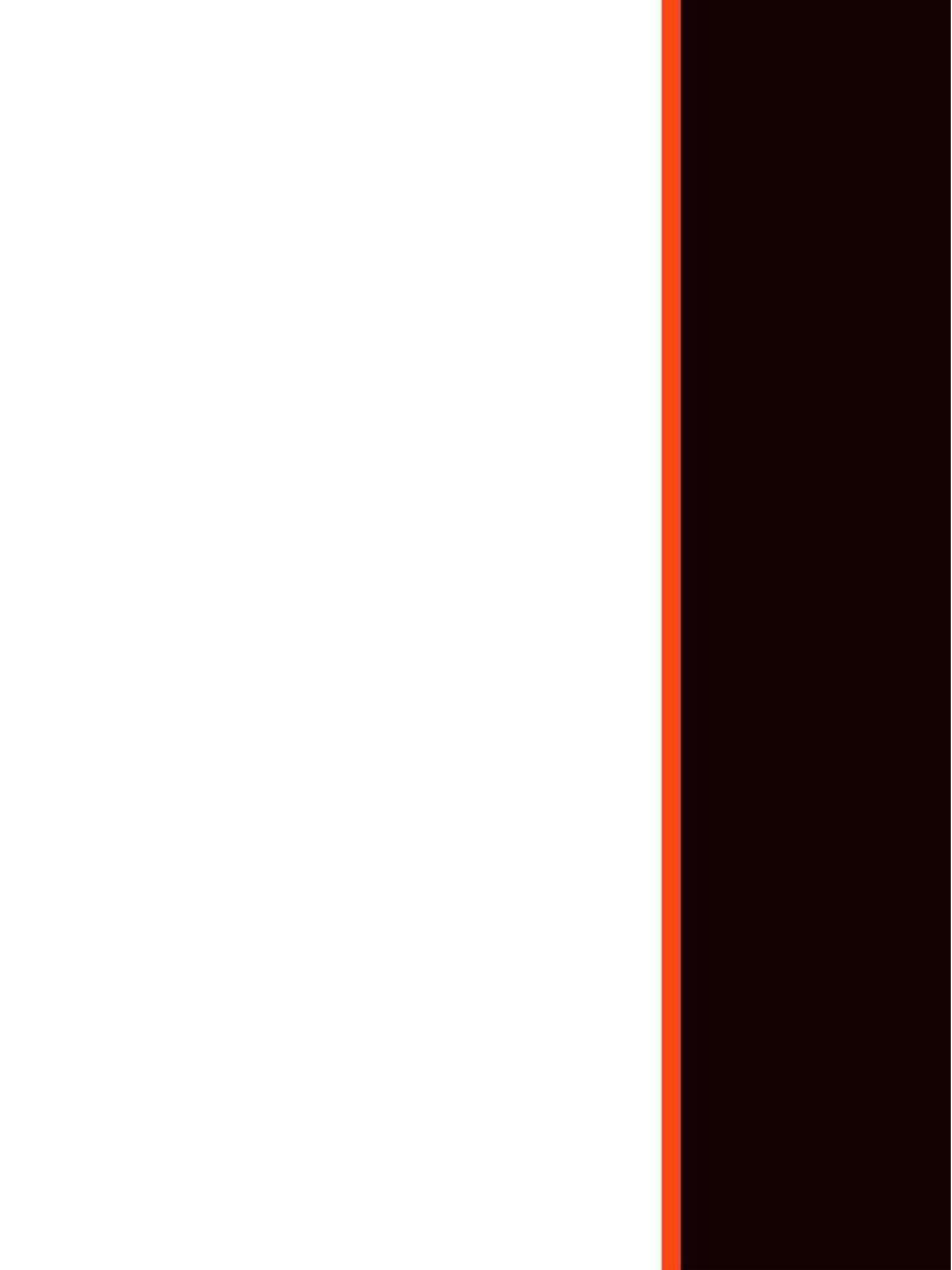


DESARROLLO WEB



CARRERA: ANÁLISIS DE SISTEMAS

SEMESTRE: CUARTO



INTRODUCCIÓN

A lo largo de este viaje, te sumergirás en los lenguajes básicos que conforman la estructura y el dinamismo de las páginas web: HTML, CSS y JavaScript. Conocerás los principios básicos de PHP y MySQL, creando sitios web dinámicos que interactúan con bases de datos.

Finalmente, pondrás en práctica tus conocimientos desarrollando un sistema de autenticación de usuarios, aplicando tus habilidades. Al finalizar este ebook, habrás adquirido las bases sólidas para desenvolverte en el mundo del desarrollo web, permitiéndote crear tus propios sitios web básicos, interactivos y dinámicos.

ELÍAS VARGAS

- Ingeniero en Informática y Docente Asistente -



Una publicación de



PROGRAMA

Materia: Desarrollo Web

1. Lenguaje HTML y CSS.
 - Sintaxis básica.
 - Maquetación básica y estilos.
 - Formularios.
2. Lenguaje de programación Javascript.
 - Sintaxis básica.
 - Eventos, validaciones.
3. Aplicaciones web.
 - Introducción a PHP y MySQL.
 - Conexión PHP y MySQL.
4. Desarrollo de Aplicaciones web.
 - Sistema de autenticación de usuarios.

ÍNDICE

UNIDAD I	8
LENGUAJE HTML Y CSS	9
Lenguaje HTML	9
Etiquetas HTML.....	10
Sintaxis básica de un documento HTML.....	11
CSS (CASCADING STYLE SHEETS)	15
Reglas CSS.....	16
Propiedades CSS.....	17
Propiedades CSS-BOX Model.....	18
Aplicar estilos a la maqueta.....	19
Estructura básica de un formulario.....	22
RECURSOS INTERACTIVOS	24
UNIDAD II	25
LENGUAJE DE PROGRAMACIÓN JAVASCRIPT	26
Tipos de datos.....	28
EVENTOS EN JAVASCRIPT.....	29
MANEJO DE EVENTOS.....	32
RETO UNIDAD 2	34
RECURSOS INTERACTIVOS	35
UNIDAD III	36
APLICACIONES WEB	37
¿COMO FUNCIONAN?	38
¿QUÉ ES BACK-END Y FRONT END?.....	39

¿QUÉ ES EL FRONT END?	40
¿QUÉ ES EL BACK END?	41
TECNOLOGÍAS USADAS	42
INTRODUCCIÓN A PHP Y MYSQL.....	43
PASOS BÁSICOS PARA CREAR UNA APLICACIÓN CON PHP y MySQL....	44
INTRODUCCIÓN A PHP	47
TIPOS DE DATOS EN PHP.....	49
ENVÍO DE DATOS AL SERVIDOR	51
INTRODUCCIÓN A MYSQL	55
CREACIÓN DE UNA BASE DE DATOS Y TABLAS	58
CONEXIÓN A MYSQL DESDE PHP	59
RETO UNIDAD 3 - PHP	60
RETO UNIDAD 3 - MYSQL	61
RECURSOS INTERACTIVOS	62
UNIDAD IV	63
DESARROLLO DE APLICACIONES WEB	63
SISTEMA DE AUTENTICACIÓN DE USUARIOS.....	64
¿Cómo funciona?.....	64
FUNCIONAMIENTO DEL SISTEMA DE AUTENTICACIÓN	65
CONEXIÓN CON LA BASE DE DATOS EN MYSQL.....	68
CÓDIGO DE CREACIÓN DEL FORMULARIO LOGIN	70
CÓDIGO DEL SCRIPT VALIDAR (LADO DEL SERVIDOR).....	71
CÓDIGO DE LA PÁGINA PRINCIPAL	72
CÓDIGO DEL SCRIPT PARA CERRAR SESIÓN (LADO DEL SERVIDOR)	73
PRUEBA CON USUARIO NO REGISTRADO	74
PRUEBAS DE INICIO DE SESIÓN	75
PRUEBA CON CREDENCIALES CORRECTOS.....	76
ACCESO A LA PÁGINA PRINCIPAL DE LA APLICACIÓN.....	77

RETO UNIDAD 4	78
RECURSOS INTERACTIVOS	80
REFERENCIAS CONSULTADAS	81

UNIDAD I

LENGUAJE HTML Y CSS

Sintaxis - etiquetas - maquetación -
formularios - estilos - atributos - propiedades.

LENGUAJE HTML Y CSS

Lenguaje HTML

El **Lenguaje de Marcado de Hipertexto (HTML)** es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. HTML nos provee **etiquetas** para describir los diferentes tipos de contenidos (elementos) de nuestra web.

Gracias a ello, el navegador podrá comprender el contenido enviado por el servidor y representarlo en pantalla.

Figura 1

Logotipo de HTML5



Fuente: De W3C, CC BY 3.0,

<https://commons.wikimedia.org/w/index.php?curid=12736763>

Etiquetas HTML

Las etiquetas HTML5 son como las palabras de un idioma con las que construimos las páginas web. Son elementos que definen la estructura y el contenido de un documento HTML.

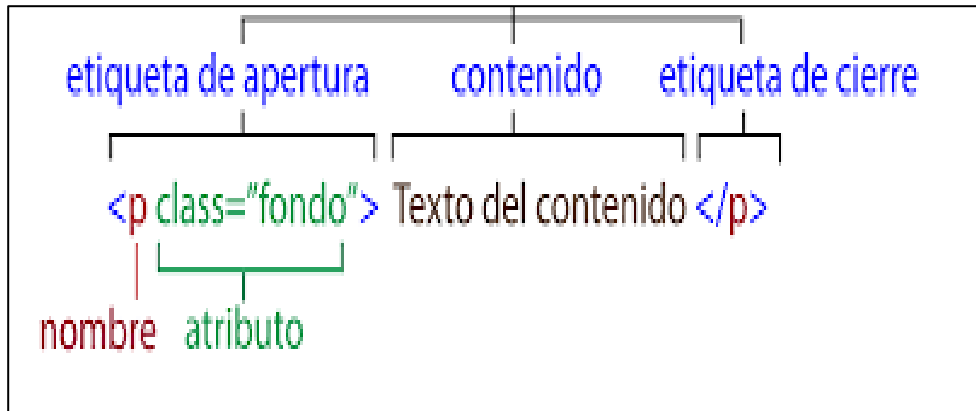
Cada etiqueta tiene una función específica y se escribe entre corchetes angulares (< y >). La estructura de las etiquetas HTML tiene el siguiente formato:

Figura 2
Estructura de una etiqueta HTML5



Fuente: <https://lenguajehtml.com/html/introduccion/estructura-etiqueta-html/>

Figura 3
Ejemplo de una etiquetaHTML5



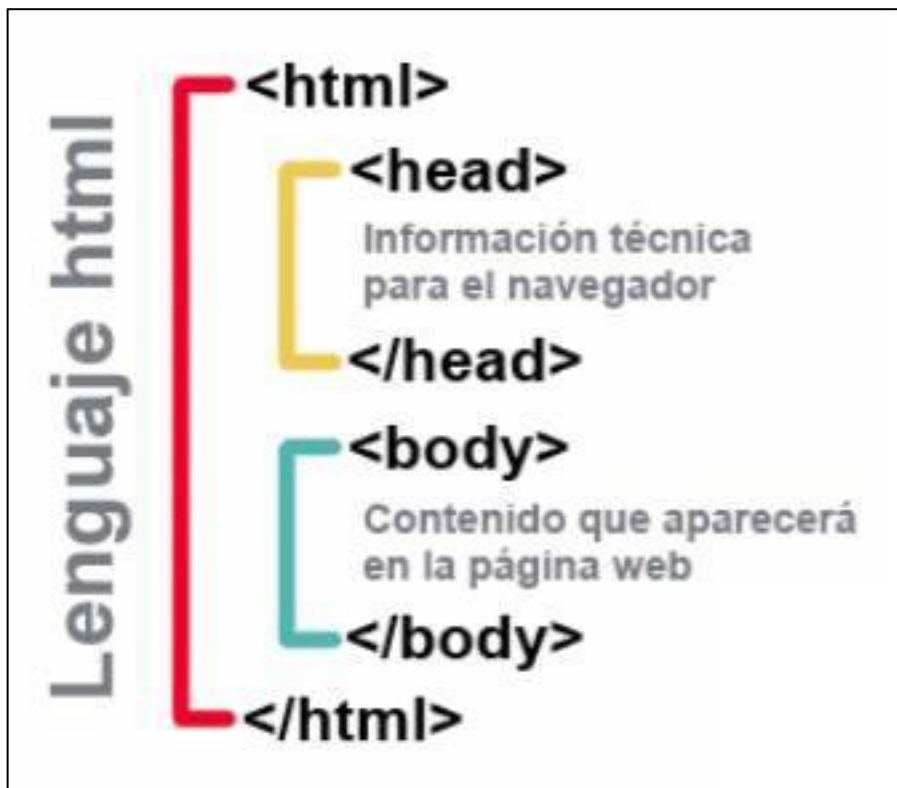
Fuente: <https://lenguajehtml.com/html/introduccion/estructura-etiqueta-html/>

Sintaxis básica de un documento HTML

Un documento HTML básico consta de las siguientes partes:

- **Doctype:** Indica al navegador el tipo de documento HTML que se está utilizando.
- **<html>:** Raíz de todo el documento
- **<head>:** Contiene metadatos sobre el documento.
- **<title>:** el título que se muestra en la pestaña del navegador.
- **<body>:** Contiene el contenido visible de la página.

