

CUADERNO DE TRABAJO

**CUADERNO DE:
PROGRAMACIÓN V**

**GUIA DE ESTUDIOS PARA
SEXTO NIVEL DE LA
CARRERA:
TECNOLOGÍAS DE LA
INFORMACIÓN**

**DOCENTE:
ING. RICARDO CELI**

**SEDE SANTO DOMINGO DE
LOS TSÁCHILAS LA
CONCORDIA**



CUADERNO DE ENFOQUES ADMINISTRATIVOS

Facultad: Ingenierías	Carrera: Tecnologías de la información
Asignatura: Programación V	Código: TICS1223356
Profesor: Ing. Ricardo Javier Celi Párraga.	Periodo académico: IIS - 2023.
Email: ricardo.celi@utelvt.edu.ec	

Contenido

INTRODUCCIÓN	3
OBJETIVO.....	3
UNIDAD I: PROGRAMACIÓN WEB DEL LADO DEL CLIENTE	4
HTTP	4
URI.....	7
Lenguaje HTML.....	8
Estructura de un documento HTML	8
Etiquetas comunes	9
BOOTSTRAP	10
Trabajo Autónomo	20
Trabajo Autónomo	21
INTRODUCCIÓN A JQUERY	22
Manipulación DOM de jQuery.....	24
Eventos de jQuery	24
UNIDAD 2. PROGRAMACIÓN BACKEND	27
Lenguajes Backend	27
WampServer	28
Apache	29
PHP	30
Sintaxis básica de PHP	30
DECLARACIONES CONDICIONALES DE PHP.....	34
Formularios PHP.....	36
Sesiones en PHP	37
Actividad Práctica 4	39
Bibliografía	43

DETALLE DEL CONTENIDO DE LA GUÍA

Estimado estudiante, la metodología que utilizaremos en la guía didáctica, implica su participación con la guía del docente, de manera flexible de acuerdo con la realidad del estudiante.

METODOLOGÍA DE CONSTRUCCIÓN DE LA GUÍA

Se construye la guía didáctica en base a los contenidos del silabo planificado de acuerdo con la asignatura. (Leer el silabo entregado por el docente)

- **CONTENIDOS**

Los mismos son obtenidos de la unidad que consta en el silabo y además la planificación correspondiente realizada para las temáticas a impartir según considere el docente.

- **TRABAJO AUTÓNOMO**

Trabajo que realizará cada estudiante después de las tutorías virtuales según lo indique el docente,

- **ACTIVIDADES POR DESARROLLAR**

Se desarrollan al final de cada unidad

- **RÚBRICA CON LA QUE SE EVALUARÁ SU TRABAJO**

Establecido por la comisión de las áreas pertinentes, para la evaluación a los estudiantes

INTRODUCCIÓN

La asignatura de programación web (Programación V) corresponde a la Unidad de Formación Profesionalizante y al Campo de Formación de praxis profesional; Tiene como propósito desarrollar en el estudiante la capacidad de elaborar aplicaciones web haciendo uso de los principios fundamentales de la programación web.

En esta asignatura se crean las bases para la utilización de lenguajes de programación web y se le brindan al estudiante las herramientas necesarias para la aplicación de los principios y características en el diseño de sitios web y fortalecer en el educando las habilidades en el desarrollo de aplicaciones de este tipo.

La asignatura de Programación V inicia con el estudio de los conceptos y elementos que intervienen en la construcción de una aplicación web. Se estudia los lenguajes de la actualidad en este ámbito, como el lenguaje de marcado de hipertexto (HTML) y hoja de estilo en cascada (CSS), lenguaje de programación frontend (JavaScript) y backend (ASP.NET).

OBJETIVO

Crear aplicaciones web con características dinámicas y de acceso a datos, que utilicen los fundamentos conceptuales y técnicos de la programación web para dar solución a problemas cotidianos.

UNIDAD I: PROGRAMACIÓN WEB DEL LADO DEL CLIENTE



HTTP

El http (del inglés HyperText Transfer Protocol o Protocolo de Transferencia de Hiper Textos) es el protocolo de transmisión de información de la World Wide Web, es decir, el código que se establece para que el computador solicitante y el que contiene la información solicitada puedan “hablar” un mismo idioma

a la hora de transmitir información por la red.

Con el http se establecen criterios de sintaxis y semántica informática (forma y significado) para el establecimiento de la comunicación entre los diferentes elementos que constituyen la arquitectura web: servidores, clientes, proxies.

El funcionamiento del http se basa en un esquema de petición-respuesta entre el servidor web y el “agente usuario” (del inglés user agent) o cliente que realiza la solicitud de transmisión de datos. Un cliente puede ser un explorador determinado, cuando intentamos abrir una página web, o los rastreadores web (webcrawlers o arañas web) que las inspeccionan.

