LENGUAJE PHP

PHP es un [lenguaje de programación](https://rockcontent.com/es/blog/tipos-de-lenguaje-de-programacion/) destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los [servidores](https://rockcontent.com/es/blog/que-es-un-servidor/) y la [interfaz de usuario](https://rockcontent.com/es/blog/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.

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza mediaBOOTSTRAP

Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario. ¡Aprende todo sobre esta tecnología y cómo utilizarla!

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Font Awesome es una librería o kit de herramientas (toolkit) gráficas que son creadas para comportarse como fuentes. Los íconos de Font Awesome somo como fuentes de texto normales, pero en lugar de letras obtienes gráficos.

**VAINILLA DATABLES**

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Un complemento de tabla HTML de javascript ligero, extensible y sin dependencias. Similar a jQuery DataTables, pero sin las dependencias.

PASO DE NUESTRO CRUD

1. Descargar el motor de Base de datos. <https://www.apachefriends.org/es/download.html>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Una vez Instalado el xamp, ingresamos y Inicializamos los servicios de Xampp para los módulos de Apache y Mysql, es decir damos clic en Start.

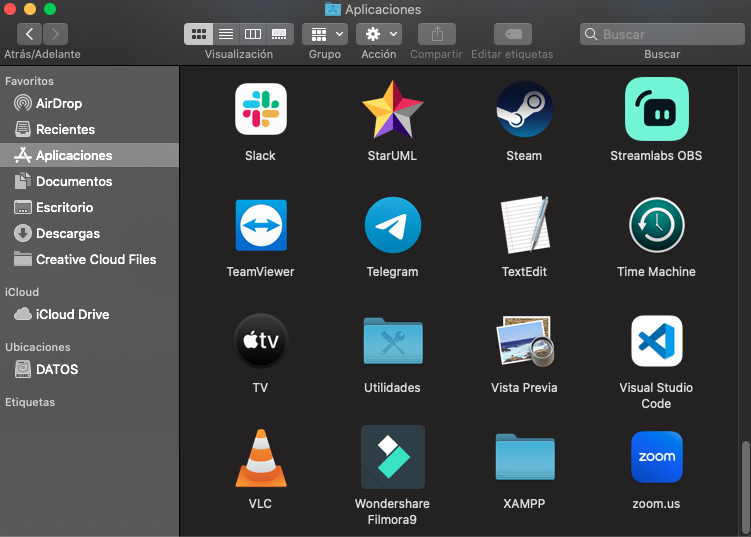
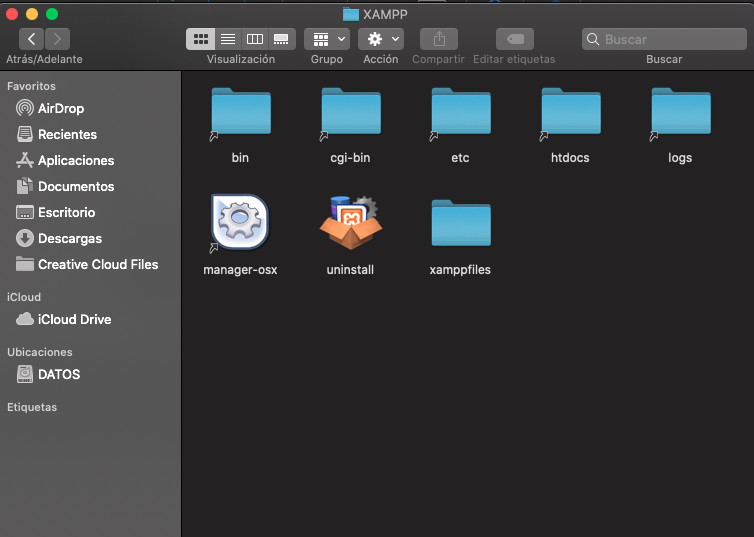
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso3 .** Debemos crear la carpeta de nuestro proyecto en la siguiente ruta:

Aplicaciones – buscamos la carpeta XAMPP – dentro de la carpeta XAMPP buscaremos htdocs-y dentro de esta carpeta creamos la de este proyecto: ADSO\_1

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Paso 4 .** Debemos crear nuestra base de datos, en XAMPP

Ingresamos <http://localhost/phpmyadmin/>

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Damos clic en “Nueva” en el panel izquierdo, le asignamos el nombre que usaremos en este caso será “banco” y en la lista desplegable usamos el utf8mb4\_unicode\_ci (es una codificación que permite almacenar y procesar texto en varios idiomas, incluyendo caracteres de 4 bytes como emojis, y realiza comparaciones de texto sin distinguir entre mayúsculas y minúsculas. Es una configuración comúnmente recomendada para bases de datos que necesitan manejar contenido multilingüe y diverso). Para que este importe sea exitoso también debemos asegurarnos que en el archivo sql esté debidamente asignada.