Figura 4
Estructura básica de una página HTML



Fuente:

https://3con14.biz/i2016/_data/difusion_contenidos/esquema_pag_web.png

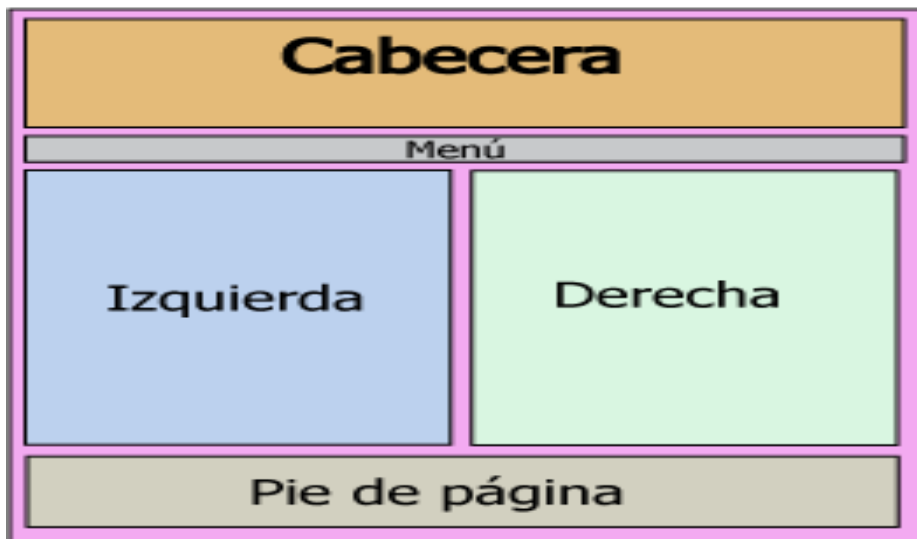
Maquetación en HTML

La **maquetación** en HTML5 consiste en utilizar las etiquetas y elementos proporcionados por este lenguaje para organizar y dar forma a los diferentes componentes de una página web.

Estas etiquetas actúan como contenedores semánticos, lo que significa que no solo definen la apariencia visual, sino también el significado del contenido.

Figura 5

Maquetación web (boceto de una página).



Fuente: Maquetación web. (.org/maquetacions. f.)
<https://evntossv.neocities>

Convertir el boceto a HTML

La imagen presenta un diseño bastante común en páginas web:

- **Cabecera:** Contiene el título principal de la página.
- **Menú:** Un área para la navegación.
- **Sección principal:** dos columnas: izquierda y derecha.
- **Pie de página:** información adicional como copyright.

Figura 6

Código HTML para maquetar la página.

```
index.html x
1
2 <!DOCTYPE html>
3 <html>
4 <head>
5   <title>Diseño Básico</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <div class="container">
10     <header>Cabecera</header>
11     <nav>Menú</nav>
12     <main>
13       <section class="left">Izquierda</section>
14       <section class="right">Derecha</section>
15     </main>
16     <footer>Pie de página</footer>
17   </div>
18 </body>
19 </html>
20
```

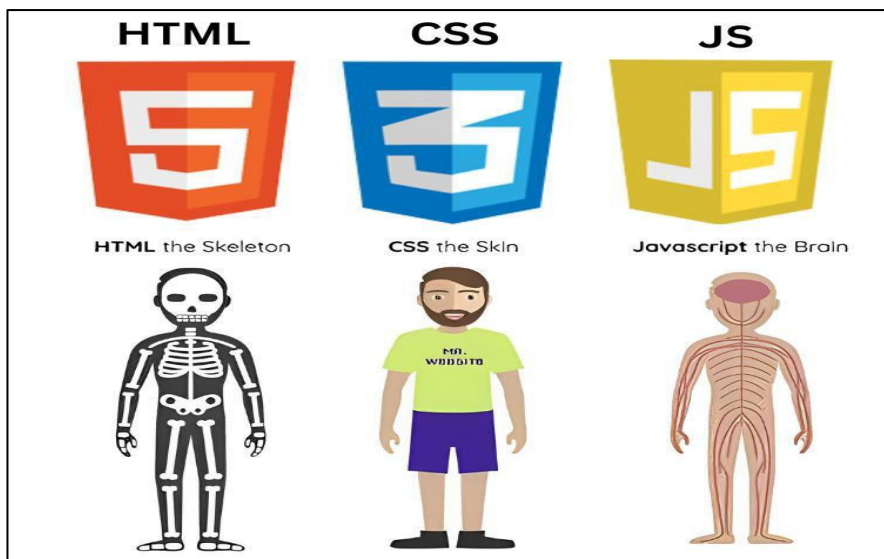
Fuente: Elaboración propia (2024).

CSS (CASCADING STYLE SHEETS)

CSS (Cascading Style Sheets, o Hojas de Estilo en Cascada) es el lenguaje que utilizamos para darle estilo y forma a las páginas web. Si HTML es el esqueleto, CSS es la ropa que viste ese esqueleto, definiendo colores, fuentes, tamaños, disposición y mucho más.

Figura 7

Logotipo de HTML5



Fuente: <https://www.linkedin.com/pulse/fundamental-elements-web-development-html-css-anass-essadikine-byn7e/>

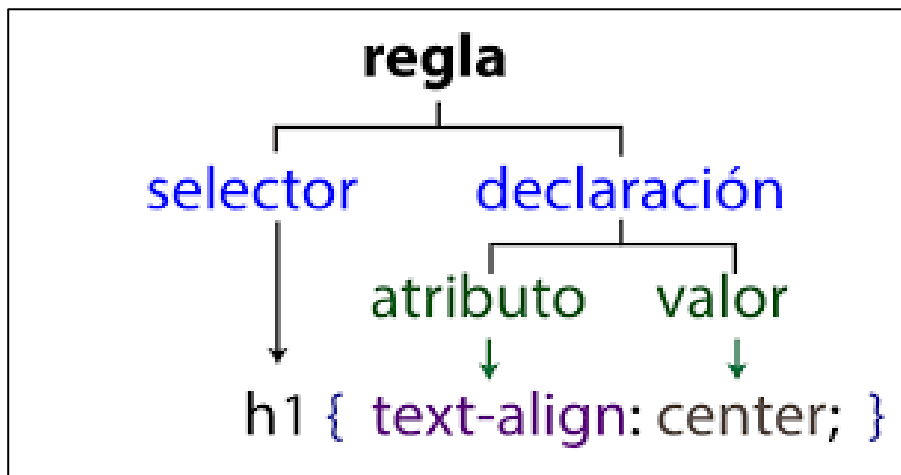
Reglas CSS

CSS funciona mediante **reglas de estilo**. Cada regla consta de un selector y una declaración.

- **Selector:** Indica a qué elemento HTML se aplicará la regla. Puede ser un elemento específico (por ejemplo, h1), una clase (.miClase) o un ID (#mild).
- **Declaración:** Define la propiedad y el valor que se asignará al elemento seleccionado. Por ejemplo, **color: blue;** establece el color del texto a azul.

Figura 8

Ejemplo de Sintaxis CSS3



Fuente: https://www.aulaclie.es/html/t_9_1.htm

Propiedades CSS

Son características que se asignan a los elementos HTML para modificar su estilo visual. Por ejemplo, puedes cambiar el color de fondo, el tamaño de la fuente, el margen, la forma, entre muchas otras cosas.

No vamos a listar todos de forma exhaustiva, pero sí vamos a presentar algunas destacados de uso habitual.

- Fuentes y texto.
- Modelo de cajas (Box model).
- Fondos, colores.
- Unidades de medida.

Figura 9

Propiedades CSS3 para Tipografías y fondos

Fuentes	Texto
font-family: tipos (<i>arial,verdana,...</i>). <i>Se puede especificar una lista de opciones</i> font-weight: grosor (<i>negrita,...</i>) font-size: tamaño font-style: estilo (<i>itálica,..</i>)	color: color text-decoration: efectos (<i>subrayado,...</i>) text-align: alineación line-height: interlineado text-indent: tabulación letter-spacing: espaciado entre letras
Listas	Fondo
list-style-type: tipo de viñeta (<i>círculo, cuadrado, número, letra...</i>) list-style-image: utilizar imágenes en las viñetas list-style-position: alineación del texto con la viñeta	background-color: color background-image: poner una imagen background-repeat: cómo se repite la imagen de fondo (<i>vertical, horizontal,..</i>) background-position: posición de la imagen de fondo

Fuente: <https://www.digitallearning.es/curso-wordpress/css-lenguaje-de-estilos.html>

Propiedades CSS-BOX Model

Otra función muy importante de los estilos, es aplicar dimensiones y posición a los elementos HTML, basándose en el denominado **Modelo de Cajas** (en inglés, Box Model) uno de los conceptos fundamentales de CSS.

Según este modelo, los navegadores ven o tratan cada elemento en la página HTML como una caja rectangular que tiene una serie de propiedades o características que vamos a ver a continuación.

Figura 10

Propiedades CSS3 para Box Model

Contenido	Borde
width: anchura height: altura line-height: distancia entre líneas overflow: qué presentar cuando el contenido excede el espacio asignado (<i>scroll</i> , <i>ocultar</i> , <i>mostrar...</i>)	border-width: anchura border-style: tipo (<i>línea</i> , <i>puntos</i> , <i>guiónes...</i>) border-color: color border-radius: redondeo de las esquinas
Padding y Márgenes	
padding: anchura de la zona de relleno margin: anchura de los márgenes <i>Pueden aplicarse valores distintos a cada lado: arriba (top), derecha (right), abajo (bottom), izquierda (left)</i> <i>Ejemplos: margin-top padding-right border-width-bottom</i>	

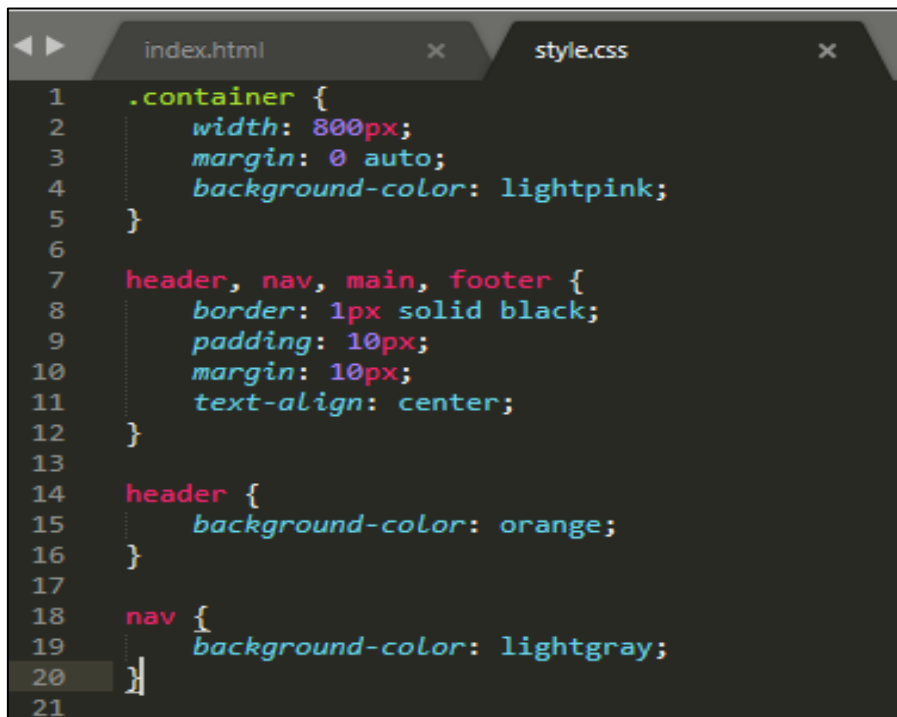
Fuente: <https://www.digitallearning.es/curso-wordpress/css-lenguaje-de-estilos.html>

Aplicar estilos a la maqueta

Crearemos un archivo llamado `style.css` para personalizar cada elemento de la página. Esto permite mantener el código HTML limpio y fácil de leer, separando la estructura del contenido de su presentación visual.

Figura 11

Código CSS para maquetar la página.

A screenshot of a code editor with two tabs: 'index.html' and 'style.css'. The 'style.css' tab is active and shows the following CSS code:

```
1  .container {
2      width: 800px;
3      margin: 0 auto;
4      background-color: lightpink;
5  }
6
7  header, nav, main, footer {
8      border: 1px solid black;
9      padding: 10px;
10     margin: 10px;
11     text-align: center;
12 }
13
14 header {
15     background-color: orange;
16 }
17
18 nav {
19     background-color: lightgray;
20 }
21
```

Fuente: Elaboración propia (2024)

.container: Establece el ancho máximo del contenedor y lo centra en la página.

header, nav, main, footer: Aplica estilos como bordes y relleno.

main: Utiliza `display: flex` para crear un layout de flexbox, permitiendo distribuir el contenido de manera flexible en las secciones izquierda y derecha.

.left, .right: Ocupan el mismo espacio (`flex: 1`) dentro del contenedor principal.

El resto de las clases aplican los colores de fondo a cada sección.