Por ejemplo: Al abrir una página web específica, el intercambio informativo entre nuestro explorador web y el servidor donde reside la información establecerá de qué manera debe transmitirse la información, en qué lugar están las imágenes y en qué orden se me mostrarán, etc. Este intercambio de comandos de solicitud y códigos de respuesta da como resultado la representación en mi computador de la misma información contenida originalmente en el servidor, que puede estar a miles de kilómetros de distancia.

https

Por https se entiende HyperText Transfer Protocol Secure o Protocolo Seguro de Transferencia de Hipertexto, que no es más que la versión segura del http, es decir, una variante del mismo protocolo que se basa en la creación de un canal cifrado para la transmisión de la información, lo cual lo hace más apropiado para ciertos datos de tipo sensible (como claves y usuarios personales).

A diferencia del http, el https está protegido contra la intervención de terceros que puedan espiar el intercambio de información o hacerse con datos del mismo, mediante el uso de “capas de red” que permiten sólo al servidor y al cliente cifrar y descifrar la información enviada mediante el intercambio previo de certificados de red, una suerte de convalidación inicial de confianza para establecer la transferencia de información.

Request

Es una solicitud o petición. Es un mensaje que un cliente, como un navegador web, envía a un servidor para solicitar un recurso específico, como una página web, una imagen, o cualquier otro tipo de datos. Los métodos más comunes son:

- **GET:** Solicita un recurso específico.
- **POST:** Envía datos al servidor, por ejemplo, al enviar un formulario.
- **PUT:** Actualiza un recurso existente o crea uno nuevo.
- **DELETE:** Elimina un recurso específico.

Códigos de estado HTTP

Los códigos de estado HTTP describen de forma abreviada la respuesta HTTP.

El primer dígito del código de estado especifica uno de los 5 tipos de respuesta, el mínimo para que un cliente pueda trabajar con HTTP es que reconozca estas 5 clases.

- 1XX Respuestas informativas
- 2XX Peticiones correctas
- 3XX Redirecciones
- 4XX Errores del cliente
- 5XX Errores de servidor

1XX Respuestas informativas

Este tipo de código de estado indica una respuesta provisional. HTTP/1.0 no definió ningún estado 1XX, por lo que los servidores no deberían enviar una respuesta 1XX a un cliente HTTP/1.0 salvo para experimentos.

- **100 Continue.** El servidor ha recibido los headers del request y el cliente debería proceder a enviar el cuerpo de la respuesta.
- **101 Switching Protocols.** El requester ha solicitado al servidor conmutar protocolos.
- **102 Processing.** Usado en requests para reanudar peticiones PUT o POST abortadas.

2XX Peticiones correctas

Este código de estado indica que la acción solicitada por el cliente ha sido recibida, entendida, aceptada y procesada correctamente.

- **200 OK.** El request es correcto. Esta es la respuesta estándar para respuestas correctas.
- **201 Created.** El request se ha completado y se ha creado un nuevo recurso.
- **202 Aceptada.** El request se ha aceptado para procesarlo, pero el proceso aún no ha terminado.
- **203 Non-Authoritative Information.** El request se ha procesado correctamente, pero devuelve información que podría venir de otra fuente.

- **204 No Content.** El request se ha procesado correctamente, pero no devuelve ningún contenido.
- **Otros: 205 Reset Content, 206 Partial Content, 207 Multi-Status**

3XX Redirecciones

El cliente ha de tomar una acción adicional para completar el request. Muchos de estos estados se utilizan para redirecciones. El user-agent puede llevar a cabo la acción adicional sin necesidad de que actúe el usuario sólo si el método utilizado en la segunda petición es GET o HEAD. Un user-agent no debería redireccionar automáticamente un request más de cinco veces, ya que esas redirecciones suelen indicar un infinite loop.

- **300 Multiple Choices.** Es una lista de enlaces. El usuario puede seleccionar un enlace e ir a esa dirección. Hay un máximo de cinco direcciones.
- **301 Moved Permanently.** La página solicitada se ha movido permanentemente a una nueva URI.
- **302 Found.** La página solicitada se ha movido temporalmente a una nueva URI.
- **303 See Other.** La página solicitada se puede encontrar en una URI diferente.
- **304 Not Modified.** Indica que la página solicitada no se ha modificado desde la última petición.
- **Otros: 305 Use Proxy, 307 Temporary Redirect, 308 Permanent Redirect**

4XX Errores del cliente

Excepto cuando se responde a un HEAD request, el servidor debe incluir una entidad que contiene una explicación del error, y si es temporal o permanente. Son aplicables a cualquier método de solicitud (GET, POST...). Los user agents deben mostrar cualquier entidad al usuario:

- **400 Bad Request.** El servidor no puede o no va a procesar el request por un error de sintaxis del cliente.
- **401 Unauthorized.** Similar al error 403, pero se usa cuando se requiere una autenticación y ha fallado o todavía no se a facilitado.
- **403 Forbidden.** El request fue válido pero el servidor se niega a responder.
- **404 Not Found.** El recurso del request no se ha podido encontrar pero podría estar disponible en el futuro.
- **405 Method Not Allowed.** Se ha hecho un request con un recurso usando un método request no soportado por ese recurso (por ejemplo usando GET en un formulario que requiere POST).
- **406 Not Acceptable.** El recurso solicitado solo genera contenido no aceptado de acuerdo con los headers Accept enviados en el request.
- **Otros: 402 Payment Required, 407 Proxy Authentication Required, 408 Request Timeout, 409 Conflict, 410 Gone.**

