacen si no se cum-
ple
        // ninguno de los demás casos.
}
```

Bucles

```
while( /* Condición */) {
    // Bloque de instrucciones
    // Se ejecutan mientras condición === true
}
```

```
for (inicialización; condición; actualización) {
    // Instruciones a repetir
}
```

Funciones

```
function nombre (param1, param2, ...) {
    /* instrucciones  de la función */
}
```

Ejemplo:

```
function multiplicar(a, b) {
  var prod = 0;
  var i = 1;
  while (i <= b) {
    prod += a;
    i++;
  }
  return prod;
};

console.log(multiplicar(2, 3));
console.log(multiplicar(10, 100));
```

Arrays y Cadenas

Un array es una colección de variables de este o de distinto tipo.

```
var nombre_array = [valor0, valor1, ..., valorN-1];
var sin_inicializar = new Array(5);
```

Podemos acceder a cada uno de los elementos del array con el operador []. Los elementos están numerados entre 0 y N-1, donde N es el tamaño del array (el número de elementos que almacena).

```
console.log(nombre_array[0]);
nombre_array[1] = "otro valor";
console.log(sin_inicializar[7]); // Fuera de rango: undefined
```

Ejemplo

```
var lista = [12,10,4,35];

// Ahora mostramos el contenido
for (i=0; i < lista.length; i++) {
    console.log(lista[i]+ " ");
}
```

Las Cadenas

Las cadenas (strings) de JavaScript permiten realizar algunas de las operaciones que usamos sobre los arrays. En particular, el operador + es equivalente a concat():

```
var cadena = "hola" + " mundo";
console.log(cadena[0]);  // Muestra la "h"
```

El Objeto Window

- **console:** es el objeto utilizado para escribir por consola. Como vemos, para escribir en consola deberíamos haberlo referenciado como window. console.log(...), pero no es necesario indicar que está dentro de window.
- **alert(msg):** es un método que ya hemos utilizado para presentar mensajes en el navegador.
- **prompt (msg):** es un Método que solicita al usuario información mediante una línea de edición.
- **confirm(msg):** muestra una ventana con un mensaje que solicita la confirmación del usuario.
- **location:** es un objeto que contiene la URL, de la página actualmente cargada. Nos permite cargar una nueva página con location.assign(URL), teniendo en cuenta la política de same-origin.

Los eventos onclick Y ondblclick

Sin lugar a duda, éste es uno de los eventos más conocidos de JavaScript, y se produce cuando el usuario efectúa un clic (onclick) o un doble clic (ondblclick) con algún botón del ratón sobre algún elemento de un documento.

<input type="tipo de elemento" onclick="función" />

Ejemplo evento onclick:

```
<input type="button" value="Pulsa 1 click" onclick="alert('OK. Botón
pulsado');" />

<input type="button" value="Pulsa doble click" ondblclick="alert(OK.
Botón pulsado');" />
```

LOS EVENTOS onmouseover Y onmouseout

El evento onmouseover sucede cada vez que el cursor del ratón pasa por encima de un elemento de la página, mientras que el evento onmouseout sucede cuando se deja de seleccionar dicho elemento.

```
<a href="url" onmouseover="función">Enlace</a>
<a href="url" onmouseout="función">Enlace</a>
```

EL EVENTO onchange

El evento onchange se produce cuando un elemento select, text o textarea de un formulario pierde el focus y además su valor ha sido modificado.

Para que el evento funcione el usuario tiene que cambiar de elemento en el formulario, es decir, si estamos introduciendo datos en una casilla de texto, el evento onchange no se activará hasta que no pasemos a otra casilla.

```
<input type="tipo de elemento" onchange="función" />
```

EL EVENTO onsubmit

Este evento se encarga de ejecutar un determinado código JavaScript al realizarse el envío de un formulario.