Figura 12

Código CSS para maquetar la página.

```
21
22  main {
23    height: 400px;
24    display: flex;
25  }
26
27  .left, .right {
28    flex: 1;
29  }
30
31  .left {
32    background-color: lightblue;
33  }
34
35  .right {
36    background-color: lightgreen;
37  }
38
39
40  footer {
41    background-color: lightgray;
42  }
43
```

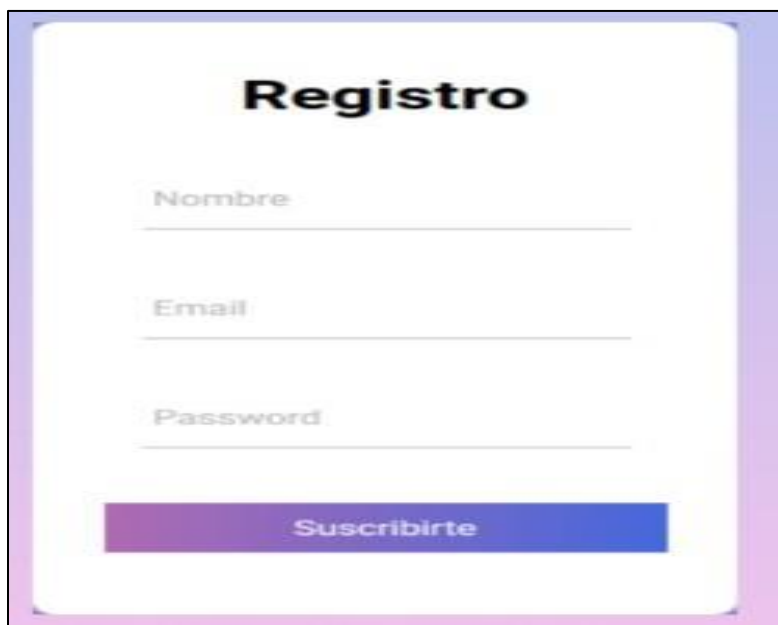
Fuente: Elaboración propia (2024).

FORMULARIOS WEB

Un formulario web o formulario HTML, **es un elemento principal del sitio web** que permite a los usuarios ingresar información y enviarla a un servidor para su procesamiento.

Figura 13

Ejemplo de formulario web de suscripción.

Un formulario web de suscripción con el título "Registro" en el centro. Debajo del título hay tres campos de entrada de texto etiquetados "Nombre", "Email" y "Password". En la parte inferior del formulario hay un botón rectangular azul con el texto "Suscribirse" en blanco.

Fuente: <https://alexcgdesign.com/blog/crear-un-formulario-de-registro-en-html>

Estructura básica de un formulario

Un formulario HTML se define con la etiqueta **<form>**. Dentro de esta etiqueta, se colocan los diferentes elementos de entrada, como campos de texto, botones, casillas de verificación, entre otros.

Figura 14

Código HTML para la creación de un formulario web.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Formularios</title>
5   <link rel="stylesheet" href="style.css">
6 </head>
7 <body>
8   <!--
9     action: Especifica la URL a la que se enviarán los datos del formulario cuando se envíe.
10    method: Indica el método HTTP utilizado para enviar los datos (GET o POST).
11  -->
12  <form action="procesar_formulario.php" method="post">
13    <!-- Asocia una etiqueta con un elemento de entrada para mejorar la accesibilidad.-->
14    <label for="nombre">Nombre:</label>
15
16    <!-- input: Define el tipo de campo de entrada (texto, correo electrónico, contraseña, etc.).-->
17    <input type="text" id="nombre" name="nombre">
18
19    <!-- button: Crea un botón para enviar el formulario. s-->
20    <button type="submit">Enviar</button>
21
22  </form>
23
24 </body>
25 </html>
```

Fuente: Elaboración propia (2024)

RETO UNIDAD 1

A continuación, se presenta la conceptualización inicial (boceto) de una página web. El reto consiste en maquetar con HTML y CSS. Hasta lograr el aspecto y estructura del boceto.

Se evaluará la capacidad del estudiante para transformar un diseño gráfico en una estructura funcional y estéticamente agradable en la web.

Figura 15

Código HTML para la creación de un formulario web.



Fuente: Elaboración propia (2024).

RECURSOS INTERACTIVOS

"¡Amplía tus conocimientos en HTML5 y CSS3!

En esta sección encontrarás una selección de tutoriales interactivos, ejemplos prácticos y la documentación oficial más actualizada. ¡Haz clic aquí para comenzar a crear sitios web increíbles y responsive!"

En el siguiente enlace encontrarás un blog donde explican las principales etiquetas HTML5 más usadas, sigue el enlace de abajo:

<https://platzi.com/blog/etiquetas-html-debes-conocer/>

En el siguiente enlace encontrarás un videotutorial:

Aprende CSS3 en 15 minutos:

<https://www.youtube.com/watch?v=3yM5uXp-T>

UNIDAD II

LENGUAJE DE PROGRAMACIÓN JAVASCRIPT

Sintaxis básica – eventos - validaciones.

LENGUAJE DE PROGRAMACIÓN JAVASCRIPT

JavaScript es un lenguaje utilizado para hacer páginas web interactivas. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

Figura 16

Logotipo de Javascript



Fuente: Bibliografía Ramírez, F. (2023, julio 18). *JavaScript: Qué es, historia, para qué sirve, dónde se usa*. ITSoftware | Apps | Software | Data Analytics; ITSoftware SAS. <https://itsoftware.com.co/content/javascript-que-es-historia-para-que-sirve-donde-se-usa/>

Sintaxis básica

- **Mayúsculas y minúsculas:** JavaScript diferencia entre mayúsculas y minúsculas.
- **Punto y coma:** Aunque no es obligatorio, se recomienda finalizar cada instrucción con un punto y coma (;).
- **Comentarios:** Se utilizan para explicar el código y hacerlo más comprensible.

Figura 17

Ejemplo de sintaxis de javascript

```
ejemplo.js x
1  /*
2     Declaración de variables
3     Diferencia entre mayusculas y minusculas
4  */
5
6  let nombre = "Juan"; // Correcto
7  let Nombre = "Pedro"; // Variable diferente
8
9  // Ejemplo con punto y coma (recomendado):
10 let mensaje = "Hola, mundo";
11 console.log(mensaje);
12
13 // Ejemplo sin punto y coma (puede funcionar, pero no es recomendado):
14 let numero = 5
15 numero++
```

Fuente: Elaboración propia (2024).

Tipos de datos

- **Números:** Pueden ser enteros o decimales.
- **Cadenas de texto:** Se encierran entre comillas simples (") o dobles ("").
- **Booleanos:** Solo pueden tener dos valores: true o false.
- **Null:** Representa la ausencia intencional de un valor.
- **Undefined:** Indica que una variable ha sido declarada pero no ha sido asignada ningún valor.
- **Arrays:** Colecciones ordenadas de elementos.

Figura 18

Ejemplo de tipos de datos en javascript

```
1
2 let nombre = "Juan Pérez"; // Cadena
3 let edad = 30; // Enteros
4 let salario = 100.99; // Decimal
5
6 let esMayorDeEdad = true; // Booleano
7
8 let variableSinValor; // Declarada pero sin valor asignado
9 console.log(variableSinValor); // Imprimirá "undefined"
10 // Array
11 let frutas = ["manzana", "banana", "pera"];
12 let numeros = [1, 2, 3, 4, 5];
```

Fuente: Elaboración propia (2024).

EVENTOS EN JAVASCRIPT

¿Qué son los eventos?

Un evento no es más que cualquier acción que el usuario puede realizar en la pantalla del navegador, como hacer clic en algún lugar con el mouse, presionar un en el teclado, pasar el mouse sobre un elemento, cerrar la pestaña de navegación, enviar un formulario, entre otros.

Figura 19

Ejemplo de sintaxis de javascript



Nota: Boton click. (2021). Evento click. [Imagen]. Recuperado de <https://www.shutterstock.com/es/image-vector/click-here-button-hand-pointer-clicking-1934677607>

Tipos comunes de eventos

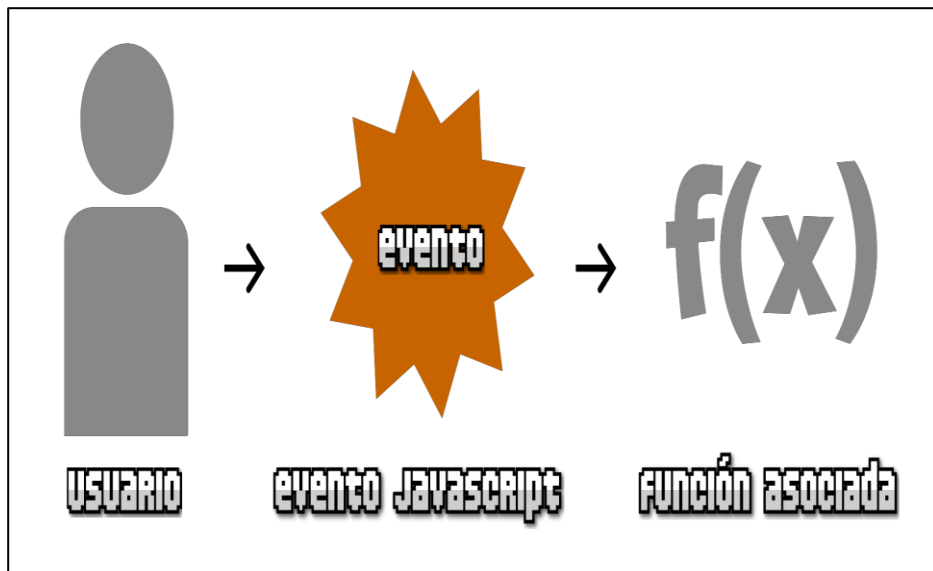
- **Eventos del mouse:** click, doubleclick, mouseover, mouseout, etc.
- **Eventos del teclado:** keydown, keyup, keypress.
- **Eventos del formulario:** submit, change, input.
- **Eventos de la ventana:** load, resize, scroll.

¿Cómo funcionan los eventos?

- **Detectores de eventos:** Son como los oídos de la página, siempre atentos a lo que ocurre.
- **Manejadores de eventos:** Son las funciones que se ejecutan cuando se detecta un evento.

Figura 20

Proceso de ejecución de un evento en javascript



Nota: Manz.dev (2024). ¿Qué son los eventos? [Imagen]
<https://lenguajejs.com/javascript/eventos/que-son-eventos/>

MANEJO DE EVENTOS

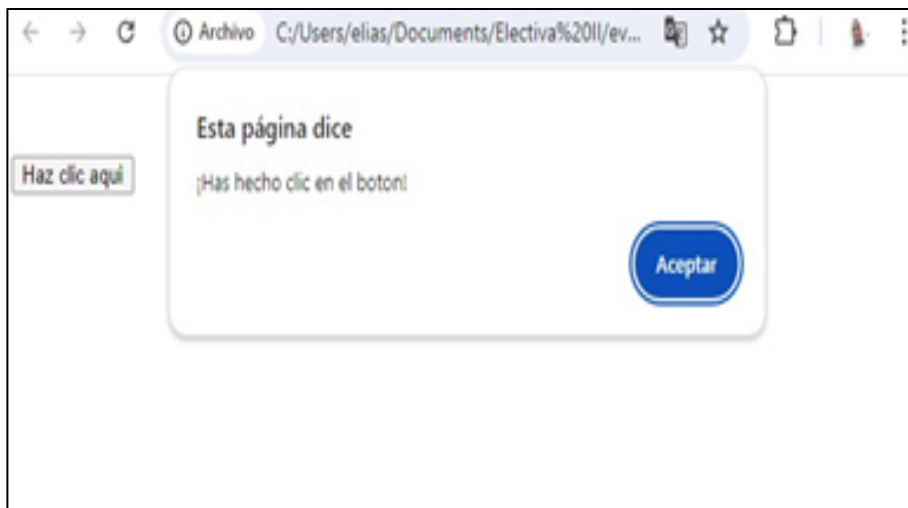
Mediante propiedades Javascript

En el siguiente este ejemplo:

- El botón tiene un atributo onclick **que** llama a la función **mostrarMensaje()** cuando se hace clic.
- La función **mostrarMensaje()** muestra un cuadro de alerta con un mensaje..

Figura 21

Ejemplo de evento **onclick** en javascript



Nota: Ejemplo de manejo de eventos mediante propiedades javascript. Evento click. [Capture]. Elaboración propia (2024).

Figura 22

Código javascript manejo de eventos mediante propiedades

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Evento Click</title>
5  </head>
6
7  <body>
8    <br><br>
9    <button onclick="mostrarMensaje()">Haz clic aqui</button>
10
11  <script>
12
13  function mostrarMensaje() {
14    alert(";Has hecho clic en el boton!");
15  }
16
17  </script>
18
19  </body>
20
21  </html>
22
```

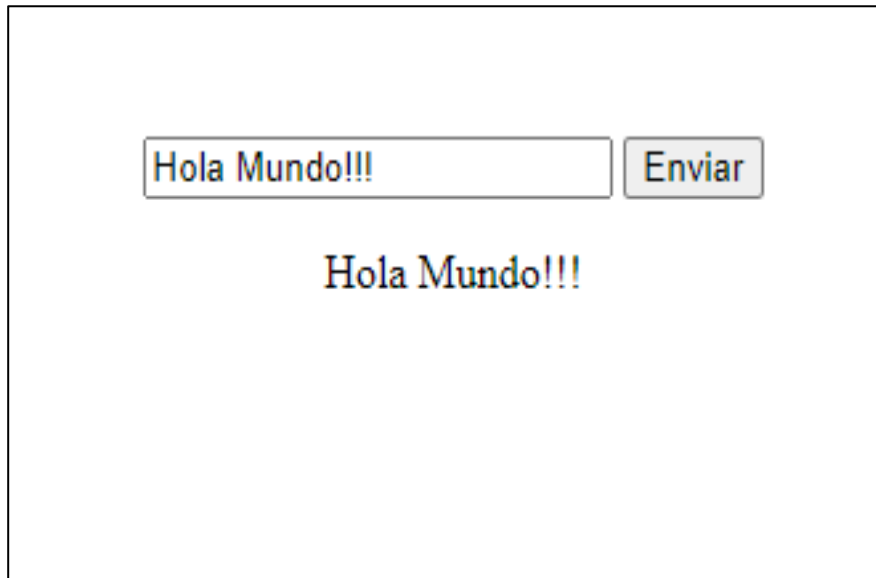
Nota: Código manejo de eventos mediante propiedades javascript. Evento click. [Capture]. Elaboración propia (2024).