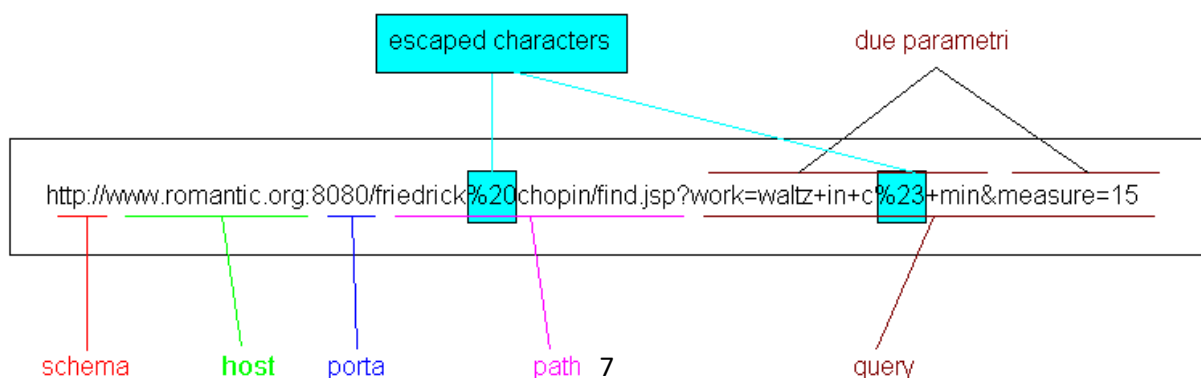
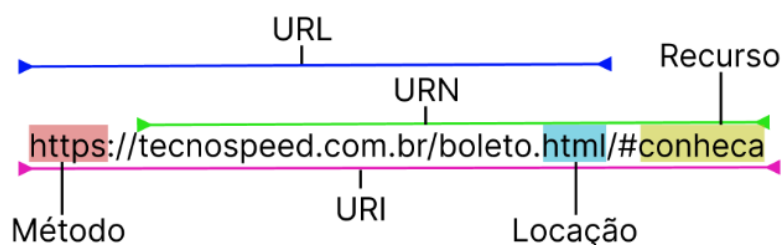
5XX Errores del servidor

El servidor ha fallado al completar una solicitud aparentemente válida. Cuando los códigos de estado empiezan por 5 indica casos en los que el servidor sabe que tiene un error o realmente es incapaz de procesar el request. Salvo cuando se trata de un request HEAD, el servidor debe incluir una entidad conteniendo una explicación del error, y de si es temporal o permanente. Igualmente los user-agents deberán mostrar cualquier entidad al usuario. Estos códigos de respuesta se aplican a cualquier método request.

- **500 Internal Server Error.** Error genérico, cuando se ha dado una condición no esperada y no se puede concretar el mensaje.
- **501 Not Implemented.** El servidor o no reconoce el método del request o carece de la capacidad para completarlo. Normalmente es algo que se ofrecerá en el futuro, como un nuevo servicio de una API.
- **502 Bad Gateway.** El server actuaba como puerta de entrada o proxy y recibió una respuesta inválida del servidor upstream.
- **503 Service Unavailable.** El servidor está actualmente no disponible, ya sea por mantenimiento o por sobrecarga.
- **504 Gateway Timeout.** El servidor estaba actuando como puerta de entrada o proxy y no recibió una respuesta oportuna por parte del servidor upstream.
- **505 HTTP Version Not Supported.** El servidor no soporta la versión del protocolo HTTP usada en el request.
- **511 Network Authentication Required** (RFC 6585). El cliente necesita autenticarse para poder acceder a la red.

URI

- **URI** Uniform Resource Identifier: identificador uniforme de recursos.
- **URL** Uniform Resource Locator: localizador uniforme de recursos.
- **URN** Uniform Resource Name: nombre uniforme de recursos.



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> y se pone el elemento <TITLE></TITLE> que es una breve descripción que identifica el documento. Es lo que veremos como 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 Html5

```
1. <!DOCTYPE html>
2. <html lang="es">
3.   <head>
4.     <meta charset="utf-8" />
5.     <title>Hola Mundo!</title>
6.   </head>
7.   <body>
8.     <h1>Hola Mundo!</h1>
9.   </body>
10. </html>
```

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
- : 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 hipervínculo o vínculo que direcciona a otro recurso o página web
-
: Crea un salto de línea en una 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.

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.