```
<form method="post" action="documento.htm" onsubmit="return
verificaForm()'>
```

Otros Eventos en JavaScript

- **Onfocus:** detecta la entrada (focus) en un elemento de un formulario, por ejemplo un text, textarea o window,
- **Onblur:** actúa de manera contraria, es decir, detecta la pérdida del focus.
- **Onselect**: ejecute un script cuando se selecciona texto dentro de un elemento de un formulario, ya sea una casilla de texto o de área de texto.
- **Onreset**: se produce cuando se pulsa un botón de Reset en un formulario.
- **onkeydown**. Se produce cuando se pulsa una tecla.
- **onkeyup**. Se produce cuando se deja de pulsar una tecla.
- **onkeypress**. produce cuando se mantiene pulsada una tecla.

3.2. JQuery

JQuery es una librería muy sencilla de JavaScript que responde a la filosofía “escribe menos, haz más”. El propósito de esta librería es hacer mucho más fácil escribir código JavaScript. Para ello, permite realizar en una sola llamada tareas que requerirían varias líneas de código JavaScript. Además, simplifica muchas de las tareas más complicadas de JavaScript, como AJAX o la manipulación del DOM.

La librería JQuery nos va a permitir manipular el árbol DOM y las propiedades CSS, personalizar los eventos HTML, crear efectos o animaciones, y realizar llamadas AJAX.

Instalación y carga

Simplemente descarga la librería y añádela como un script más en tu página HTML. Puedes descargarla desde jquery.com.

```
<script src="jquery-2.1.3.min.js"></script>
```

Otra opción es utilizar alguna de las versiones alojadas tanto en los servidores de Google con en los de Microsoft (lo que se conoce como content delivery network):

```
<script src=
"https://ajax.qgoogleapis.com/ajax/libs/iquery/2.1.3/jquery.min.js">
</script>
```

Dentro de ese fichero JavaScript debes tener en cuenta que los métodos JQuery se deben ejecutar una vez que el documento esté cargado. Para ello utilizamos el evento correspondiente:

```
<script>
  $ (document).ready(function()(
  // Código jQuery
  ));
</script>
```

Sintaxis básica

La sintaxis de JQuery es muy sencilla y, en cierto modo, bastante similar a la de CSS. Todas las llamadas se componen de un selector seguido por una acción:

```
$(selector).accion()
```

- Con el signo $ accedemos a JQuery.
- El selector nos permite seleccionar elementos HTML.
- La acción se realiza sobre los elementos seleccionados.

Veamos unos ejemplos

```
$(this) .hide() // oculta el elemento actual
$("p") .hide() // oculta todos los elementos “<p="
$(".test") .hide() // oculta todos los elementos con clase “test”
$("#test") .hide() // oculta todos los elementos con id "test”.
```

Selectores en la práctica

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/1.11.3/jquery.min.js"></script>
<script>
$(function(){
    $("#btn01").click(function(){
        $("#importante").hide();
    });
    $("#btn02").click(function(){
        $(".color").hide();
    });
    $("#btn03").click(function(){
        $("h4").hide();
    });
});

</head>
<body>

<h4>Una sección</h4>
<p class="color">Un párrafo a color</p>
<p class="color">Otro párrafo a color</p>
<p id="importante">Un párrafo importante</p>
<h4>Otra sección</h4>

<button id="btn01">Quitar elemento 'importante' </button>
<button id="btn02">Quitar elementos clase 'color'</button>
<button id="btn03">Quitar todos los encabezados de sección</
button>

</body>
</html>
```

El método val

El método val () devuelve o establece el atributo de valor de los elementos seleccionados.

Devuelve el atributo de valor: $(*selector*).val()

Establecer el atributo de valor: $(*selector*).val(*value*)

3.2.1.Manipulación del DOM

EL DOM

Las siglas DOM significan Document Object Model, o lo que es lo mismo, la estructura del documento HTML. Una página HTML está formada por múltiples etiquetas HTML, anidadas una dentro de otra, formando un árbol de etiquetas relacionadas entre sí, que se denomina árbol DOM (o simplemente DOM).