RETO UNIDAD 2

¡Convierte tu navegador en un eco! Diseña una página que "repita" cualquier mensaje que el usuario escriba en un campo de texto. Al presionar un botón, el texto ingresado debe aparecer en otra parte de la pantalla, como si la página estuviera respondiendo al usuario.

Figura 23

Resultado del reto propuesto Simple Chat Web.



Fuente: Elaboración propia (2024).

RECURSOS INTERACTIVOS

"¡Amplía tus conocimientos en HTML5 y CSS3!

En esta sección encontrarás una selección de tutoriales interactivos, ejemplos prácticos y la documentación oficial más actualizada. ¡Haz clic aquí para comenzar a crear sitios web increíbles y responsive!"

En el siguiente enlace encontrarás un blog donde explican las principales etiquetas HTML5 más usadas. sigue el enlace de abajo:

<https://platzi.com/blog/etiquetas-html-debes-conocer/>

En el siguiente enlace encontrarás un videotutorial

Aprende CSS3 en 15 minutos:

https://www.youtube.com/watch?v=aNEOu7de_FQ

UNIDAD III

APLICACIONES WEB

Introducción a PHP y MySql.

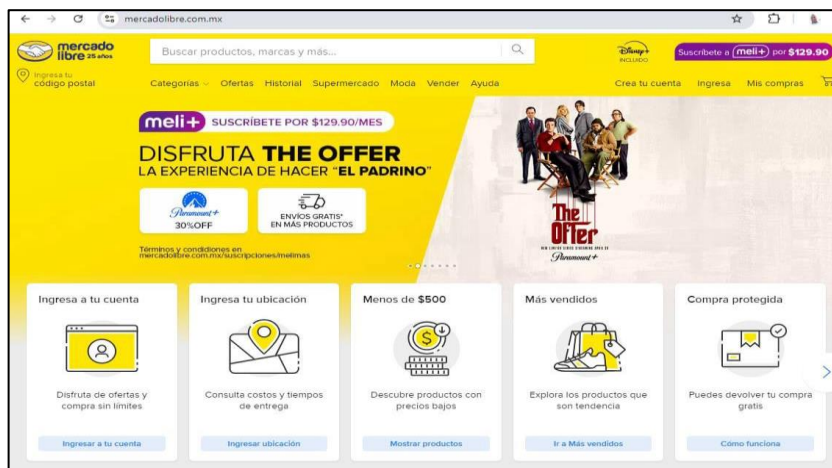
APLICACIONES WEB

Es un software **cliente-servidor** que permite realizar funciones determinadas en internet, como enviar mensajes, realizar compras, jugar videojuegos, hacer pagos, entre otras acciones.

Muchos sitios web te permiten realizar ciertas acciones en línea como realizar compras, comunicarte con tus clientes o colegas, hacer cálculos, realizar procesos, intercambiar información y proporcionar servicios de forma remota. Todos estos, conocidos también como **aplicaciones web**,

Figura 24

Aplicación de comercio electrónico



Nota: Mercado Libre. (s/f). Com.mx. Recuperado el 9 de agosto de 2024, de <https://www.mercadolibre.com.mx>

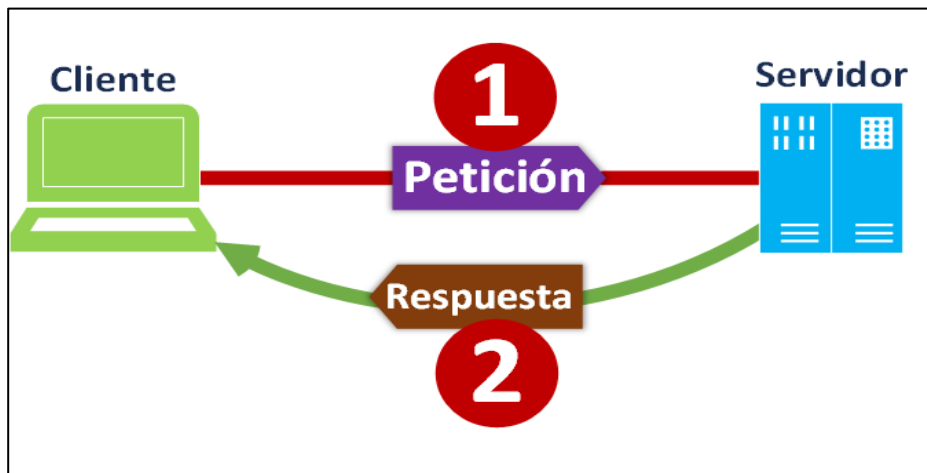
¿COMO FUNCIONAN?

El funcionamiento de las aplicaciones web es posible gracias a la interacción entre **el cliente** (normalmente el navegador) y el **servidor** (donde reside la aplicación).

Cuando utilizas una aplicación web, tu navegador envía una solicitud al servidor, que procesa la información y devuelve los datos necesarios para que la aplicación funcione en tu dispositivo.

Figura 25

Arquitectura Cliente-Servidor



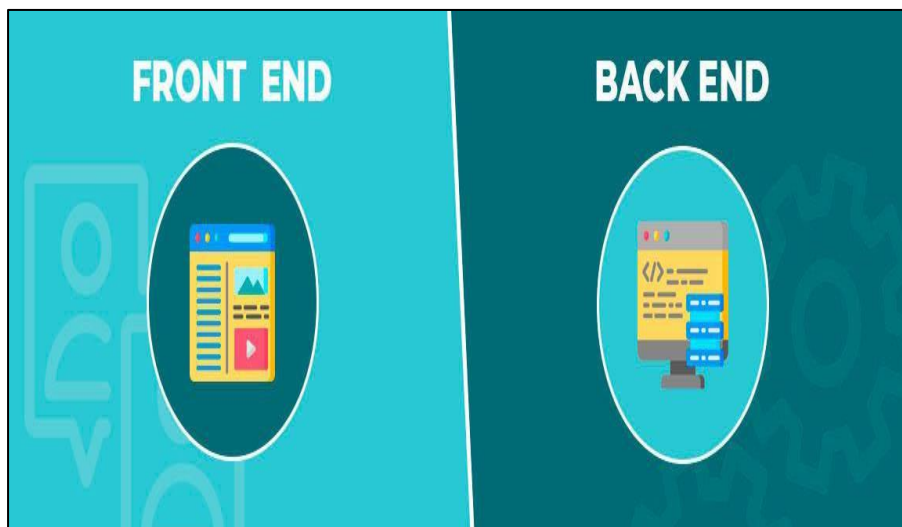
Nota: Ramón Invarato (01-09-2019). [Imagen] Arquitectura Cliente-Servidor. Recuperado de <https://jarroba.com/cliente-vs-servidor/>

¿QUÉ ES BACK-END Y FRONT END?

¿Has escuchado alguna vez estos términos? El Backend y el Frontend son dos partes fundamentales de la programación de una aplicación web. En este apartado vamos a hablar sobre cada una de ellas y explicaremos las diferencias entre **Frontend** (el lado del cliente) y **Backend** (el lado del servidor).

Figura 26

Front End y Back End



Nota: Comunicación, D. (2019, julio 26). ¿Qué es Backend y Frontend? Descubre Comunicación. <https://descubrecomunicacion.com/que-es-Backend-y-Frontend/>

¿QUÉ ES EL FRONT END?

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.

Figura 27

Elementos del Front End



Nota: Coco Solution. (2023, marzo 29). ¿Qué es el Frontend y sus características? Recuperado de <https://cocosolution.com/blog/que-es-frontend-y-backend/>

¿QUÉ ES EL BACK END?

Concepto engloba los códigos ocultos o que se encuentran en la parte de peticiones de una web y sirven para que en una página web o aplicación funcione perfectamente la capa de acceso a datos de un software o de cualquier dispositivo, es la parte no visible de una web.

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.

Figura 28

Elementos del Back End



Nota: Coco Solution. (2023, marzo 29). ¿Qué es el Backend y sus características? Recuperado de <https://cocosolution.com/blog/que-es-frontend-y-backend/>

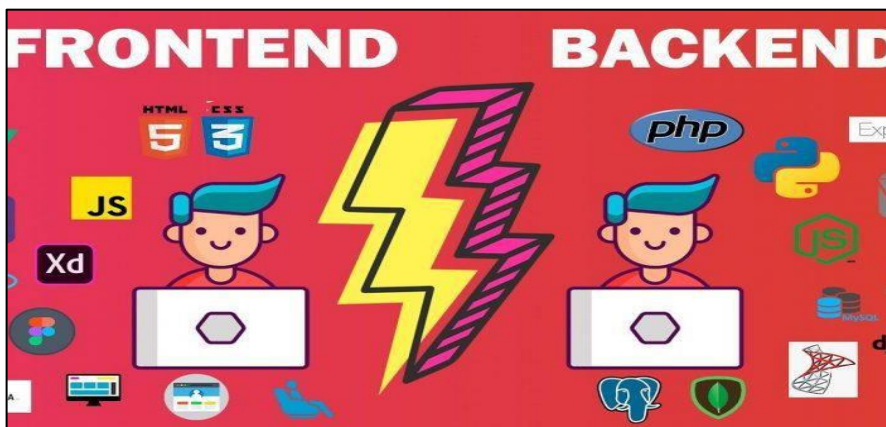
TECNOLOGÍAS USADAS

Los lenguajes principales para el **frontend** son HTML, CSS y JavaScript. De ellos se generan una gran cantidad de frameworks, por ejemplo: React, Redux, Angular, Bootstrap, Foundation, LESS, Sass, Stylus, entre otros.

Python, PHP, Ruby, C# y Java son algunos de los lenguajes de programación de **Backend**. Igual que el Frontend cuenta con frameworks, pero en su caso trabaja con Django, Laravel, Ruby On Rails y ASP.Net, entre otros.

Figura 29

Tecnologías usadas en el desarrollo Frontend y Backend



Nota: Ecdisis Estudio. (2021, 30 julio). [Imagen] ¿Qué es frontend y backend? <https://ecdisis.com/que-es-frontend-y-backend/>

INTRODUCCIÓN A PHP Y MYSQL

¡Absolutamente! PHP y MySQL son una combinación clásica y poderosa para desarrollar aplicaciones web dinámicas. Esta combinación te permite crear desde simples sitios web hasta aplicaciones complejas con bases de datos robustas.

¿Por qué PHP y MySQL?

- **PHP:** Es un lenguaje de programación del lado del servidor, fácil de aprender y ampliamente utilizado. Su sintaxis es similar al C, lo que lo hace accesible para muchos programadores.
- **MySQL:** Es un sistema de gestión de bases de datos relacionales (RDBMS) muy popular, conocido por su velocidad, fiabilidad y facilidad de uso. Es ideal para almacenar y gestionar grandes cantidades de datos.

¿Qué puedes hacer con PHP y MySQL?

Las posibilidades son casi ilimitadas:

- Sitios web dinámicos.
- Aplicaciones web.
- Backend de aplicaciones móviles.
- Herramientas de gestión.

PASOS BÁSICOS PARA CREAR UNA APLICACIÓN CON PHP y MySQL

1.- Configuración del Entorno de Desarrollo

Es necesario instalar un servidor web local (como XAMPP o WAMP) que incluya PHP, MySQL y un servidor web (Apache).

Figura 30

Aplicaciones necesarias para desarrollo web



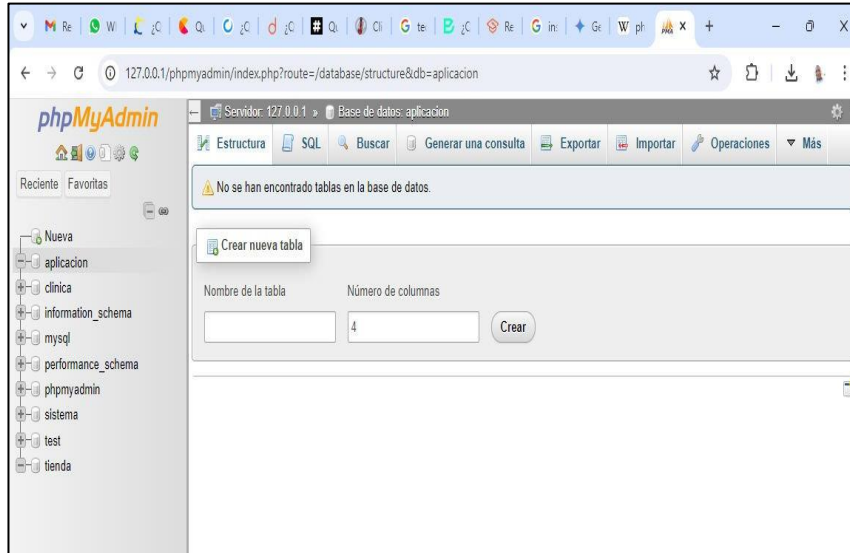
Nota: Ecdisis Estudio. (2021, 30 julio). [Imagen] ¿Qué es frontend y backend? <https://ecdisis.com/que-es-frontend-y-backend/>

2.- Creación de la base de datos

Utilizar **phpMyAdmin** o la línea de comandos para crear la base de datos y las tablas necesarias. phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL y MariaDB a través de páginas web, utilizando un navegador web.