Características de Bootstrap

- **Fácil de usar:** cualquiera con conocimientos básicos de HTML y CSS puede comenzar a usar Bootstrap
- **Funciones de respuesta:** el CSS responsive 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

Empezar a trabajar con Bootstrap

Hay dos formas de comenzar a utilizar Bootstrap 4 en 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.esdocu.com/docs>

Descarga

Si desea descargar y alojar Bootstrap usted mismo, vaya a <https://getbootstrap.esdocu.com/docs> 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/
    ├── glyphs-halflings-regular.eot
    ├── glyphs-halflings-regular.svg
    ├── glyphs-halflings-regular.ttf
    ├── glyphs-halflings-regular.woff
    └── glyphs-halflings-regular.woff2
```

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.

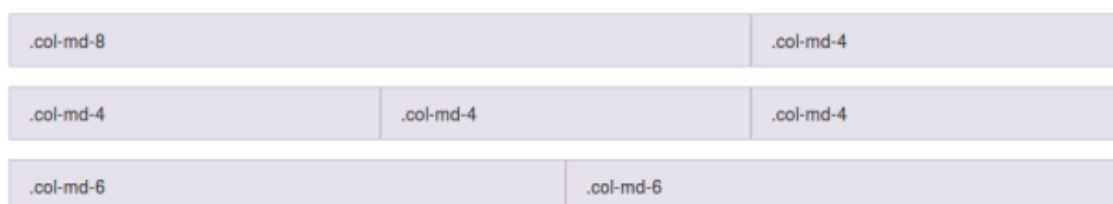


- El contenido se debe disponer dentro de columnas, las cuales serán los únicos hijos de las filas, y que a su vez serán los únicos hijos del contenedor
- Cada fila se puede dividir hasta un máximo de 12 columnas pero seremos nosotros los que tendremos que definir el número de columnas en el que queremos 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.



** Revise la documentación de Bootstrap en <https://getbootstrap.com/>

Botones

Bootstrap tiene una clase base `.btn` que configura estilos básicos como relleno y alineación de contenido



[Enlace](#)

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Tamaños

Botón grande **Botón grande**

HTML  

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```

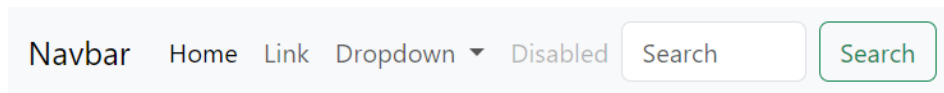
Botón pequeño **Botón pequeño**

HTML  

```
<button type="button" class="btn btn-primary btn-sm">Small button</button>
<button type="button" class="btn btn-secondary btn-sm">Small button</button>
```

Barra de navegación

Las barras de navegación requieren la clase `.navbar`



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
```

```
      <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="#">Action</a></li>
        <li><a class="dropdown-item" href="#">Another action</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item" href="#">Something else here</a></li>
      </ul>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" aria-disabled="true">Disabled</a>
    </li>
  </ul>
  <form class="d-flex" role="search">
    <input class="form-control me-2" type="search" placeholder="Search" aria-la
    <button class="btn btn-outline-success" type="submit">Search</button>
  </form>
</div>
</div>
</nav>
```

Alertas

Las alertas están disponibles para cualquier longitud de texto, así como un botón de cierre opcional.

```
<div class="alert alert-primary" role="alert">
  A simple primary alert with <a href="#" class="alert-link">an example link</a>. C
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert with <a href="#" class="alert-link">an example link</a>.
</div>
<div class="alert alert-success" role="alert">
  A simple success alert with <a href="#" class="alert-link">an example link</a>. C
</div>
```

Una alerta primaria simple: ¡compruébala!

Una alerta secundaria simple: ¡compruébala!

Una simple alerta de éxito: ¡compruébala!

Una alerta de peligro simple: ¡compruébala!

Una simple alerta de advertencia: ¡compruébala!

Una alerta de información simple: ¡compruébala!

Una simple alerta clara: ¡compruébalo!

Una simple alerta oscura: ¡compruébala!

Modales

Usa el complemento modal JavaScript de Bootstrap para agregar cuadros de diálogo a tu sitio para lightboxes, notificaciones de usuario o contenido completamente personalizado.

Título modal

El texto del cuerpo modal va aquí.

Cerrar Guardar cambios

```

<div class="modal" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal title</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label=
      </div>
      <div class="modal-body">
        <p>Modal body text goes here.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Clc
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>

```

Tablas

la clase para las tablas es **.table**

#	Primero	Último	Manejar
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

```

<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>

```

```

  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>

```



```

<tr>
  <th scope="row">2</th>
  <td>Jacob</td>
  <td>Thornton</td>
  <td>@fat</td>
</tr>
<tr>
  <th scope="row">3</th>
  <td colspan="2">Larry the Bird</td>
  <td>@twitter</td>
</tr>
</tbody>
</table>

```

Usa **.table-striped** para agregar rayas de cebra a cualquier fila de la tabla dentro del <tbody>.

#	Primero	Último	Manejar
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Usa **.table-striped-columns** para agregar rayas de cebra a cualquier columna de la tabla.

#	Primero	Último	Manejar
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Agrega **.table-hover** para habilitar un estado de desplazamiento en las filas de la tabla dentro de un <tbody>. Esta función permite resaltar una fila cuando el mouse pasa sobre esta.

```

<table class="table table-hover">
  ...
</table>

```

Agrega **.table-bordered** para los bordes en todos los lados de la tabla y las celdas.