**Paso 5.** Creamos la tabla con 4 columnas: id, Nombre, email, teléfono

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Ingresamos nuestro datos de la tabla y nos deberá quedar de la siguiente manera:

Texto

Descripción generada automáticamente con confianza media

PASO 6: Ingresamos a Visual studio code y arrastramos la carpeta creada de nuestro proyecto y creamos la estructura como se observa en la Imagen.

Interfaz de usuario gráfica

Descripción generada automáticamente

adicionar.php

configurar.php

editar.php

excluir.php

index.php

inserir.php

Arrancamos con nuestra codificación en configurar.php

Texto

Descripción generada automáticamente

Este código en PHP se encarga de establecer una conexión a una base de datos MySQL utilizando la extensión PDO (PHP Data Objects).

Explicación del código:

$dsn: Es una cadena de conexión que especifica el tipo de base de datos, el nombre de la base de datos y la ubicación del host. En este caso, está configurado para MySQL, con el nombre de la base de datos "banco" y el host "localhost".

$dbuser: Es el nombre de usuario para acceder a la base de datos. En este caso, se establece como "root".

$dbpass: Es la contraseña correspondiente al nombre de usuario. En este caso, está vacía (""), lo que significa que no se proporciona una contraseña.

try: Se inicia un bloque "try-catch", que permite manejar excepciones (errores) que puedan ocurrir dentro del bloque de código.

$pdo = new PDO($dsn, $dbuser, $dbpass): Aquí se crea un objeto PDO que representa la conexión a la base de datos. Se pasan los parámetros de conexión (cadena DSN, nombre de usuario y contraseña) al constructor de PDO. Si la conexión tiene éxito, $pdo contendrá la instancia de la conexión a la base de datos.

catch (PDOException $e): Si ocurre una excepción de tipo PDOException (que representa errores relacionados con la base de datos), se captura en este bloque catch.

echo "falhou: ".$e->getMessage();: Si se captura una excepción, este bloque se ejecutará y mostrará un mensaje de error. La función getMessage() se utiliza para obtener el mensaje de error asociado con la excepción.

Seguiremos con nuestro index.php

Texto

Descripción generada automáticamente

Este fragmento de código corresponde a la estructura básica de una página web en HTML. Aquí está el desglose de las etiquetas y atributos utilizados:

<!doctype html>: Esto define el tipo de documento como un documento HTML5. Indica al navegador cómo debe interpretar y renderizar el contenido que sigue.

<html lang="es">: Esta etiqueta <html> inicia el elemento raíz de un documento HTML. El atributo lang se utiliza para especificar el idioma del contenido. En este caso, se establece en "es" para español.

<head>: Aquí comienza la sección de la cabeza del documento HTML. La cabeza contiene metadatos y enlaces a recursos externos como hojas de estilo, scripts, fuentes, etc.

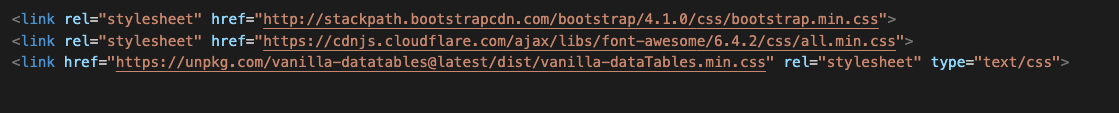
<meta charset="utf-8">: Esta etiqueta meta establece la codificación de caracteres del documento. En este caso, se utiliza UTF-8, que es una codificación que admite una amplia gama de caracteres y es compatible con muchos idiomas.

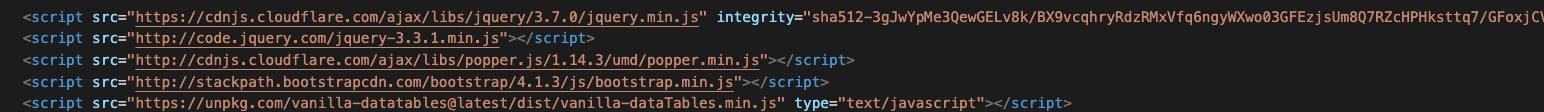
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">: Esta etiqueta meta controla cómo se muestra la página en dispositivos móviles y otros dispositivos con diferentes tamaños de pantalla. Los atributos aquí tienen los siguientes significados:

width=device-width: Hace que el ancho de la página sea igual al ancho de la pantalla del dispositivo.

initial-scale=1: Establece el nivel de zoom inicial en 1, lo que significa que la página no se ampliará ni reducirá al cargarla.