Figura 31

phpMyAdmin interfaz web para administrar MySQL



Nota: Elaboración propia. (2024). [Capture] Interfaz gráfica de phpMyAdmin.

3.- Diseño de la interfaz

Crear las páginas HTML y utilizar CSS para darle estilo.

Realizaremos un ejemplo de cómo crear un formulario HTML básico con campos para email y contraseña, y cómo procesar esos datos en PHP para mostrarlos por pantalla.

Figura 32
Formulario de Login



The image shows a login form with the following elements:

- Inicio de Sesion**: The title of the form, displayed in a large, bold, black serif font.
- Email:**: A label followed by a rectangular text input field.
- Clave:**: A label followed by a rectangular text input field.
- Enviar**: A rectangular button with a light gray background and a thin border.

Nota: Elaboración propia. (2024). [Capture] Interfaz gráfica de Inicio de sesión.

INTRODUCCIÓN A PHP

Sintaxis básica

- **Etiquetas de PHP:**

Todo código PHP debe estar entre las etiquetas

```
<?php
```

```
?>
```

- **Comentarios:** Se utilizan para explicar el código y hacerlo

más legible. Existen dos tipos principales:

- **De una línea:**

```
// Este es un comentario.
```

- **De múltiples líneas:**

```
/* Este es un comentario que puede ocupar varias líneas */
```

- **Variables:** Se utilizan para almacenar datos. Se declaran con el símbolo \$ seguido del nombre de la variable. Los nombres de las variables son sensibles a mayúsculas y minúsculas.

```
$nombre = "Juan"; // Asignación de un valor a una variable.
```

A continuación, se muestra un ejemplo de código, usando el editor Visual Studio Code de la sintaxis explicada en la página anterior.

Figura 33
Ejemplo de sintaxis en PHP

```
1 <?php
2
3 /* COMENTARIOS
4  Sintaxis básica de un programa en PHP; etiquetas de apertura y cierre, declaración de
5  variables, uso de comentarios e instrucciones,
6
7 */
8
9 /* VARIABLES
10  -En PHP no hace falta declarar una variableEs un espacio de memoria en el computador
11  destinado para almacenar un valor. antes de su uso, ni establecer su tipo.
12
13  -Su nombre puede ser una combinación de letras (a-Z), números y guiones bajos, precedidos
14  con el signo $.
15
16  -El primer carácter no puede ser un número, y php diferencia entre mayúsculas y minúsculas.
17  Por ejemplo $nombre no es igual a $Nombre.
18 */
19
20 $nombre = "Juan"; // Declaración y Asignación de un valor a una variable
21
22 // PHP requiere que las instrucciones terminen en punto y coma al final de cada sentenciaEs una
23 porción de código de algún lenguaje de programación.
24
25 echo $nombre; // Impresión de la variable
26
27 ?>
```

Nota: Elaboración propia. (2024). [Capture] Ejemplo de declaración de variables, comentarios en PHP.

TIPOS DE DATOS EN PHP

PHP es un lenguaje de tipado débil, no es necesario declarar el tipo de dato de una variable. Sin embargo, los tipos de datos más comunes son: **String**, **Integer**, **Float**, **Boolean** y **Array**.

Figura 34

Ejemplo de tipos de datos en PHP

```
1 <?php
2
3 /* TIPOS DE DATOS
4  PHP es un lenguaje de programación dinámico que no requiere una declaración explícita de
5  tipos de datos. Sin embargo, entender los tipos de datos es fundamental para escribir
6  código eficiente y correcto.
7  */
8 $id = 123456; // Integer: Representan números enteros, tanto positivos como negativos.
9
10 $nombre = 'Mike Peterson'; // String: Cadenas de caracteres, texto encerrado entre comillas
11    simples o dobles
12 $sueldo = 500.50; // Float: Representan números de punto flotante (decimales).
13
14 $activo = true; // Boolean: Valores lógicos, true o false.
15
16 //Array: Colecciones ordenadas de valores, pueden ser de cualquier tipo de dato.
17 $materias = ["Programación III", "Elctiva II", "Bases de Datos"];
18
19 // Imprimir los valores de las variables y sus tipos con la función gettype($var)
20
21 echo "<h1>Tipos de Datos</h1>".<br><br>";
22
23
24 echo "Id: $id" . " Tipo: " . gettype($id);
25 echo "Nombre: $nombre" . " Tipo: " . gettype($nombre) . "<br><br>";
26 echo "Sueldo: $sueldo" . " Tipo: " . gettype($sueldo) . "<br><br>";
27 echo "Activo: $activo" . " Tipo: " . gettype($activo) . "<br><br>";
28 echo "Materias: " . print_r($materias) . " Tipo: " . gettype($materias);
29
30 ?>
```

Nota: Elaboración propia. (2024). [Capture] Ejemplo de tipos de datos en PHP.

EJECUCIÓN EN EL SERVIDOR WEB

Ejecutamos el archivo escribiendo la siguiente dirección en la URI (identificador de recursos uniforme):

127.0.0.1/{carpeta}/nombre_de_archivo.php

Figura 35

Ejecución del ejemplo de tipos de datos en PHP



Nota: Elaboración propia. (2024). [Capture] Ejecución del ejemplo de tipos de datos en PHP.

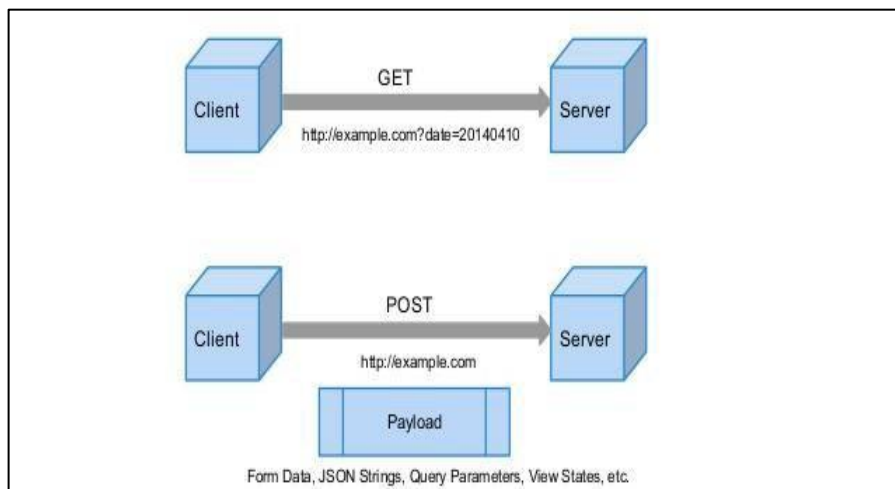
ENVÍO DE DATOS AL SERVIDOR

Un navegador web se comunica con el servidor típicamente usando uno de los dos métodos HTTP (Protocolo de transferencia de hipertexto): **GET** y **POST**.

En el método **GET**, los datos se envían como parámetros de **URL**. Los datos enviados a través del método **POST** no serán visibles en la **URL**.

Figura 36

Métodos de envío de datos en PHP



Nota: [Capture] Métodos de envío de datos en PHP. Recuperado de <https://programacion7ulatsaavas.wordpress.com/2016/06/07/metodo-get-y-post/>

Envío de datos con GET

Si tenemos el siguiente formulario para enviar los datos de un producto:

Figura 37

Métodos de envío de datos en PHP



The image shows a browser window with the address bar containing the URL: 127.0.0.1/sistema/venta.html?producto=Harina&cantidad=5&precio=35.6. The main content area displays the title "Datos del Producto" in a large, bold, black serif font. Below the title, there are three text input fields stacked vertically, each with a label to its left: "Producto", "Cantidad", and "Precio". At the bottom left of the form area is a button labeled "Enviar".

Nota: Elaboración propia (2024). [Capture] Métodos de envío de datos con el método GET.

Observamos que los datos ingresados en los campos de texto, se envían a través del método GET y son visibles en la URI, nunca se envía información sensible. No se pueden enviar datos binarios (archivos, imágenes...).

Respuesta del servidor

En PHP los datos se administran con el array asociativo \$_GET.

Figura 38

Respuesta del servidor del envío por el método GET



Nota: Elaboración propia. [Capture] Respuesta de envío de datos con el método GET.

La url que resulta de hacer clic en submit es de la forma:

<http://127.0.0.1/sistema/respuesta.php?producto=Harina&cantidad=5&precio=35.6>

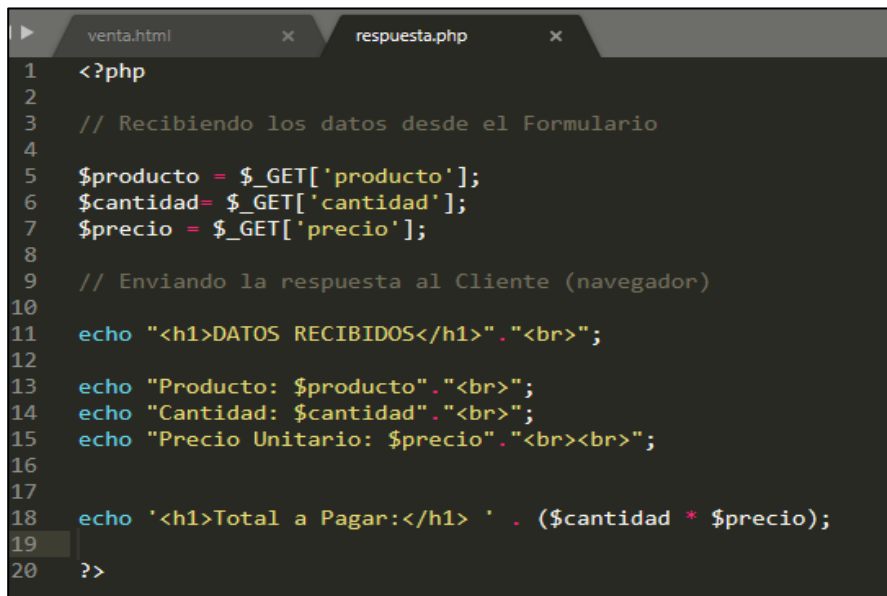
Se observan los valores enviados en la URL.

Código php en el servidor

Para obtener los datos se usa la variable global `$_GET['clave']`, donde la clave es el **name** del campo de texto asociado en el formulario. Se asignan a las variables y luego se imprimen, A continuación, se muestra el código del script “**respuesta.php**”:

Figura 39

Respuesta del servidor del envío por el método GET

A screenshot of a code editor with two tabs: 'venta.html' and 'respuesta.php'. The 'respuesta.php' tab is active and shows the following PHP code:

```
1 <?php
2
3 // Recibiendo los datos desde el Formulario
4
5 $producto = $_GET['producto'];
6 $cantidad= $_GET['cantidad'];
7 $precio = $_GET['precio'];
8
9 // Enviando la respuesta al Cliente (navegador)
10
11 echo "<h1>DATOS RECIBIDOS</h1>". "<br>";
12
13 echo "Producto: $producto". "<br>";
14 echo "Cantidad: $cantidad". "<br>";
15 echo "Precio Unitario: $precio". "<br><br>";
16
17
18 echo '<h1>Total a Pagar:</h1> ' . ($cantidad * $precio);
19
20 ?>
```

Nota: Elaboración propia (2024). [Capture] Script PHP para obtener y procesar los datos enviados desde el formulario con el método GET.

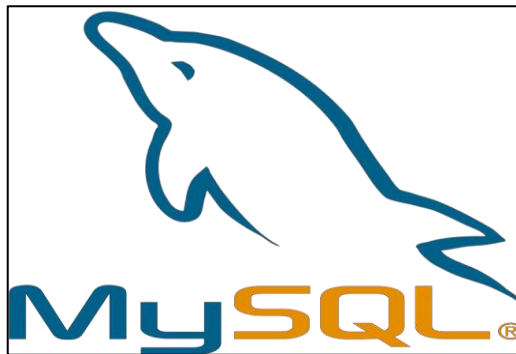
INTRODUCCIÓN A MYSQL

MySQL es un sistema de gestión de bases de datos relacionales (SGBDR). Esto significa que organiza la información en tablas relacionadas entre sí, lo que facilita la gestión y recuperación de datos. Es conocido por su facilidad de uso, alta performance y código abierto.

Es la base de datos de elección para muchas aplicaciones web, desde blogs y foros hasta tiendas en línea y sistemas de gestión de contenido (CMS).