En Javascript, la forma de acceder al DOM es a través de un objeto llamado document, que representa el árbol DOM de la página o pestaña del navegador donde nos encontramos. En su interior pueden existir varios tipos de elementos HTML.

- **text()** - Establece o devuelve el contenido de texto de los elementos seleccionados
- **html()** - Establece o devuelve el contenido de los elementos seleccionados (incluido el marcado HTML)
- **val()** - Establece o devuelve el valor de los campos del formulario.
- **attr ()** - El attr()método jQuery se usa para obtener valores de atributos.

Ejemplo:

```
$("#btn1").click(function(){
  alert("Text: " + $("#test").text());
});
$("#btn2").click(function(){
  alert("HTML: " + $("#test").html());
});

$("button").click(function(){
  alert($("#w3s").attr("href"));
});
```

```
$("#btn1").click(function(){
  $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
  $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
  $("#test3").val("Dolly Duck");
});
```

Eventos de jQuery

jQuery está hecho a medida para responder a eventos en una página HTML.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Ejemplos:

```
$("p").click(function(){
  // action goes here!!
});
```

```
$("p").dblclick(function(){
  $(this).hide();
});
```

```
$("#p1").mouseenter(function(){
  alert("You entered p1!");
});
```

Añadir o eliminar propiedades CSS

Mediante los métodos addClass() y removeClass() podemos asignar o quitar una clase CSS a cualquier elemento.

```
$("button").click(function(){
  $("h1,h2,p").addClass("color");
  $("div").addClass("importante");
});
$("button").click(function(){
  $("h1,h2,p").removeClass("color");
});
```

Manipulación de las propiedades CSS

El método css() nos permite consultar o modificar cualquier propiedad de la hoja de estilos de un elemento HTML. Su funcionamiento es sencillo:

```
$("p").css("background-color","yellow");
```

Hide y Show

Estos dos métodos permiten ocultar o mostrar un elemento. Podemos indicar la velocidad (en milisegundos) como argumento del método:

```
<button id="hide">Ocultar</button>
<button id="show">Mostrar</button>

$("#hide").click(function(){
  $("p").hide(800);
});

$("#show").click(function(){
  $("p").show(1000);
});
```

Fadein y Fadeout

Hace que un elemento oculto aparezca. Puede indicarse la velocidad en milisegundos o con los valores “slow” o “fast”:

```
$("button").click(function(){
  $("#div1").fadeIn();
  $("#div2").fadeIn("slow");
  $("#div3").fadeIn(3000);
});
```

```
$("button").click(function(){
  $("#div1").fadeOut();
  $("#div2").fadeOut("slow");
  $("#div3").fadeOut(3000);
});
```

Capítulo IV: Programación Backend

Programación Backend

Backend es un término que se usa para referirse a un ámbito de la programación de aplicaciones, la parte que se encarga del acceso a los datos y la seguridad de la información. Aunque se usa mucho para referirnos a la programación web que se realiza del lado del servidor, es válido hablar de backend para el desarrollo de todo tipo de aplicaciones.

Dependiendo de la fuente, podemos ver este término escrito como backend, Back end, o Back-end.

Para explicar lo que es backend debemos acudir al término frontend, ya que en la división de tareas es donde ganan su significado ambos términos. Mientras que en frontend nos ocupamos de la capa de presentación, en el backend nos ocupamos del acceso a los datos.

En el desarrollo de aplicaciones modernas generalmente se desea un aislamiento de lo que es el código backend del código frontend, pues facilita el mantenimiento y la escalabilidad de los proyectos. Esta separación de responsabilidades, entre el código backend y el código frontend es la que produce que haya profesionales especializados en una u otra área.