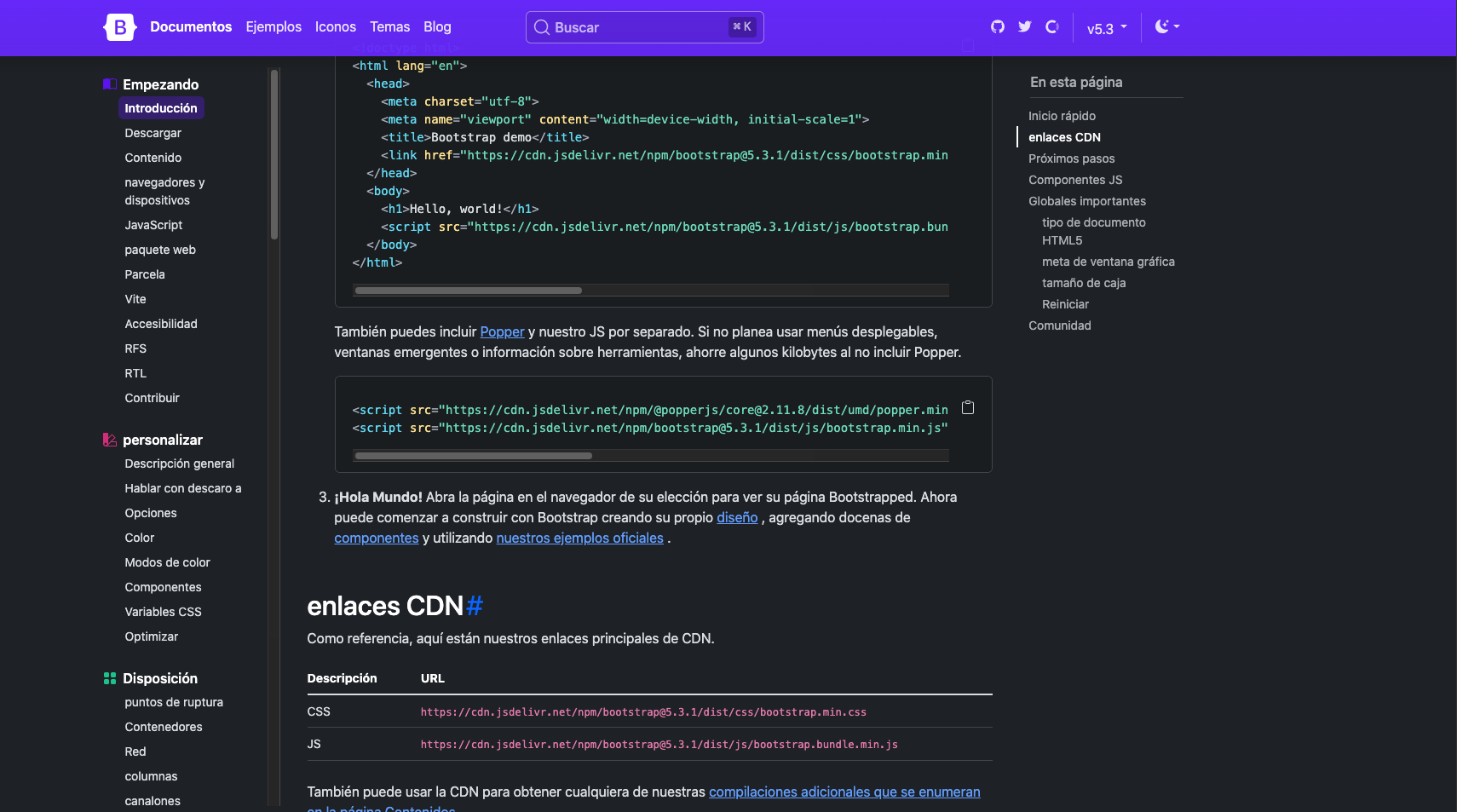
shrink-to-fit=no: Evita que el contenido se ajuste automáticamente al ancho de la pantalla. Esto puede ser útil para diseñar páginas que no cambien su diseño en dispositivos pequeños.