Ejemplo:

#	Primero	Último	Manejar
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Agrega **.table-sm** para hacer cualquier **.table** más compacta cortando todo el padding de la celda por la mitad.

#	Primero	Último	Manejar
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Formularios

Otorga a controles de texto como `<input>`s y `<textarea>`s una actualización con estilos, tamaños, estados de enfoque personalizados y más con la clase **.form-control**

Ejemplo:

Dirección de correo electrónico

Ejemplo de área de texto

```
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Email address</label>
  <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
</div>
<div class="mb-3">
  <label for="exampleFormControlTextarea1" class="form-label">Example textarea</label>
  <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
</div>
```

Formularios en línea

Puede usar cualquier elemento HTML en línea típico (ya sea ``, `<small>`, o algo más) creando una fila y columnas.

Contraseña

Debe tener entre 8 y 20 caracteres.

```
<div class="row g-3 align-items-center">
  <div class="col-auto">
    <label for="inputPassword6" class="col-form-label">Password</label>
  </div>
  <div class="col-auto">
    <input type="password" id="inputPassword6" class="form-control" aria-describedby="passwordHelpInline">
  </div>
  <div class="col-auto">
    <span id="passwordHelpInline" class="form-text">
      Must be 8-20 characters long.
    </span>
  </div>
</div>
```

Select

Los menús <select> personalizados solo necesitan una clase personalizada, **.form-select**

Abre este menú de selección



```
<select class="form-select" aria-label="Default select example">
  <option selected>Open this select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
```

Check

Las casillas de verificación y radios predeterminadas del navegador se reemplazan con la ayuda de **.form-check**

☐ Casilla de verificación predeterminada

☒ Casilla marcada

```
<div class="form-check">
  <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
  <label class="form-check-label" for="flexCheckDefault">
    Default checkbox
  </label>
</div>
<div class="form-check">
  <input checked="" class="form-check-input" type="checkbox" value="" id="flexCheckChecked" class="form-check-input">
  <label class="form-check-label" for="flexCheckChecked">
    Checked checkbox
  </label>
</div>
```

Radios

- ☐ Radio predeterminada
- ☒ Radio marcada por defecto

```
<div class="form-check">
  <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadi
  <label class="form-check-label" for="flexRadioDefault1">
    Default radio
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadi
  <label class="form-check-label" for="flexRadioDefault2">
    Default checked radio
  </label>
</div>
```

Trabajo Autónomo

Diseñar una página web usando el framework Bootstrap y el sistema de rejillas.

RÚBRICA DE EVALUACIÓN		PUNTAJE
1	Crear tres filas de la siguiente manera Fila1: Una columna de 3, otra de 4 y otra de 5. Fila2: Una columna de tamaño 7 y una de 5. Fila3: Tres columnas del mismo tamaño.	2
2	La fila1 debe mostrar texto alternativo (lorem ipsum) en cada columna	2
3	La fila2 debe mostrar un un botón en cada columna de diferentes colores.	2
4	La fila3 debe mostrar una imagen en cada columna. (Imágenes accedidas de la web) La primera imagen debe tener borde con esquinas redondeadas La segunda imagen debe tener borde circular La tercera imagen debe tener borde externo "tumbail"	2
5	La página debe tener un navbar	2
TOTAL		10
Nota: El plagio o copia anulará su trabajo.		

Trabajo Autónomo

Realizar un formulario web aplicando Bootstrap, donde el usuario pueda ingresar la siguiente información:

Nombres, número de cédula, fecha de nacimiento, ciudad, correo, teléfono.

Escribir la dirección en un textarea

Seleccionar mediante una lista desplegable el estado civil.

Seleccionar mediante una lista desplegable la etnia.

Seleccionar mediante radio button el género.

Seleccionar mediante checkbox si tiene malos hábitos: fuma, toma alcohol, consume drogas.

RÚBRICA DE EVALUACIÓN		PUNTAJE
1	Utilizar inline forms de Bootstrap	2
1	Uso correcto de los tipos de input	2
2	Uso correcto de listas desplegables	2
3	Uso correcto de radio button y checkbox	2
5	Presentación del documento PDF. Incluye imágenes legibles y en buena calidad	2
TOTAL		10
Nota: El plagio o copia anulará su trabajo.		

INTRODUCCIÓN A 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 <https://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 como en los de Microsoft (lo que se conoce como content delivery network):

```
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/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:

```
$(document).ready(function() {
    // Código jQuery
});
```

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)

Manipulación DOM de jQuery

- **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.

Ejemplo:

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

- **attr ()** - El attr() método jQuery se usa para obtener valores de atributos.

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

Ejemplo:

```
$("#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);  
});
```

Revise la documentación de JQuery: <https://api.jquery.com/>

Para más información revise: <https://www.w3schools.com/jquery/default.asp>

UNIDAD 2. 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.

Lenguajes 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.

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.

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).

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP es un lenguaje de programación de servidor y una poderosa herramienta para crear páginas web dinámicas e interactivas. Es una alternativa eficaz, gratuita y ampliamente utilizada frente a competidores como ASP de Microsoft.

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.

Sintaxis básica de PHP

Se puede colocar un script PHP en cualquier parte del documento.

Un script PHP comienza `<?php` y termina con `?>` :

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

- La extensión de archivo predeterminada para archivos PHP es ".php"
- las declaraciones PHP terminan con un punto y coma (;).

Ejemplo de un archivo simple .php con código HTML y código PHP:

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

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

</body>
</html>
```

Declaraciones echo y print

Con PHP, hay dos formas básicas de obtener resultados: echo y print. Echo y print son más o menos iguales. Ambos se utilizan para enviar datos a la pantalla.