4.1. Lenguajes del Backend

En el ámbito backend podemos usar lenguajes muy variados, siendo los más populares:

- PHP
- ASP
- JSP
- Ruby
- NodeJS

- Python
- Otros

La elección de la tecnología de backend suele ser bastante arbitraria, no existiendo un mejor lenguaje para la realización de todo tipo de tareas. Pueden existir lenguajes más apropiados para determinados proyectos, pero generalmente todos permiten acceder a los mismos recursos, por lo que la elección final depende muchas veces de intereses o experiencia del equipo de desarrollo con una alternativa particular.

ASP

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

ASP ha sido comercializada como un anexo a Internet Information Services (IIS).

ASP.NET

ASP.NET es un framework de desarrollo web del lado del servidor creado por Microsoft y altamente desarrollado. ASP.NET se utiliza para crear páginas web dinámicas, aplicaciones web y servicios basados en web. Después de décadas de desarrollo, el framework sigue existiendo bajo el nombre de ASP.NET Core.

PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto.

Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:

1. Es de código abierto, no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.
2. Está en constante perfeccionamiento, gracias a una comunidad de desarrolladores proactiva y comprometida.

4.2. Servidores Web

Apache

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Le permite a los propietarios de sitios web servir contenido en la web y es uno de los servidores más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

WampServer

WampServer es un ambiente de desarrollo web popular para Windows que permite la creación de aplicaciones que se sustentan en Apache, PHP y bases de datos MySQL. Este genial paquete todo-en-uno tiene todo lo que necesitas para empezar a desarrollar aplicaciones web, configurar finamente su servidor y

crear un servicio poderoso de sitios web que pueden ser accedidos por millones de usuarios de Internet. WampServer ofrece un proceso de instalación dinamizado y una herramienta fácil de usar para manejar Apache y los servicios de MySQL, actualización fácil de nuevos lanzamientos de bases de datos, gestión de configuración del servidor, logs detallados y más.

4.3. PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto. Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:

1. Es de código abierto, no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.
2. Está en constante perfeccionamiento, gracias a una comunidad de desarrolladores proactiva y comprometida.

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

La manera más rápida de aprender PHP es entendiendo los ejemplos y adaptándolos a las necesidades de cada proyecto web. Todo lo que se necesita para la programación con PHP es un servidor web con un intérprete de PHP, un editor de textos (por ejemplo, Notepad++ o Visual Studio Code) y un navegador web. Como servidor para una primera inclusión es recomendable utilizar el entorno de prueba local XAMPP o WAMP, puesto a disposición por Apache Friends para los sistemas operativos Windows, Linux y macOS de forma gratuita.

4.3.1. Sintaxis básica de PHP

- Un script PHP se puede colocar en cualquier parte del documento.
- Un script PHP comienza <?php y termina con ?>

  Ejemplo:

```
<?php
// PHP code goes here
?>
```

- La extensión de archivo predeterminada para los archivos PHP es " .php".
- Para ponre comentarios se utiliza: //
- A continuación, tenemos un ejemplo de un archivo PHP simple, con un script PHP que utiliza una función PHP integrada " echo" para generar el texto "¡Hola mundo!" en una pagina web:

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>
```

```
</body>
</html>
```

- En PHP, una variable comienza con el signo $, seguido del nombre de la variable:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- Echo y print son más o menos lo mismo. Ambos se utilizan para enviar datos a la pantalla.

```
echo "Hello world!";
print "Hello world!";
```

4.3.2. CRUD con PHP y MySQL

CRUD (Create, Read, Update, Delete)

En informática, CRUD es el acrónimo de "Crear, Leer, Actualizar y Borrar", que se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

Ejemplo:

Index.html

```
<form action="agregar.php" method="post" >
      <input type="text" name="cedula" id="" class="form-control"
         placeholder="Cédula">
      <input type="text" name="nombre" id="" class="form-control"
         placeholder="Nombres">
      <input type="text" name="edad" id="" class="form-control"
         placeholder="Edad">
      <input type="text" name="correo" id="" class="form-control"
         placeholder="Correo">
      <input type="submit" value="Guardar" class="btn btn-primary" >
</form>
```

Conexión

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "bd1";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Agregar