En esta parte van a escribir los estilos que vamos a traer de diferentes Librerías mencionadas al inicial este manual:

Ingresamos a la pagina oficial de <https://getbootstrap.com/docs/5.3/getting-started/introduction/#cdn-links>



Copiamos el css y el js y lo agregamos.

2. Ingresamos a <https://cdnjs.com/libraries/font-awesome>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

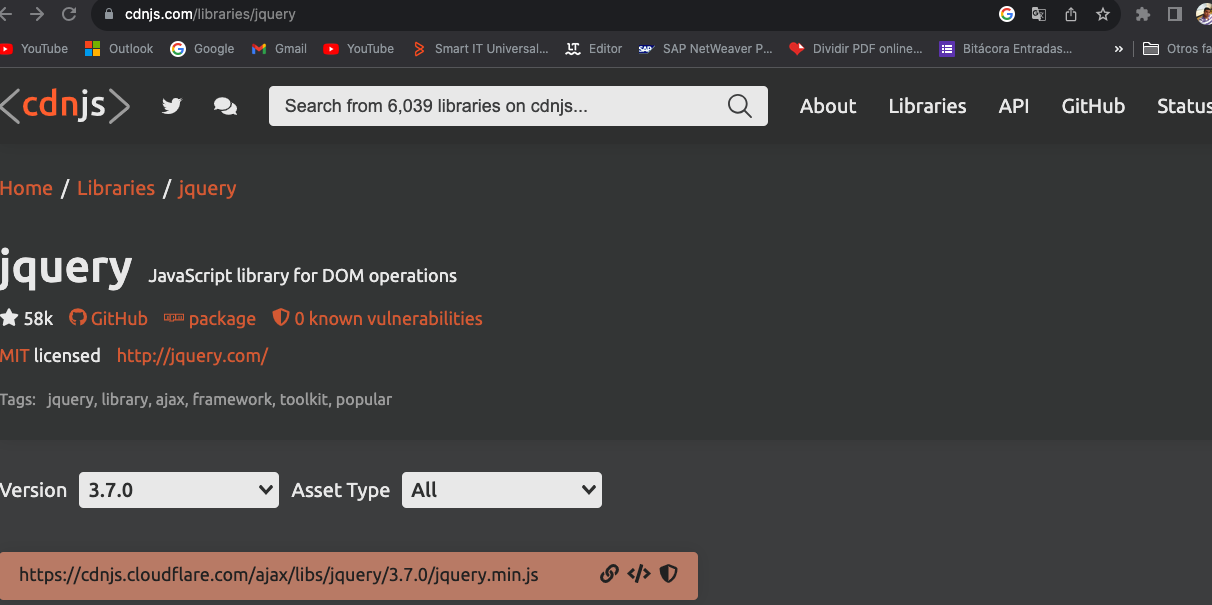
Esta librería nos permite ingresar iconos en cada botón que vamos a utilizar.

1. Ingresamos a <https://github.com/Mobius1/Vanilla-DataTables>



Esta librería nos sirve para crear al datables de búsqueda de nuestro CRUD

4 Ingresamos a la siguiente link <https://cdnjs.com/libraries/jquery>



Copiamos y lo agregamos en nuestro script en la parte final.

Una vez ingresado los link a nuestro proyecto seguiremos con nuestro código

Imagen que contiene Texto

Descripción generada automáticamente

En este código vamos a estructurar lo que va hacer nuestra tabla donde va ir la información que vamos ingresando.

**Explicación de código :**

**<?php require 'config.php'; ?>**

Esta línea de código está utilizando la instrucción require para incluir el contenido del archivo "config.php" en este punto del código. La instrucción require se utiliza para insertar contenido de un archivo externo en el lugar donde se encuentra. En este caso, se está incluyendo el contenido del archivo "config.php". Esto se utiliza a menudo para importar variables, configuraciones y funciones desde archivos externos para su uso en el archivo actual.

Este bloque de código HTML representa una estructura de interfaz de usuario (UI) que podría estar mostrando una lista de usuarios en una tabla. Aquí tienes una explicación de las partes clave:

<div class="container-fluid">: Un contenedor fluido de Bootstrap que ajustará su ancho para adaptarse al tamaño de la pantalla.

<div class="col-md-12 mb-2 mt-2">: Una columna de ancho completo (12 columnas en Bootstrap) con márgenes inferiores y superiores de 2 unidades.

<div class="row justify-content-center">: Una fila que contiene elementos centrados horizontalmente.