Las diferencias son pequeñas: echo no tiene valor de retorno mientras que print tiene un valor de retorno de 1, por lo que puede usarse en expresiones. Echo puede tomar múltiples parámetros (aunque dicho uso es raro) mientras que print puede tomar un argumento. Echo es algo más rápido que print.

El siguiente ejemplo muestra cómo generar texto con el echo comando (tenga en cuenta que el texto puede contener HTML):

```
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
```

Variables PHP

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

```
$x = 5;
$y = "John"
```

Reglas para variables PHP:

- Una variable comienza con el \$signo, seguido del nombre de la variable.
- El nombre de una variable debe comenzar con una letra o un guion bajo
- El nombre de una variable no puede comenzar con un número
- El nombre de una variable solo puede contener caracteres alfanuméricos y guiones bajos (Az, 0-9 y _).
- Los nombres de las variables distinguen entre mayúsculas y minúsculas (\$age y \$AGE son dos variables diferentes)

Tipos de variables

PHP es un lenguaje poco tipificado; no tiene ningún comando para declarar una variable y el tipo de datos depende del valor de la variable.

Ejemplo

```
$x = 5;          // $x is an integer
$y = "John";    // $y is a string
echo $x;
echo $y;
```

Comentarios en PHP

PHP admite varias formas de comentar:

Sintaxis de comentarios en código PHP:

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

# Este también es un comentario de una línea

/* Este es un
comentario multilínea */
```

Concatenar strings

Las comillas dobles permiten insertar variables. Ejemplo

```
$x = "John";
echo "Hello $x";
```


Para concatenar o combinar dos cadenas puede utilizar el operador punto (.)

Ejemplo

```
$x = "Hello";  
$y = "World";  
$z = $x . $y;  
echo $z;
```

El resultado del ejemplo anterior es HelloWorld, sin espacio entre las dos palabras.

Puedes agregar un carácter de espacio como este:

```
$x = "Hello";  
$y = "World";  
$z = $x . " " . $y;  
echo $z;
```

Una forma mejor y más fácil es utilizar el poder de las comillas dobles.

```
$x = "Hello";  
$y = "World";  
$z = "$x $y";  
echo $z;
```

Operadores de comparación PHP

Los operadores de comparación de PHP se utilizan para comparar dos valores

Operator	Name	Example
==	Equal	\$x == \$y
===	Identical	\$x === \$y
!=	Not equal	\$x != \$y
<>	Not equal	\$x <> \$y
!==	Not identical	\$x !== \$y
>	Greater than	\$x > \$y
<	Less than	\$x < \$y

DECLARACIONES CONDICIONALES DE PHP

IF ELSE

```
if (condition) {  
    //code to be executed if this condition is true;  
} elseif (condition) {  
    // code to be executed if first condition is false and this condition is true;  
} else {  
    // code to be executed if all conditions are false;  
}
```

SWITCH

```
switch (expression) {  
    case label1:  
        //code block  
        break;  
    case label2:  
        //code block;  
        break;  
    case label3:  
        //code block  
        break;  
    default:  
        //code block  
}
```

WHILE

```
$i = 1;  
while ($i < 6) {  
    echo $i;  
    $i++;  
}
```

DO WHILE

```
$i = 1;  
  
do {  
    echo $i;  
    $i++;  
} while ($i < 6);
```

FOR

```
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}
```

FOREACH

```
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $x) {  
    echo "$x <br>";  
}
```

Funciones

Una declaración de función definida por el usuario comienza con la palabra clave function, seguida del nombre de la función:

Para llamar a la función, simplemente escriba su nombre seguido de paréntesis ():

Ejemplo:

```
function myMessage() {  
    echo "Hello world!";  
}  
  
myMessage();
```

Arrays

```
$cars = array("Volvo", "BMW", "Toyota");
```

Declaración Include

La declaración include(o require) toma todo el texto/código/marcado que existe en el archivo especificado y lo copia en el archivo que usa la declaración de inclusión. Incluir archivos es muy útil cuando desea incluir el mismo PHP, HTML o texto en varias páginas de un sitio web.

Las declaraciones include y require son idénticas, excepto en caso de error:

- **require** producirá un error fatal (E_COMPILE_ERROR) y detendrá el script
- **include** solo producirá una advertencia (E_WARNING) y el script continuará

Sintaxis