```
$sql = "INSERT INTO persona (cedula, nombre, edad, correo)
        VALUES ('$cedula','$nombre','$edad','$correo')";
$conn->query($sql);
```

Editar

```
$sql = "UPDATE persona SET
nombre='$nombre',
edad='$edad',
correo='$correo'
WHERE cedula='$cedula'";
$conn->query($sql);
```

Eliminar

```
$sql = "DELETE FROM persona WHERE cedula='$cedula'";
$conn->query($sql);
```

Leer

```
$sql = "SELECT * FROM persona";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo'<tr>
                <td>'.$row['cedula'].'</td>
                <td>'.$row['nombre'].'</td>
                <td>'.$row['edad'].'</td>
                <td>'.$row['correo'].'</td>
            </tr>';
    }
}
$conn->close();
```

Referencias Bibliográficas

Referencias Bibliográficas

Bootstrap. (2023). *Get started with Bootstrap*. Obtenido de https://getbootstrap.com/docs/

López, M., & Sanchez, D. (2015). *Programación web en el entorno cliente.* Madrid, España: RA-MA. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5759023

Lopez, M., & Soltero, F. (2016). *Programación web en el entorno servidor.* Madrid, España: RA-MA. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5759024

Mohedano, J., & Saiz, J. M. (2012). *Iniciación a JavaScript.* Madrid, España: Ministerio de Educacion España. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=3214795

Recio, J. A. (2016). *HTML5, CSS3 y JQuery.* Madrid, España: Ra-Ma. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5758976

Ceballos, J. (2013). *Enciclopedia de Microsoft Visual C#.* Madrid, España: Ra-MA. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=3229332

Ferrer, J. (2014). *Aplicaciones Web.* Madrid, España: Ra-MA. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5758903

López, J. (2014). *Domine JavaScript.* Madrid, España: Ra-MA. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5758942

Orós, J. C. (2014). *Diseño de páginas Web con XHTML, JavaScript y CSS.* Madrid, España: Ra-Ma. Obtenido de https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=5758937

Mohedano, J., & Saiz, J. M. (2012). *Iniciación a JavaScript.* Madrid, España: Ministerio de Educacion España. Obtenido de

https://ebookcentral.proquest.com/lib/uteclvtsp/reader.action?docID=3214795

PHP. (2023). ¿Qué es PHP?. Php.net. Obtenido de https://www.php.net/manual/es/intro-whatis.php

Resumen

En este libro se investigó los fundamentos de la programación web, así como las tecnologías esenciales para lograr crear aplicaciones web, tales como HTML y CSS. Se documentó los conceptos básicos de la programación Frontend utilizando el lenguaje de programación JavaScript, se documentó también los conceptos de la programación Backend utilizando el lenguaje de programación PHP. Este libro tiene como objetivo involucrarse en la programación web iniciando con el estudio del Frontend hasta el Backend con los conceptos básicos, códigos y ejemplos, utilizando tecnologías de programación gratuitas accesibles para cualquier estudiante, para llegar a crear aplicaciones dinámicas con acceso a base de datos. Como resultado se obtuvo que se puede llegar a crear aplicaciones web complejas utilizando tecnologías actuales gratuitas, con un diseño agradable, diseñándolo fácilmente con un Framework CSS como Bootstrap, y utilizando JQuery como una librería para crear rápidamente aplicaciones con JavaScript. Se utilizó además PHP por ser uno de los lenguajes más utilizados para crear aplicaciones conectadas a bases de datos MySQL.

Editorial Grupo AEA

www.grupo-aea.com

www.editorialgrupo-aea.com

Grupo de Asesoría Empresarial & Académica

Grupoaea.ecuador

Editorial Grupo AEA

ISBN: 978-9942-7014-5-9

9 789942 701459