<button class="btn btn-primary " data-toggle="modal" data-target="#exampleModal">...: Un botón estilizado con la clase btn btn-primary. También se define que este botón activará un modal (ventana emergente) con la propiedad data-toggle y data-target.

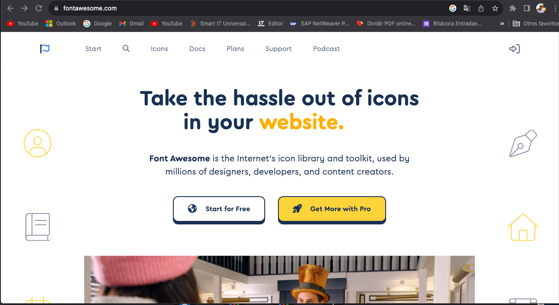
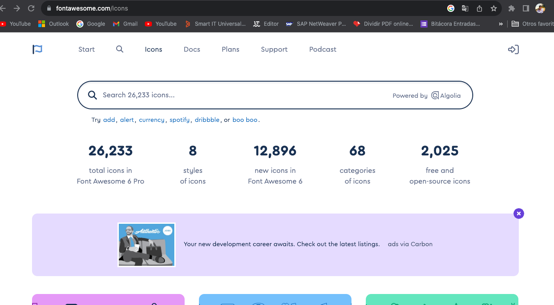
<table class="table mt -10" id="tabla">: Una tabla con la clase table y un margen superior de -10 unidades (no es una configuración estándar en Bootstrap). Se le asigna el id "tabla".

<thead> ... </thead>: La sección de encabezado de la tabla, donde se definen las columnas.

<tbody> ...: El cuerpo de la tabla, donde se agregarán las filas correspondientes a los datos de los usuarios.

Cuando vayamos en la línea del crear el botón Adicionar Nuevo usuario, ingresamos a la siguiente página para agregar el icono.

<https://fontawesome.com/>



Seleccionamos en la parte de arriba y buscamos el icono que queramos en este caso buscamos save y no saldrá como la forma de un disket lo seleccionamos y nos saldrá de esta manera:

Interfaz de usuario gráfica

Descripción generada automáticamente

Copiamos el código y lo pegamos antes del título de nuestro botón es decir de esta manera.



Este proceso lo realizamos en todo los botones que vamos a utilizar en nuestro proyecto.

Texto

Descripción generada automáticamenteSeguimos con el siguiente código



Este código lo que hacer es que ingresa la información en las filas que vamos ingresando con el botón de Agregar

Explicación del código.

Estas líneas verifican si la consulta recuperó algún resultado (rowCount() devuelve el número de filas afectadas por la última operación de consulta). Si hay resultados, se itera a través de cada fila recuperada utilizando un bucle foreach.

Dentro del bucle, se genera una nueva fila (<tr>) para cada usuario en la tabla. Cada columna de la fila (<td>) muestra el valor correspondiente del campo del usuario.

El cuarto <td> contiene dos botones:

Un botón de "Editar" que, al hacer clic, abrirá un modal (ventana emergente) para editar el usuario.

Un botón "Eliminar" que redirige al usuario a "excluir.php" con el id del usuario en la URL. Antes de realizar la eliminación, se muestra un mensaje de confirmación utilizando JavaScript.

En resumen, este fragmento de código recupera información de usuarios de una base de datos y genera filas de tabla HTML con detalles de usuario, incluyendo opciones para editar y eliminar usuarios.

**Continuamos con el código**

****

Este código representa la ventana modal o flotante que vamos a utilizar para nustro proyecto.

Lo primero es Ingresar a <https://getbootstrap.com/>

En la parte inferior escribimos Modal

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Seleccionamos el Live demo y copiamos el código que se encuentra

En la estructura y lo pegamos, y realizaremos una configuraciones en el código que se encuentra en la descripción de arriba.



Realizaremos algunas configuraciones par el botón editar aquí explicaremos el código.

Dentro del modal, se define la apariencia y el contenido del formulario de edición. Aquí está el desglose:

<div class="modal-dialog" role="document">: Define la apariencia del modal y su comportamiento en relación con el contenido de la página.

<div class="modal-content">: Contiene todo el contenido del modal, incluyendo el encabezado, el cuerpo y el pie.

<div class="modal-header">: Encabezado del modal que contiene el título y un botón de cierre (&times; representa una "X").

<div class="modal-body">: Cuerpo del modal que contiene el formulario de edición.