Figura 40

Logotipo del manejador de base de datos MySQL



Nota: Boton click. (septiembre 09 de 2024). Logotipo de MySQL. [Imagen]. Recuperado de Guerra, J. (2022, 14 septiembre). Conexiones asíncronas a MySQL en Node.js. JavGuerra. <https://javguerra.github.io/2022-09-14-mysql-connect/>

Conexión a el servidor MySQL

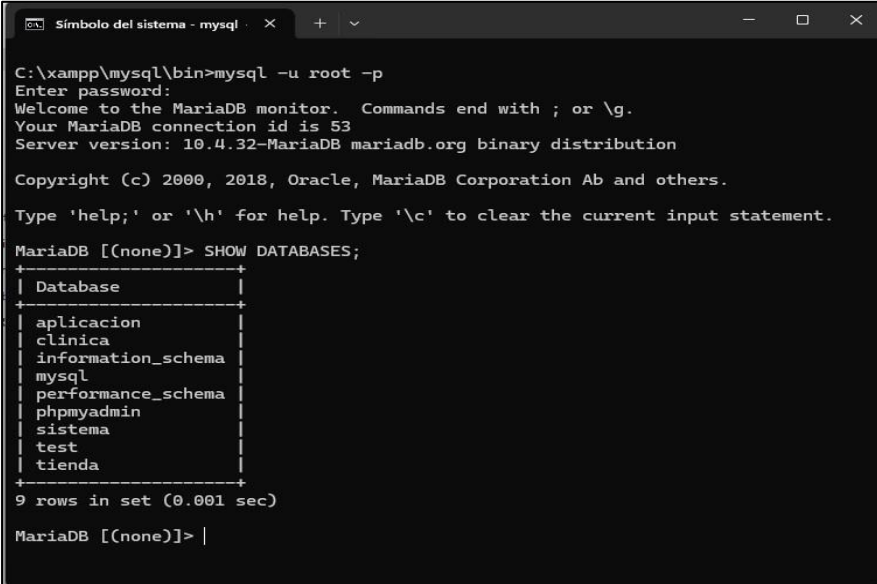
1. Línea de comandos

Abrimos la consola de Windows (cmd) y nos ubicarnos en la carpeta donde está instalado Mysql. En nuestro caso como instalamos XAMPP, iremos a la siguiente ruta: **C:\xampp\mysql\bin>** Escribimos el siguiente comando:

```
mysql -u usuario -p password
```

Figura 41

Conexión a MySQL desde consola CMD en Windows



```
Símbolo del sistema - mysql . x + v
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 53
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| aplicacion |
| clinica |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| sistema |
| test |
| tienda |
+-----+
9 rows in set (0.001 sec)

MariaDB [(none)]> |
```

Nota: Elaboración propia. (2024). [Capture] Conexión a MySQL mediante línea de comandos.

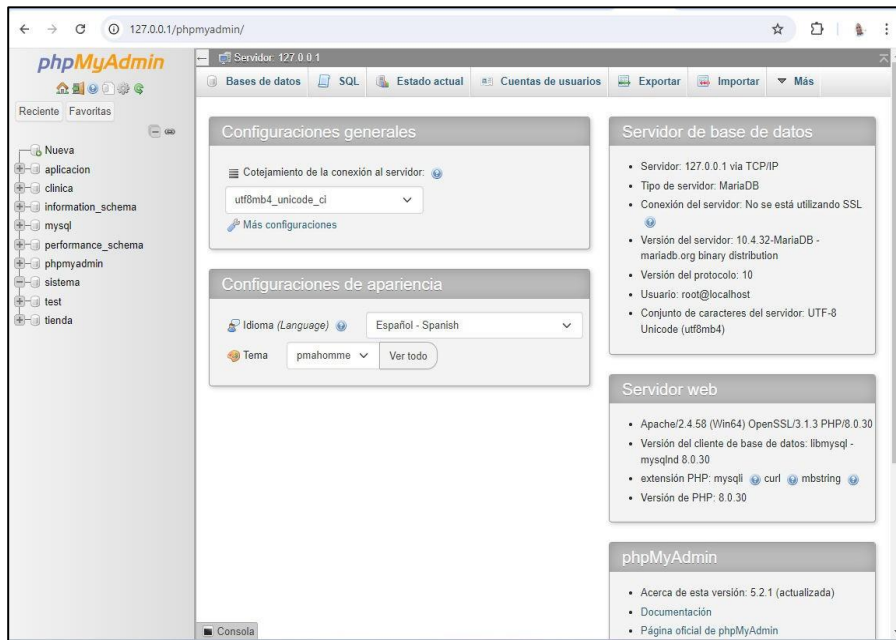
2. Cliente gráfico

Utilizarás un cliente MySQL para conectarte a tu servidor y ejecutar comandos SQL. En nuestro caso como tenemos instalado XAMPP, escribiremos en la URL del navegador lo siguiente:

<http://127.0.0.1/phpmyadmin/>

Figura 42

Conexión a MySQL usando phpMyadmin



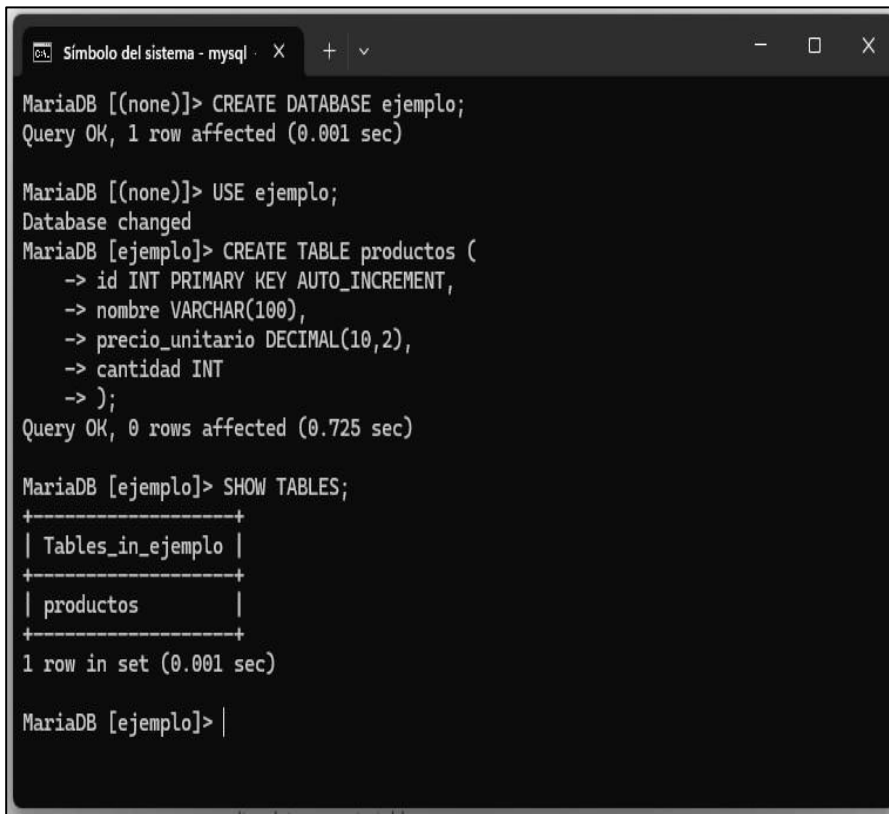
Nota: Elaboración propia (2024). [Capture] Conexión a MySQL mediante el cliente gráfico (web) phpMyadmin.

CREACIÓN DE UNA BASE DE DATOS Y TABLAS

1. Línea de comandos

Figura 43

Creación de base de datos y tabla en consola MySQL



```
Símbolo del sistema - mysql
MariaDB [(none)]> CREATE DATABASE ejemplo;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE ejemplo;
Database changed
MariaDB [ejemplo]> CREATE TABLE productos (
  -> id INT PRIMARY KEY AUTO_INCREMENT,
  -> nombre VARCHAR(100),
  -> precio_unitario DECIMAL(10,2),
  -> cantidad INT
  -> );
Query OK, 0 rows affected (0.725 sec)

MariaDB [ejemplo]> SHOW TABLES;
+-----+
| Tables_in_ejemplo |
+-----+
| productos          |
+-----+
1 row in set (0.001 sec)

MariaDB [ejemplo]> |
```

Nota: Elaboración propia. (2024). [Capture] Creación de base de datos y tabla en MySQL mediante línea de comandos.

CONEXIÓN A MYSQL DESDE PHP

Puedes acceder a las bases de datos MySQL directamente por medio de scripts PHP. Esto te permite leer y escribir datos a tu base de datos directamente desde tu sitio web. A continuación, crearemos un script para conectarnos a la base de datos de MySQL.

Figura 44

Conexión de PHP con MySQL

```
1 <?php
2
3 // parámetros de conexión
4 $servername = "localhost"; // Dirección IP del servidor
5 $username = "root"; // Nombre del usuario de mysql
6 $password = ""; // Clave del usuario de mysql
7 $dbname = "ejemplo"; // Nombre de la base de datos.
8
9 // Crear conexión
10 $conexion = new mysqli($servername, $username, $password, $dbname);
11
12 // Verificar conexión
13 if ($conexion->connect_error) {
14     die("Conexión fallida: " . $conexion->connect_error);
15 } else {
16     echo "Se ha realizado la Conexión";
17 }
18
19 ?>
```

Nota: Elaboración propia. (2024). [Capture] Conexión a MySQL desde un script PHP.

RETO UNIDAD 3 - PHP

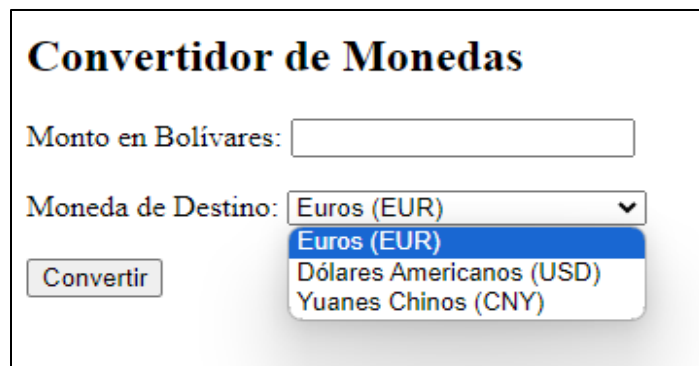
Desarrollar un conversor de monedas utilizando PHP y HTML. El convertidor debe permitir al usuario ingresar una cantidad en Bolívares y convertirlos a euros, dólares o yuanes.

Los datos se enviarán desde el formulario HTML al script PHP a través del método POST o GET, donde se procesarán y validarán. La respuesta, que muestra la cantidad convertida, se imprimirá en el lado del cliente.

- Asegúrate de validar que el monto sea numérico.
- Crear un archivo para el formulario: **index.html** y otro para procesar y realizar los cálculos: **procesar.php**

Figura 45

Convertidor de monedas.



The image shows a web form titled "Convertidor de Monedas". It contains a text input field labeled "Monto en Bolívares:", a dropdown menu labeled "Moneda de Destino:" with a downward arrow, and a button labeled "Convertir". The dropdown menu is open, showing three options: "Euros (EUR)" (highlighted in blue), "Dólares Americanos (USD)", and "Yuanes Chinos (CNY)".

Fuente: Elaboración propia (2024).

RETO UNIDAD 3 - MYSQL

Se pide crear una base de datos llamada **sistema**, que permita almacenar los datos de los usuarios de la aplicación en una tabla. Los usuarios deben autenticarse mediante su correo y contraseña. Además, se desea conocer el nombre del usuario.

Para el desarrollo del reto puede usar línea de comandos o phpMyadmin, considere los tipos de datos para cada campo de la tabla tomando en cuenta que la contraseña se almacenará encriptada.

Figura 46

Estructura de la tabla usuarios.

The screenshot shows the phpMyAdmin interface for the 'sistema' database, displaying the structure of the 'usuarios' table. The table has the following columns:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	id	int(11)			No	Ninguna		AUTO_INCREMENT
2	nombre	varchar(50)	utf8mb4_general_ci		No	Ninguna		
3	email	varchar(255)	utf8mb4_general_ci		No	Ninguna		
4	clave	varchar(255)	utf8mb4_general_ci		No	Ninguna		

Below the table structure, there are options to select all elements, and actions like 'Examinar', 'Cambiar', and 'Eliminar'. There are also checkboxes for 'Único', 'Índice', 'Espacial', and 'Texto completo'. At the bottom, there are options for 'Imprimir', 'Planteamiento de la estructura de tabla', 'Mover columnas', and 'Normalizar'.

Fuente: Elaboración propia (2024).

RECURSOS INTERACTIVOS

"¡Amplía tus conocimientos en PHP y MySQL!"

En esta sección encontrarás una selección de tutoriales interactivos, ejemplos prácticos y la documentación oficial más actualizada. ¡Haz clic aquí para comenzar a crear sitios web increíbles y responsive!"

En el siguiente enlace encontrarás un blog donde explica como configurar el entorno de desarrollo. Sigue el enlace de abajo:

https://www.ionos.com/es-us/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/?srsltid=AfmBOorc8QJJWGEQccYt1VOC3r_gWIOMGYDKfgmCC2WNQxxY134KYj7Y

En el siguiente enlace encontrarás un videotutorial **de PHP y MySQL:**

https://www.youtube.com/watch?v=_eWI9BxPF2g

UNIDAD IV

DESARROLLO DE APLICACIONES WEB

Sistema de autenticación de usuarios

SISTEMA DE AUTENTICACIÓN DE USUARIOS

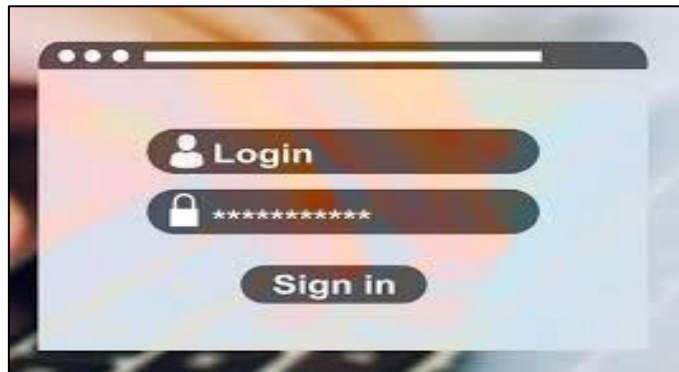
Es un **mecanismo de seguridad** que verifica la identidad de un individuo que intenta acceder a un sistema o aplicación. En términos más simples, es como un portero digital que comprueba si tienes permiso para entrar.