```
include 'filename';
```

or

```
require 'filename';
```

```
<?php include 'footer.php';?>
```

Formularios PHP

Los superglobales \$_GET y \$_POST se utilizan para recopilar datos de formularios.

- **\$_GET** es un vector de variables pasadas al script actual a través de los parámetros de URL.
- **\$_POST** es un vector de variables pasadas al script actual mediante el método HTTP POST de manera invisible u oculta para los usuarios.

Ejemplo

```
<form action="welcome.php" method="POST">  
  Name: <input type="text" name="name"><br>  
  E-mail: <input type="text" name="email"><br>  
  <input type="submit">  
</form>
```

Cuando el usuario completa el formulario anterior y hace clic en el botón enviar, los datos del formulario se envían para su procesamiento a un archivo PHP llamado "welcome.php". Los datos del formulario se envían con el método HTTP POST.

Para mostrar los datos enviados, simplemente puede repetir todas las variables.

Con el método POST

```
Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>
```

Con el método GET

```
Welcome <?php echo $_GET["name"]; ?><br>
Your email address is: <?php echo $_GET["email"]; ?>
```

Ingresar variable php en caja de texto

```
<input type="text" name="name" value="<?php echo $name;?>">
```

Sesiones en PHP

Con la función `session_start()` se inicia una sesión

Las variables de sesión se configuran con la variable global de PHP: `$_SESSION`.

La función `session_start()` debe ser lo primero en su documento. Antes de cualquier etiqueta HTML. Ejemplo:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

Obtener valores de variables de sesión PHP

Tenga en cuenta que las variables de sesión no se pasan individualmente a cada página nueva, sino que se recuperan de la sesión que abrimos al principio de cada página (`session_start()`).

Observe también que todos los valores de las variables de sesión se almacenan en la variable global `$_SESSION`:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

Destruir una sesión PHP

Para eliminar todas las variables de sesión globales y destruir la sesión, use `session_unset()` y `session_destroy()`:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

Actividad Práctica 4

Crear una aplicación web con conexión a base de datos

VUE.JS

Vue (pronunciado /vju:/, como view) es un framework de JavaScript para crear interfaces de usuario. Está construido sobre los estándares de HTML, CSS y JavaScript, y proporciona un modelo de programación declarativo y basado en componentes que te ayuda a desarrollar interfaces de usuario de manera más eficiente, ya sean simples o complejas.

Aquí tienes un pequeño ejemplo:

```
import { createApp, ref } from 'vue'

createApp({
  setup() {
    return {
      count: ref(0)
    }
  }
}).mount('#app')
```

```
<div id="app">
  <button @click="count++">
    El contador está en: {{ count }}
  </button>
</div>
```

Resultado

El contador está en: 0

El ejemplo anterior demuestra las dos características principales de Vue:

- **El Renderizado Declarativo:** Vue amplía el HTML estándar con una sintaxis de plantilla que nos permite describir de forma declarativa la salida HTML en función al estado de JavaScript.
- **Reactividad:** Vue rastrea automáticamente los cambios de estado de JavaScript y actualiza de manera eficiente el DOM cuando ocurren cambios.

Técnicamente, Vue.js se centra en la capa ViewModel del patrón MVVM. Conecta la Vista y el Modelo a través de enlaces de datos bidireccionales. Las manipulaciones DOM reales y el formato de salida se abstraen en Directivas y Filtros .

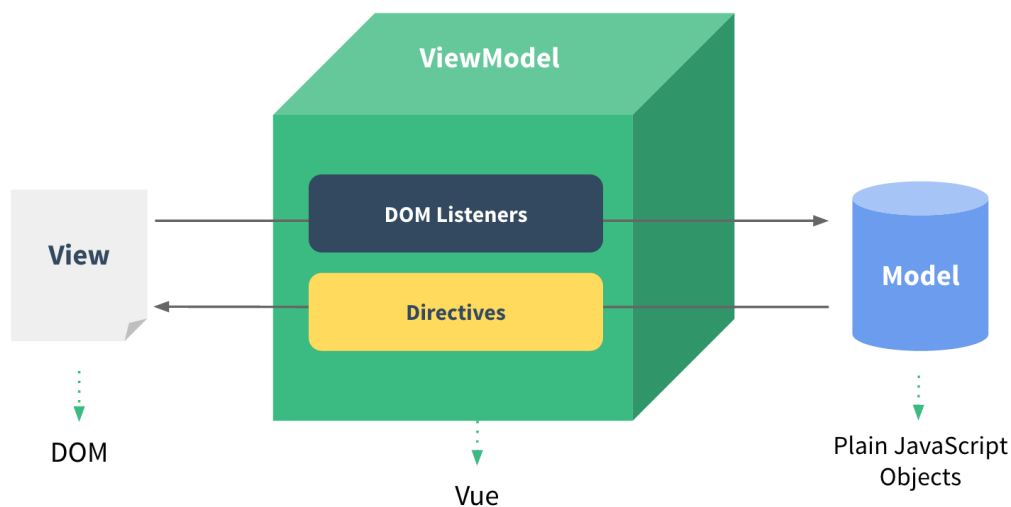
Filosóficamente, el objetivo es proporcionar los beneficios del enlace de datos reactivo y los componentes de vista componibles con una API que sea lo más simple posible. No es un marco completo: está diseñado para ser una capa de vista simple y flexible. Puede usarlo solo para crear prototipos rápidamente o mezclarlo y combinarlo con otras bibliotecas para obtener una pila de interfaz de usuario personalizada. También es una opción natural para servicios sin backend como Firebase.

El Framework Progresivo

Vue es un framework y un ecosistema que cubre la mayoría de las características comunes necesarias en el desarrollo frontend. Pero la web es extremadamente diversa: las cosas que construimos en la web pueden variar drásticamente en forma y escala. Con eso en mente, Vue está diseñado para ser flexible y ser adoptado de forma incremental. Dependiendo de tu caso de uso, Vue se puede usar de diferentes maneras:

- Mejorar el HTML estático sin un paso de compilación
- Incrustar componentes web en cualquier página
- Aplicaciones de una sola página (SPA)
- Fullstack / Renderizado del lado del servidor (SSR)
- Jamstack / Generación de sitios estáticos (SSG)
- Orientación hacia aplicaciones de escritorio, móvil, WebGL o incluso de la terminal.

Descripción general de conceptos



Mira la documentación de vue.js en: <https://vuejs.org/guide/introduction.html>

MVVM

El patrón modelo–vista–modelo de vista (en inglés: model–view–viewmodel, abreviado MVVM) es un patrón de arquitectura de software. Se caracteriza por tratar de desacoplar lo máximo posible la interfaz de usuario de la lógica de la aplicación.

- **El modelo**

Representa la capa de datos y/o la lógica de negocio, también denominado como el objeto del dominio. El modelo contiene la información, pero nunca las acciones o servicios que la manipulan. En ningún caso tiene dependencia alguna con la vista.

- **La vista**

La misión de la vista es representar la información a través de los elementos visuales que la componen. Las vistas en MVVM son activas, contienen comportamientos, eventos y enlaces a datos que, en cierta manera, necesitan tener conocimiento del modelo subyacente.

- **Modelo de vista**

El modelo de vista es un actor intermediario entre el modelo y la vista, contiene toda la lógica de presentación y se comporta como una abstracción de la interfaz. La comunicación entre la vista y el modelo de vista se realiza por medio los enlaces de datos.

AXIOS

Axios es un Cliente HTTP basado en promesas para node.js y el navegador. Es isomorfo (= puede ejecutarse en el navegador y nodejs con el mismo código base). En el lado del servidor usa el modulo nativo http de node.js, mientras que en el lado del cliente (navegador) usa XMLHttpRequests.

Características

- Hace XMLHttpRequests desde el navegador
- Hace peticiones http desde node.js
- Soporta el API de Promesa
- Intercepta petición y respuesta
- Transforma petición y datos de respuesta
- Cancela peticiones
- Transformación automática de datos JSON
- Manejo automático de datos JSON en respuesta
- Captura de progreso para navegadores y node.js con información adicional (velocidad, tiempo restante)
- Establecer límites de ancho de banda para node.js
- Soporte para proteger al cliente contra XSRF

PHP PDO

La extensión PDO (PHP Data Objects) permite acceder a distintas bases de datos (MySQL/MariaBD, PostgreSQL, Oracle, MS SQL Server, SQLite, Firebird, DB2, Informix, etc.) de una forma uniforme, lo que facilita la portabilidad de las aplicaciones. La portabilidad entre bases de datos no es absoluta debido por ejemplo a las diferencias entre diferentes bases de datos en aspectos como la forma de conectarse o la sintaxis del lenguaje de definición de datos. Para ilustrar esas diferencias y como tratarlas adecuadamente, se ofrecen ejemplos de código específicos para SQLite y para MySQL.

La extensión PDO no evalúa la corrección de las consultas SQL, aunque sí implementa algunas medidas de seguridad mediante consultas preparadas, como se comenta en el apartado correspondiente.

Ventajas

PDO ofrece las siguientes características funcionales:

- Soporta muchas bases de datos
- No hay que reescribir el código si se cambia de base de datos
- Velocidad
- Facilita la instalación
- Manejo de conexiones con los diferentes proveedores de base de datos
- Manejo de los errores producidos en la conexión
- Cierre y persistencias de las conexiones
- Manejo de las transacciones, permitiendo el autocommit
- Inserciones de procedimientos e instrucciones preparadas
- Manejo de errores y avisos

Conexión a base de datos

Las conexiones se establecen creando instancias de la clase base PDO. No importa el controlador que se utilice; siempre se usará el nombre de la clase PDO. El constructor acepta parámetros para especificar el origen de la base de datos (conocido como DSN) y, opcionalmente, el nombre de usuario y la contraseña (si la hubiera).

Ejemplo:

```
<?php
$mbd = new PDO('mysql:host=localhost;dbname=prueba', $usuario, $contraseña);
?>
```

Bibliografía

Básica

- 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>

Especializada

- 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>

Complementaria

- 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>