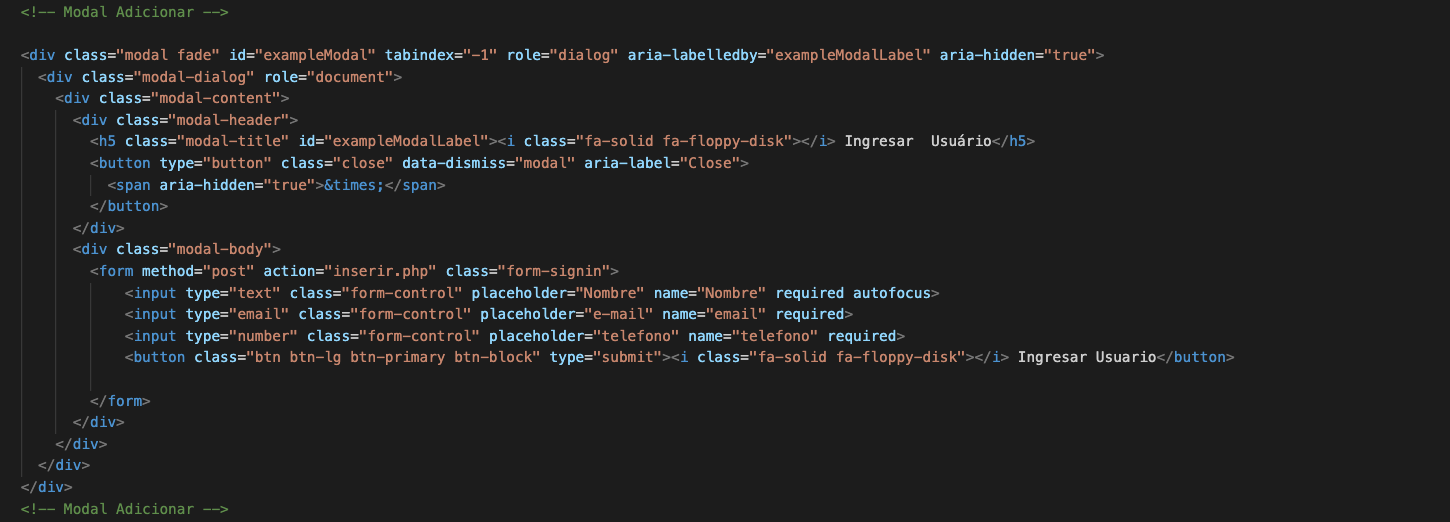
<form action="editar.php" method="post" class="form-signin">: Formulario que envía datos a "editar.php" utilizando el método POST. Se aplica la clase "form-signin".

<input id="id" class="form-control" value="'.$usuario['id'].'" name="id" type="hidden">: Un campo oculto que almacena el ID del usuario para la actualización.

<input type="text" class="form-control" placeholder="Nombre" name="Nombre" value="'.$usuario['Nombre'].'" required autofocus>: Campos de entrada para editar el nombre, correo electrónico y teléfono del usuario. El valor se prellena con los datos existentes.

<button class="btn btn-lg btn-primary btn-block" type="submit">Actualizar datos</button>: Botón para enviar el formulario y actualizar los datos del usuario.

De igual manera lo hacemos con el botón de adicionar usuarios.