¿Cómo funciona?

- **Credenciales:** El usuario proporciona unas credenciales, típicamente un nombre de usuario y una contraseña, aunque pueden utilizarse otros métodos como huellas dactilares, reconocimiento facial o tokens.
- **Verificación:** El sistema compara estas credenciales con una base de datos de usuarios autorizados.
- **Acceso:** Si las credenciales coinciden, se otorga acceso al usuario. De lo contrario, se le deniega.

Figura 47

Formulario de inicio de sesión.



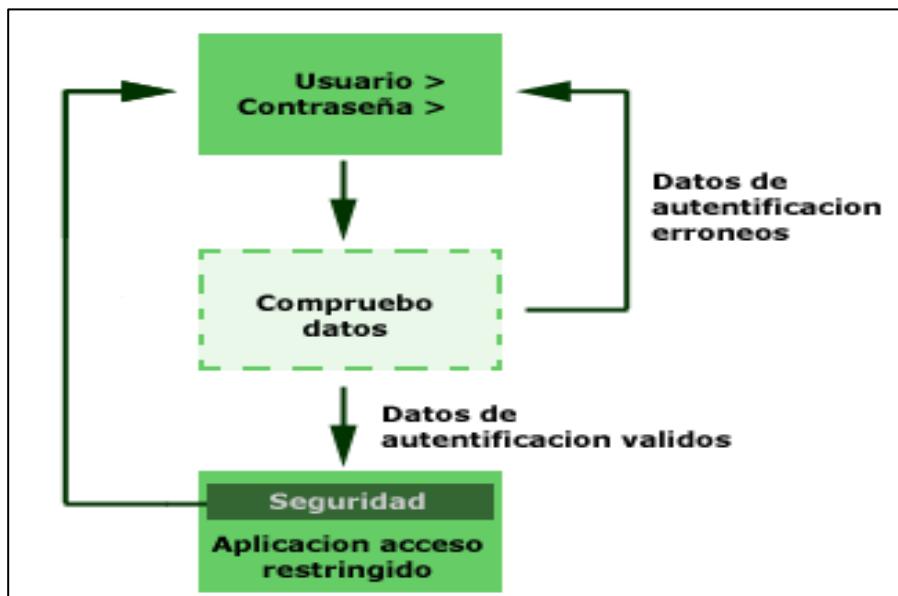
Nota: Alura Latam. (s/f). Tipos de Autenticación. [Imagen] Recuperado el 11 de agosto de 2024, de <https://www.aluracursos.com/blog/assets/tipos-deautenticacion/img3.png>

FUNCIONAMIENTO DEL SISTEMA DE AUTENTICACIÓN

Vamos a empezar por definir un diagrama para realizar la autenticación de usuario en unas páginas web, que nos servirá para programar luego las páginas ajustándose al diagrama de la imagen a continuación.

Figura 48

Esquema de un sistema de autenticación



Nota: Miguel Ángel Álvarez EscuelaIT.com. Diagrama de Autenticación. [Imagen] Recuperado el 11 de agosto de 2024, de <https://desarrolloweb.com/articulos/images/autenticacion/diagrama-inicial.gi>

Estructura del sistema de autenticación

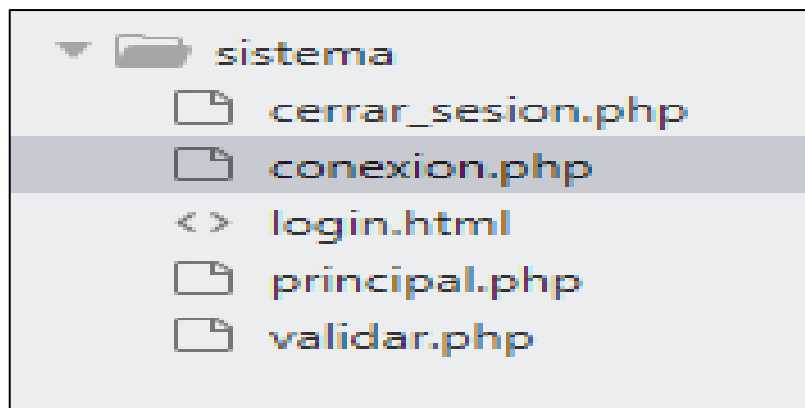
Los archivos necesarios para la creación del sistema de autenticación de usuarios son los siguientes.

- 1.- **conexion.php:** Contiene los datos de conexión a la base de datos.

- 2.- **login.html**: Muestra el formulario de inicio de sesión.
- 3.- **validar.php**: Recibe los datos del formulario, verifica si existe el usuario en la base de datos. Si las credenciales son correctas, inicia una sesión y redirige al usuario a la página principal.
- 4.- **principal.php**: Muestra el contenido de la página principal solo si el usuario está autenticado.
- 5.- **logout.php**: Destruye la sesión y redirige al usuario al formulario de inicio de sesión.

Figura 49

Estructura de los archivos en la carpeta sistema dentro del directorio web.



Nota: Elaboración propia (2024). [Capture]. Estructura de los archivos dentro de la carpeta sistema en el directorio web.

CONEXIÓN CON LA BASE DE DATOS EN MYSQL

Mediante el script **conexión.php** nos conectaremos a la base de datos llamada. “**sistema**” planteada como Reto de la unidad 3. Los parámetros para la conexión serán:

- **HOST:** Localhost.
- **USUARIO:** root.
- **CONTRASEÑA:** sin contraseña.
- **BASE DE DATOS:** sistema.

Figura 50

Archivo de conexión a MySQL desde PHP

```
conexion.php x
1  <?php
2
3  // parámetros de conexión
4  $servername = "localhost"; // Dirección IP del servidor
5  $username = "root";       // Nombre del usuario de mysql
6  $password = "";          // Clave del usuario de mysql
7  $dbname = "sistema";     // Nombre de la base de datos.
8
9  // Crear conexión
10 $conexion = new mysqli($servername, $username, $password, $dbname);
11
12 // Verificar conexión
13 if ($conexion->connect_error) {
14     die("Conexión fallida: " . $conn->connect_error);
15 }
16
17 ?>
18
```

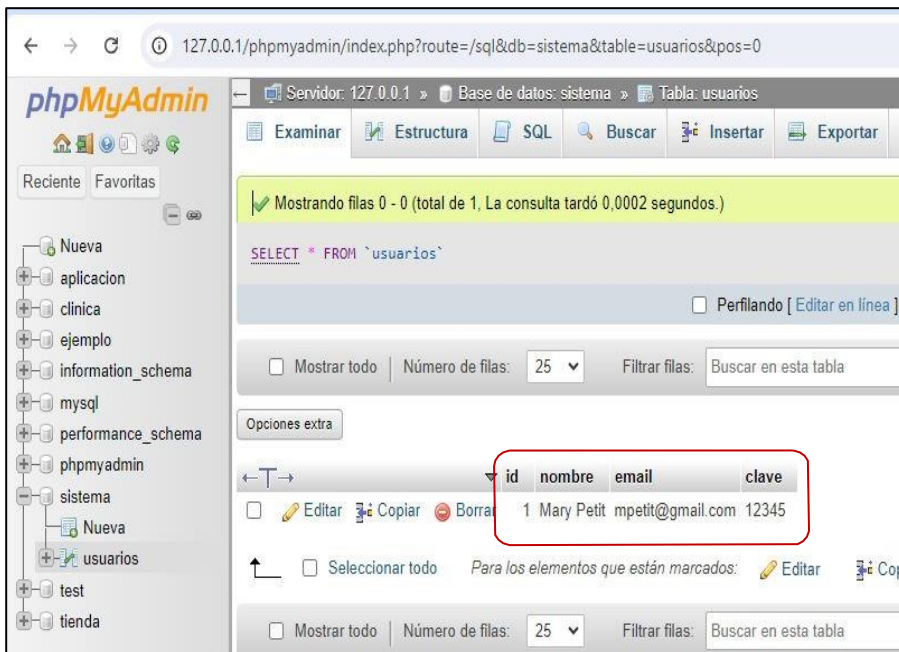
Nota: Elaboración propia (2024). [Capture]. Archivo de conexión a MySQL desde PHP.

AÑADIR UN REGISTRO EN LA TABLA USUARIOS

Usando phpMyAdmin crearemos un usuario para la aplicación en la tabla **usuarios** de la base de datos **sistema**.

Figura 51

Añadir un registro en la tabla usuarios en MySQL



The screenshot shows the phpMyAdmin interface for the 'sistema' database, specifically the 'usuarios' table. The table structure is as follows:

id	nombre	email	clave
1	Mary Petit	mpetit@gmail.com	12345

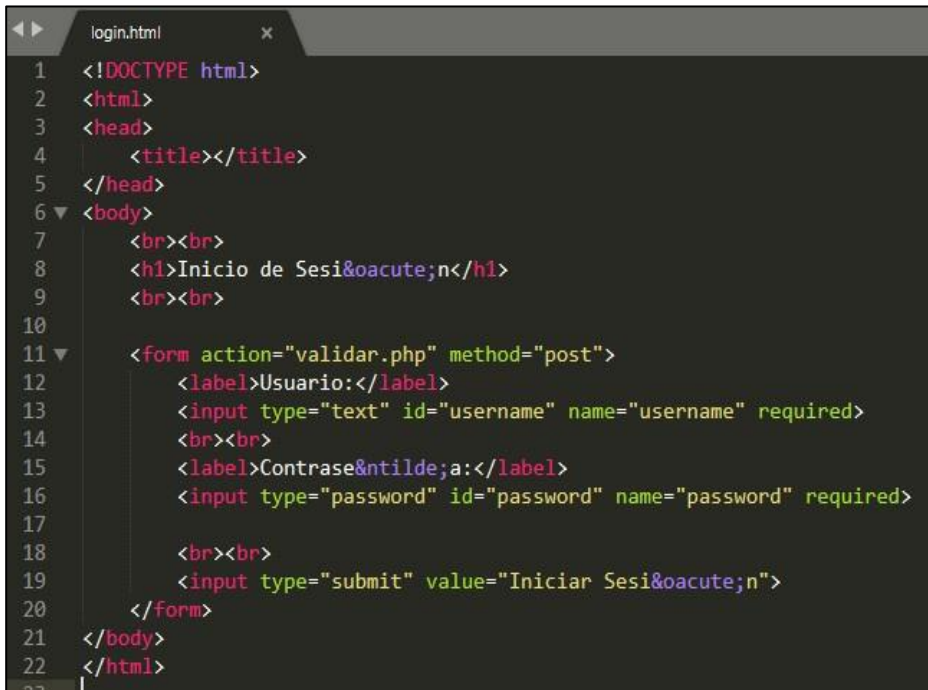
Nota: Elaboración propia (2024). [Capture]. Añadir un registro en la tabla usuarios de la base de datos sistema.

CÓDIGO DE CREACIÓN DEL FORMULARIO LOGIN

Usando el editor de código, realizaremos nuestro formulario de Inicio de Sesión.

Figura 52

Código HTML del formulario login.html

A screenshot of a code editor window titled 'login.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7   <br><br>
8   <h1>Inicio de Sesi&oacute;n</h1>
9   <br><br>
10
11 <form action="validar.php" method="post">
12   <label>Usuario:</label>
13   <input type="text" id="username" name="username" required>
14   <br><br>
15   <label>Contrase&ntilde;a:</label>
16   <input type="password" id="password" name="password" required>
17
18   <br><br>
19   <input type="submit" value="Iniciar Sesi&oacute;n">
20 </form>
21 </body>
22 </html>
```

Nota: Elaboración propia (2024). [Capture]. Código HTML para la creación del formulario de Login donde se ingresarán las credenciales de autenticación.

CÓDIGO DEL SCRIPT VALIDAR (LADO DEL SERVIDOR)

Código del SCRIPT que validará si los datos ingresados en el formulario coinciden con los de la base de datos.

Figura 53

Código del script validar.php

```
validar.php x
1 <?php
2 // Incluir archivo de conexión
3 include 'conexion.php';
4
5 // Obtener datos del formulario
6 $username = $_POST['username'];
7 $password = $_POST['password'];
8
9 // Consultar usuario en la base de datos
10 $sql = "SELECT * FROM usuarios WHERE email='$username'";
11
12 // Ejecutar la consulta y almacenar el resultado
13 $result = $conn->query($sql);
14
15 // Si la consulta devuelve un registro, significa que
16 if ($result->num_rows == 1) {
17     $row = $result->fetch_assoc();
18
19     if ($row['clave']==$password) {
20         // Inicio de sesión exitoso
21         session_start();
22         $_SESSION['user'] = $row['nombre']; // Creamos una variable de sesión
23         header("Location: principal.php"); exit();
24     } else {
25         echo "Contraseña incorrecta."."<br><br>";
26     }
27 } else {
28     echo "Usuario no encontrado."."<br><br>";
29 }
30 echo "<a href='login.html'>Volver al Inicio de Sesión</a>";
31
32 $conn->close();
33 ?>
```

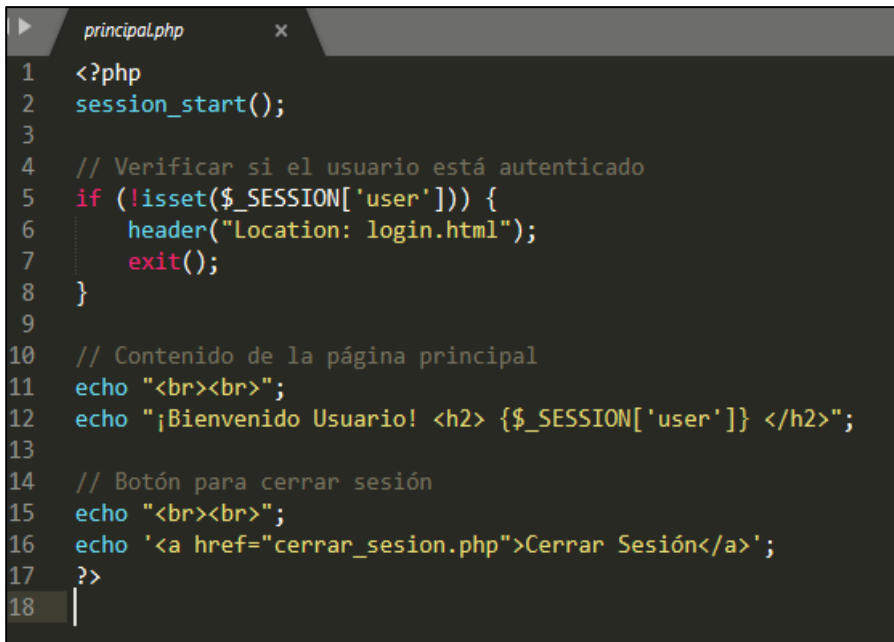
Nota: Elaboración propia (2024). [Capture]. Código PHP para validar los datos enviados desde el formulario.

CÓDIGO DE LA PÁGINA PRINCIPAL

Código del SCRIPT de la página principal de la aplicación. si los datos de autenticación son correctos, se redirigirá al navegador a la aplicación página principal del sistema.

Figura 54

Código del script de seguridad de la aplicación

A screenshot of a code editor window titled 'principal.php'. The code is as follows:

```
1 <?php
2 session_start();
3
4 // Verificar si el usuario está autenticado
5 if (!isset($_SESSION['user'])) {
6     header("Location: login.html");
7     exit();
8 }
9
10 // Contenido de la página principal
11 echo "<br><br>";
12 echo "¡Bienvenido Usuario! <h2> {$_SESSION['user']} </h2>";
13
14 // Botón para cerrar sesión
15 echo "<br><br>";
16 echo '<a href="cerrar_sesion.php">Cerrar Sesión</a>';
17 ?>
18 |
```

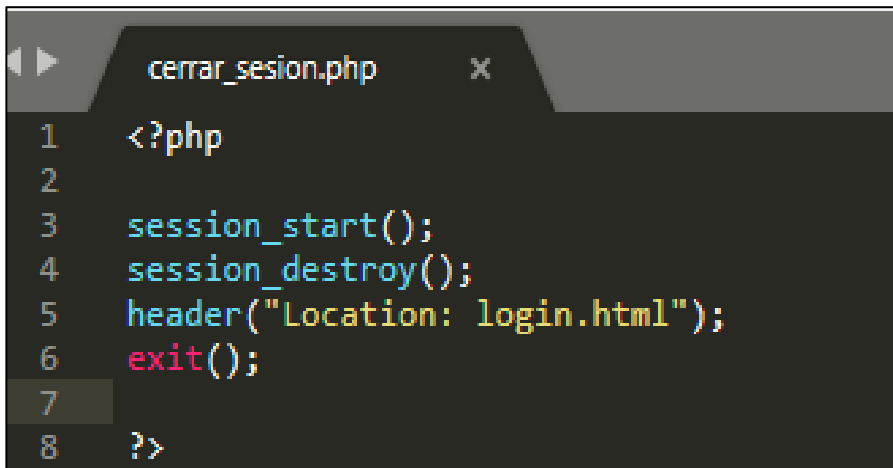
Nota: Elaboración propia (2024). [Capture]. Código PHP de la página principal de la aplicación. Se valida si está autenticado en caso que quisiera acceder de manera no permitida a esa página.

CÓDIGO DEL SCRIPT PARA CERRAR SESIÓN (LADO DEL SERVIDOR)

Parece interesante, pues, ofrecer al visitante la opción de acabar la sesión en cualquier momento, para asegurarnos en ese caso que la sesión se ha terminado y no se podrá acceder si no es introduciendo nuevamente el usuario y contraseña correctos.

Figura 55

Código del script `cerrar_sesion.php`



```
cerrar_sesion.php x
1  <?php
2
3  session_start();
4  session_destroy();
5  header("Location: login.html");
6  exit();
7
8  ?>
```

Nota: Elaboración propia (2024). [Capture]. Código PHP de la página principal de la aplicación. Se valida si está autenticado en caso que quisiera acceder de manera no permitida a esa página.

PRUEBA CON USUARIO NO REGISTRADO

Ingreso de una cuenta de usuario que no esté registrada en la base de datos.

Figura 56

Prueba ingresando un usuario No registrado



The screenshot shows a web browser window with the address bar displaying '127.0.0.1/sistema/login.html'. The main content of the page is a login form with the following elements:

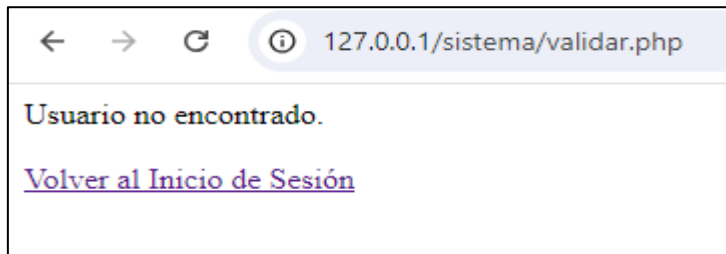
- A large heading: **Inicio de Sesión**
- A label 'Usuario:' followed by a text input field containing the email address 'eliasboc@gmail.com'.
- A label 'Contraseña:' followed by a password input field containing six dots '.....'.
- A button labeled 'Iniciar Sesión'.

Nota: Elaboración propia (2024). [Capture]. Prueba de inicio de sesión ingresando una cuenta de usuario no registrada.

PRUEBAS DE INICIO DE SESIÓN

Figura 57

Prueba ingresando un usuario No registrado



Nota: Elaboración propia (2024). [Capture]. Respuesta del servidor para el caso usuario no registrado.

Figura 58

Prueba ingresando una contraseña incorrecta



Nota: Elaboración propia (2024). [Capture]. Prueba ingresando una cuenta registrada pero la contraseña incorrecta.

PRUEBA CON CREDENCIALES CORRECTOS

Finalmente probaremos introduciendo las credenciales correctamente (usuario y contraseña). Enviamos los datos y esperamos la respuesta del servidor.

Figura 59

Prueba ingresando un usuario No registrado



The image shows a login form with the following elements:

- Inicio de Sesión**: The main title of the form.
- Usuario:** A text input field containing the email address `mpetit@gmail.com`.
- Contraseña:** A password input field containing four dots (••••).
- Iniciar Sesión**: A button to submit the login information.

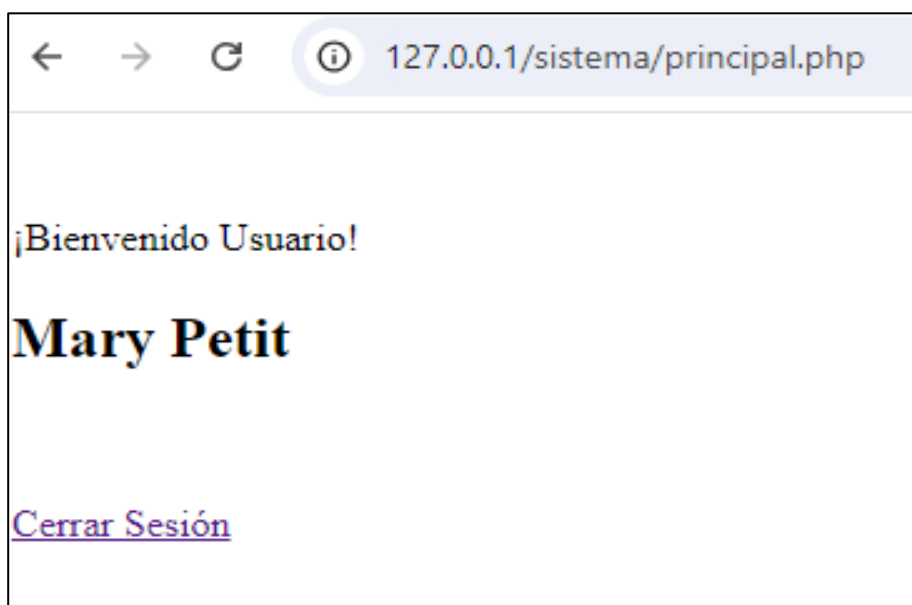
Nota: Elaboración propia (2024). [Capture]. Prueba de inicio de sesión ingresando las credenciales correctamente.

ACCESO A LA PÁGINA PRINCIPAL DE LA APLICACIÓN

Una vez validados las credenciales correctamente (usuario y contraseña) en el lado del servidor. Nos redirige a la ventana principal de la aplicación y guarda la sesión activa.

Figura 60

Página principal de la aplicación (Inicio de sesión correcto)



Nota: Elaboración propia (2024). [Capture]. Inicio de sesión correcto, da acceso a la página principal de la aplicación.

RETO UNIDAD 4

Para aumentar la capa de seguridad de la aplicación se plantea realizar lo siguiente:

Control de los datos de autenticación en PHP.

- Encriptar la contraseña usando **password_hash()**.

Validación de Intentos Fallidos de Inicio de Sesión.

- Queremos limitar el número de intentos fallidos de inicio de sesión para proteger las cuentas de usuarios y prevenir ataques de fuerza bruta. Para ello deberá crear: **id, email, fecha_intento, bloqueado**.

Tabla de Logeos Fallidos:

- Crear una tabla en la base de datos para registrar los intentos fallidos.
-

Contador de Intentos:

Cada vez que un usuario ingrese credenciales incorrectas, incrementamos un contador asociado a ese usuario. Si el contador supera un umbral predefinido (por ejemplo, 3 intentos), bloqueamos la cuenta del usuario por un período determinado.

Desbloqueo Automático:

Implementar un mecanismo para desbloquear automáticamente las cuentas después de un período de tiempo.

RECURSOS INTERACTIVOS

“Aprende a crear un sistema de autenticación!

¡En esta sección encontrarás una tutorial interactivo para crear un sistema de autenticación profesional en PHP y MySQL!"

Un tutorial interactivo para crear un sistema de autenticación profesional en PHP y MySQL. Sigue el enlace de abajo:

https://www.youtube.com/watch?v=ZNbcmbVs5_M

REFERENCIAS CONSULTADAS

Alura Latam. (2023, 21 de abril). Tipos de autenticación: contraseña, token, JWT, dos factores y más. <https://www.aluracursos.com/blog/tipos-de-autenticacion>

aulaClic. (s.f.). 1 - Introducción a HTML. Etiquetas. https://www.aulaclic.es/html/t_1_2.htm

Amazon Web Services, Inc. (s.f.). ¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS. <https://aws.amazon.com/es/what-is/javascript/>

Codezta. (s.f.). Estructurado HTML5. [Imagen]. [se quitó una URL no válida]

Comunicación, D. (2019, julio 26). ¿Qué es Backend y Frontend? Descubre Comunicación. <https://descubrecomunicacion.com/que-es-backend-y-frontend/>

Cocosolution.com; Coco Solution. (2023, marzo 29). Qué es el Frontend y Backend y sus características.

<https://cocosolution.com/blog/que-es-frontend-y-backend/>
colaboradores de Wikipedia. (2024, 22 de abril).

PhpMyAdmin. <https://es.wikipedia.org/wiki/PhpMyAdmin>

Ecdisis Estudio. (2021, 30 de julio). ¿Qué es frontend y backend? <https://ecdisis.com/que-es-frontend-y-backend/>

Gallo, V. L. (2022, 10 de marzo). Eventos con JavaScript. Computer Science Master. <https://www.computersciencemaster.com.br/eventos-com-javascript/>

Invarato, R. (2019, septiembre 1). Cliente vs Servidor.

Jarroba. <https://jarroba.com/cliente-vs-servidor/>

Juanexplosions. (2021, 16 de marzo). 44 etiquetas HTML que debes conocer. Platzi. <https://platzi.com/blog/etiquetas-html-debes-conocer/>

Londoño, P. (2023, enero 4). Qué son las aplicaciones web y 8 ejemplos. Hubspot.es. <https://blog.hubspot.es/website/que-es-aplicacion-web>

Mercado Libre. (s.f.). Com.mx. Recuperado el 9 de agosto de

2024, de <https://www.mercadolibre.com.mx/>

Método Get y Post. (2016, 8 de junio). PROGRAMACION 7 universidad latina sede Azuero. <https://programacion7ulatsaavas.wordpress.com/2016/06/07/metodo-get-y-post/>

Pavlik, V. (2023, 14 de junio). HTML semántico: qué es y cómo usarlo correctamente. Semrush Blog. <https://es.semrush.com/blog/html-semantico/>

PHP GET y POST – Programa en Línea. (s.f.). <https://www.programaenlinea.net/php-get-post/>

Solution, C. (2023, marzo 29). Qué es el Frontend y Backend y sus características. Cocosolution.com; Coco Solution. <https://cocosolution.com/blog/que-es-frontend-y-backend/>

Shutterstock.com. (s.f.). Recuperado el 7 de agosto de 2024, de <https://www.shutterstock.com/es/image-vector/click-here-button-hand-pointer-clicking-1934677607>

Funcionamiento del sistema de autenticación en PHP. (s.f.). DesarrolloWeb.com. <https://desarrolloweb.com/articulos/1007.php>