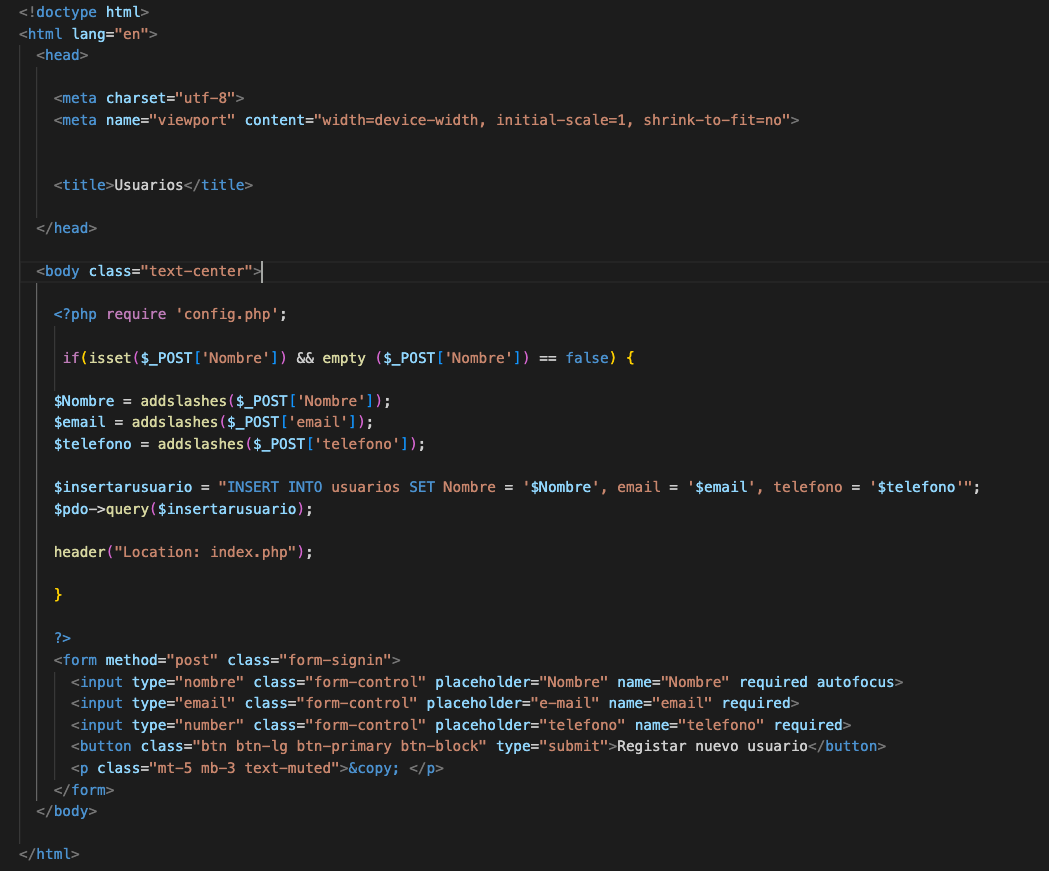


Para terminar nuestro index,php realicemos un código de script para que nos funcione la aparecía de datatables

Texto

Descripción generada automáticamente

Continuamos con adicionar.php



Este es un formulario HTML y un fragmento de código PHP que interactúan para permitir a los usuarios registrar nuevos usuarios en una base de datos.

Explicación

Esta sección de código PHP incluye el archivo "config.php" para establecer la conexión con la base de datos. Luego, verifica si se envió el formulario ($\_POST['Nombre'] existe) y si el campo "Nombre" no está vacío.

Si se cumplen estas condiciones, se recopilan y sanitizan los datos del formulario. Luego, se construye una consulta SQL para insertar los datos en la tabla "usuarios".

La consulta se ejecuta utilizando $pdo->query($insertarusuario) para insertar el nuevo usuario en la base de datos.

Finalmente, se redirige al usuario a la página "index.php" después de realizar la inserción

EDITAR PHP

Texto

Descripción generada automáticamente

Este fragmento de código PHP está destinado a manejar la actualización de información de un usuario en la base de datos.

EXPLICACION DEL CODIGO

Incluye el archivo "config.php" para establecer la conexión con la base de datos.

Inicializa la variable $id en 0. Esto se hace para asegurarse de que haya un valor válido para la variable en caso de que el formulario no haya sido enviado.

Verifica si el formulario ha sido enviado ($\_POST['Nombre'] existe) y si el campo "Nombre" no está vacío.

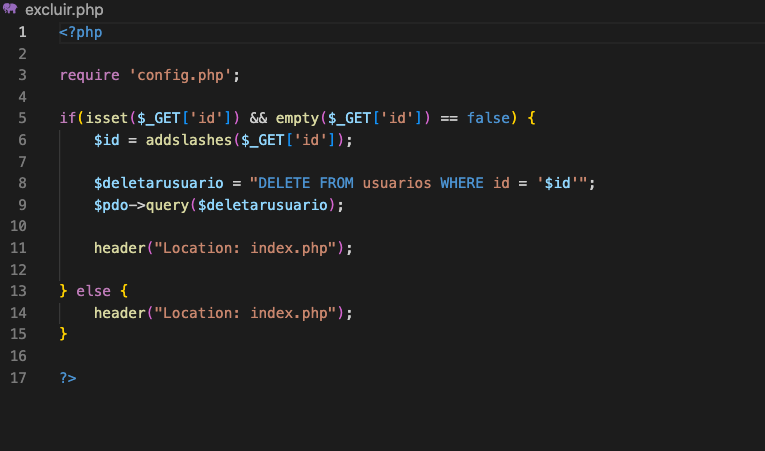
Si se cumplen las condiciones, se recopilan y sanitizan los datos del formulario, incluyendo el id del usuario que se pasa a través del formulario. Las variables $Nombre, $email y $telefono contienen los nuevos valores.

Se construye una consulta SQL utilizando la información recopilada para actualizar el registro del usuario en la base de datos utilizando la sentencia UPDATE.

La consulta se ejecuta utilizando $pdo->query($actualizarusuario) para llevar a cabo la actualización.

Después de la actualización, el código redirecciona al usuario a la página "index.php" para mostrar la lista actualizada de usuarios.

EXCLUIR PHP



Este fragmento de código PHP maneja la eliminación de un usuario de la base de datos.

EXPLICACION

ncluye el archivo "config.php" para establecer la conexión con la base de datos.

Verifica si se ha pasado un parámetro "id" en la URL utilizando $\_GET['id'], y si ese parámetro no está vacío.

Si se cumple la condición anterior, se recopila y se sanitiza el valor del parámetro "id" utilizando addslashes($\_GET['id']).

Se construye una consulta SQL para eliminar el usuario correspondiente a ese "id" de la tabla "usuarios" utilizando la sentencia DELETE.

La consulta se ejecuta utilizando $pdo->query($deletarusuario) para eliminar el registro del usuario.

Luego de la eliminación, se redirige al usuario de regreso a la página "index.php" para mostrar la lista actualizada de usuarios.

Si no se encuentra un parámetro "id" en la URL, o si el valor está vacío, se redirige al usuario a la página "index.php" sin realizar ninguna acción.

INSERIR PHP



Este fragmento de código PHP se encarga de manejar la inserción de un nuevo usuario en la base de datos.

**EXPLICACION**

Incluye el archivo "config.php" para establecer la conexión con la base de datos.

Verifica si el formulario ha sido enviado ($\_POST['Nombre'] existe) y si el campo "Nombre" no está vacío.

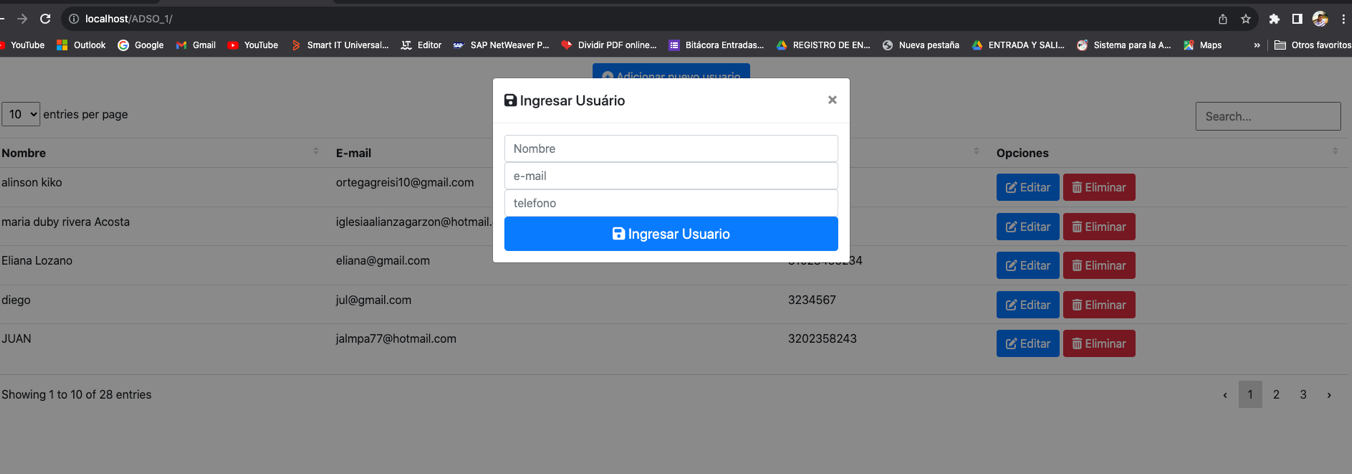
Si se cumplen las condiciones, se recopilan y sanitizan los datos del formulario. La variable $senha se inicializa, pero no se utiliza en este fragmento.

Se construye una consulta SQL utilizando los datos del formulario para insertar un nuevo usuario en la tabla "usuarios" utilizando la sentencia INSERT INTO.

La consulta se ejecuta utilizando $pdo->query($inserirusuario) para insertar el nuevo registro del usuario en la base de datos.

Luego de la inserción exitosa, se redirige al usuario a la página "index.php" para mostrar la lista actualizada de usuarios.

ASI DEBE DE QUEDAR NUESTRO PROYECTO DEL CRUD



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente