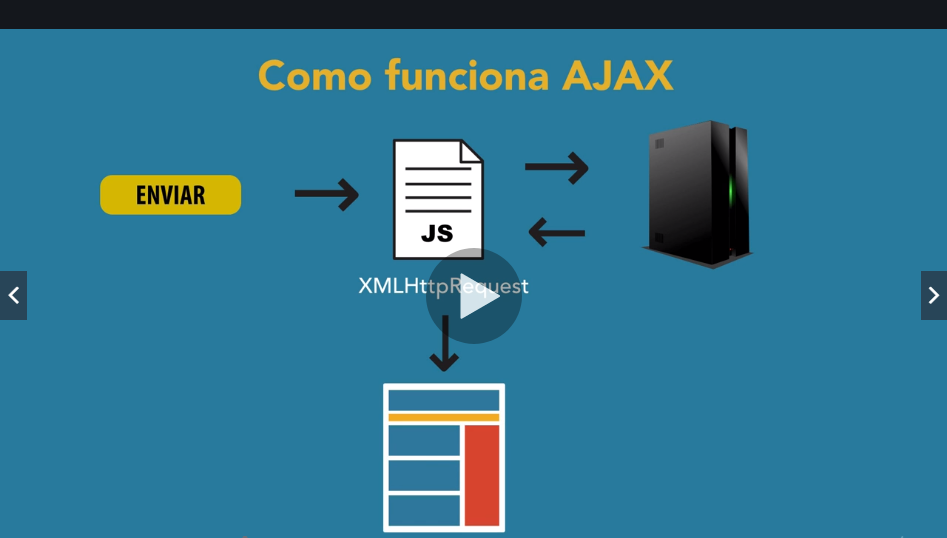
**Ajax**

Ajax es un conjunto de tecnologías que nos permiten realizar consultas al servidor en el background de la página, sin la necesidad de que esta tenga que actualizarse. Se maneja con archivos xml, jason, etc.



Como vemos en la imagen, en un ejemplo de enviar un formulario, el contacto con el servidor es intervenido por el archivo xml, que el cual recibe y envía información. De esta forma no se necesita intervenir la página y de esta forma actualizarse.

XMLHttpRequest

Este es el objeto con el que vamos a intercambiar datos detrás de escena. Se suele crean en el lenguaje JavaScript con el código:

**Var xhr = new XMLHttpRequest();**

**Métodos de XHR**

**new XMLHttpRequest();:** Crea un nuevo request.

**new FormData();:** Crea un nuevo request, pero este esta direccionado solo a formularios.

**.abort():** Cancela el request.

**.open():** Se abre la conexión y le pasas tus parámetros.

**.send():** Envia el request al servidor.

**.getAllResponseHeaders():** Retorna información del header.

**.setRequestHeader:** Agrega valores al header que se desea enviar.

Ejemplo de envio Ajax en un formulario:

**Var xhr = new XMLHttpRequest();**

**xhr.open(“POST** o **GET”, “archivo.php”, “true”);**

**xhr.setRequestHeader(“Content-type”, “application/x-www-form-urlencoded”)**

**xhr.send(“**parametros del form**”);**

**\***Dado que PHP no posee objetos de por sí, sino arreglos asociativos, pero en cambio en javascript estos si están, lo que se hace es utilizar un lenguaje de transporte como lo es **JSON**. Esto nos permite, en un ejemplo, recibir datos de un formulario en PHP por POST, pasarlos a **JSON** y luego leerlos con Ajax en Javascript.

**Post o Get**

**POST** es utilizado para enviar o intercambiar información.

**GET** es utilizado para recibir información.

**Propiedades del XHR**

**readyState:** Estado del xhr request, nos regresa un valor numérico (0- request sin iniciar, 1 – conexión establecida, 2 - request recibida, 3 – procesando el request, 4 – request finalizado y respuesta lista).

**responseText:** Retorna la respuesta com un string.

**responseXML:** Retorna la respuesta como un xml.

**status:** en los response aparece y define el estado del request. Sus números suelen ser (200 – Correcto, 403 – prohibido, 404 – no ecnontrado)

**onreadyStateChange:** Define una función que deberá ser llamada cuando cambien el estado del request, es decir cuando **readyState** cambie. Ejemplo:

**xhr.onreadyStateChange = function() {**

**if(xhr.readyState = 4 && xhr.status == 200) {**

**var divUsuario = document.getElementById(“registro”);**

**divUsuario.innerHTML = xhr.responseText;**

**}**

**}**

Se realiza un cambio en el html si se realiza la request de forma correcta.

**Ejemplos**

Archivo HTML de los dos:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title>Ajax</title>

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css" />

        <link rel="stylesheet" href="css/estilos.css" />

    </head>

    <body>

        <div class="contenedor">

            <h1>Aprendiendo AJAX</h1>

            <button id="cargar" value="cargar">Cargar</button>

            <div id="contenido"></div>

        </div>

        <!-- <script src="js/05.js"></script> -->

        <script src="js/06.js"></script>

    </body>

</html>

\*05.js es el text y 06.js es el php. Se comentara cada uno dependiendo el que usamos

**Ejemplo Con archivo Text**

Archivo Javascript:

// pasos para la creación de un request

var btnCargar = document.getElementById("cargar");

function cargarContenido() {

    // crearlo

    var xhr = new XMLHttpRequest();

    // abrirlo

    xhr.open("GET", "texto\_prueba.txt", true);

    // revisar que cambie

    xhr.onreadystatechange = function () {

        //console.log(xhr.readyState);

        if (xhr.readyState == 4 && xhr.status == 200) {

            //console.log("Se cargo correctamente");

            var contenido = document.getElementById("contenido");

            contenido.innerHTML = xhr.responseText;

        }

    };

    xhr.send();

}

btnCargar.addEventListener("click", cargarContenido);

* El archivo text solo tiene simple texto.

**Ejemplo Con archivo PHP**

Archivo PHP:

<?php

    $tecnologias = array(

        "frontend" => "html5, css3, javascript",

        "backend" => "php, mysql, apache",

        "fullstack" => "html5, css3, javascript, php, mysql, apache"

    );

    echo json\_encode($tecnologias);

?>

Archivo Javascript:

// pasos para la creación de un request

var btnCargar = document.getElementById("cargar");

function cargarContenido() {

    // crearlo

    var xhr = new XMLHttpRequest();

    // abrirlo

    xhr.open("GET", "servidor.php", true);

    // revisar que cambie

    xhr.onreadystatechange = function () {

        //console.log(xhr.readyState);

        if (xhr.readyState == 4 && xhr.status == 200) {

            //console.log("Se cargo correctamente");

            var json = JSON.parse(xhr.responseText);

            var contenido = document.getElementById("contenido");

            contenido.innerHTML = json.fullstack;

        }

    };

    xhr.send();

}

btnCargar.addEventListener("click", cargarContenido);

**Ejemplo: Enviar los datos de un Formulario a una base de datos**

**O editarlos**

Archivo Formulario (HTML o PHP):

<?php

    include "inc/funciones/funciones.php";

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Agenda de Contactos</title>

    <link rel="stylesheet" href="https://necolas.github.io/normalize.css/8.0.0/normalize.css">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.4.1/css/all.css" integrity="sha384-5sAR7xN1Nv6T6+dT2mhtzEpVJvfS3NScPQTrOxhwjIuvcA67KV2R5Jz6kr4abQsz" crossorigin="anonymous">

    <link href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="contenedor-barra">

    <h1>Agenda de Contactos</h1>

</div>

<div class="bg-amarillo contenedor sombra">

    <form id="contacto" action="#">

        <legend>Añada un Contacto<span>Todos los campos son obligatorios</span></legend>

        <?php include "inc/layout/formulario.php"; ?>

    </form>

</div>

<div class="bg-blanco contenedor sombra contactos">

    <div class="contenedor-contactos">

        <h2>Contactos</h2>

        <input type="text" id="buscar" class="buscador sombra" placeholder="Buscar Contactos...">

        <p class="total-contactos">

            <span>2 </span>Contactos

        </p>

        <div class="contenedor-tabla">

            <table id="listado-contactos" class="listado-contactos">

                <thead>

                    <tr>

                        <th>Nombre</th>

                        <th>Empresa</th>

                        <th>Teléfono</th>

                        <th>Acciones</th>

                    </tr>

                </thead>

                <tbody>

                <?php $contactos = obtenerContactos();

                    if($contactos->num\_rows) {

                        foreach($contactos as $contacto) {?>

                        <tr>

                            <td><?php echo $contacto["nombre"]?></td>

                            <td><?php echo $contacto["empresa"]?></td>

                            <td><?php echo $contacto["telefono"]?></td>

                            <td>

                                <a class="btn-editar btn" href="editar.php?id=<?php echo $contacto["id"]?>">

                                    <i class="fas fa-pen-square"></i>

                                </a>

                                <button data-id="<?php echo $contacto["id"]?>" type="button" class="btn-borrar btn">

                                    <i class="fas fa-trash-alt"></i>

                                </button>

                            </td>

                        </tr>

                    <?php }

                    } ?>

                </tbody>

            </table>

        </div>

    </div>

</div>

<script src="js/app.js"></script>

</body>

</html>

Archivo PHP Funcion Consulta a la Base De Datos

<?php

function obtenerContactos() {

    include "bd.php";

    try {

        return $conn->query("SELECT id, nombre, empresa, telefono FROM contactos");

    } catch (Exception $e) {

        echo "Error!" . $e->getMessage(). "</br>";

        return false;

    }

}

?>

Archivo Javascript:

const formularioContactos = document.querySelector("#contacto"),

    listadoContactos = document.querySelector("#listado-contactos tbody");

eventListeners();

function eventListeners() {

    // Cuando el formulario de crear o editar se ejecuta

    formularioContactos.addEventListener("submit", leerFormulario);

}

function leerFormulario(e) {

    e.preventDefault();

    // Leer los datos de los inputs

    const nombre = document.querySelector("#nombre").value,

        empresa = document.querySelector("#empresa").value,

        telefono = document.querySelector("#telefono").value,

        accion = document.querySelector("#accion").value;

    if (nombre === "" || empresa === "" || telefono === "") {

        mostrarNotificacion("Todos los campos son obligatorios", "error");

    } else {

        //Pasa la validacion, crear llamado a Ajax

        const infoContacto = new FormData();

        infoContacto.append("nombre", nombre);

        infoContacto.append("empresa", empresa);

        infoContacto.append("telefono", telefono);

        infoContacto.append("accion", accion);

        if (accion === "crear") {

            //craremos un nuevo elemento

            insertarBD(infoContacto);

        } else {

            //editar el contacto

        }

    }

}

// Tomar los datos del PHP JSON encode y trabajar con ellos

function insertarBD(datos) {

    //llamado a Ajax

    //Crear objeto

    const xhr = new XMLHttpRequest();

    //Abrir la conexion

    xhr.open("POST", "inc/modelos/modelo-contactos.php", true);

    //Pasar los datos

    xhr.onload = function () {

        if (this.status === 200) {

            const respuesta = JSON.parse(xhr.responseText);

            console.log(respuesta);

            //Insertar los elementos a la tabla

            const nuevoContacto = document.createElement("tr");

            nuevoContacto.innerHTML = `

                <td>${respuesta.datos.nombre}</td>

                <td>${respuesta.datos.empresa}</td>

                <td>${respuesta.datos.telefono}</td>

            `;

            //Crear contenido de los botones

            const contenedorAcciones = document.createElement("td");

            //Crear el icono de editar

            const iconoEditar = document.createElement("i");

            iconoEditar.classList.add("fas", "fa-pen-square");

            //Crea el enlace para editar

            const btnEditar = document.createElement("a");

            btnEditar.appendChild(iconoEditar);

            btnEditar.href = `

                editar.php?id=${respuesta.datos.id\_insertado}

            `;

            btnEditar.classList.add("btn", "btn-editar");

            //Agregarlo al padre

            contenedorAcciones.appendChild(btnEditar);

            //Crear el icono de eliminar

            const iconoEliminar = document.createElement("i");

            iconoEliminar.classList.add("fas", "fa-trash-alt");

            //Crear el enlace de eliminar

            const btnEliminar = document.createElement("button");

            btnEliminar.appendChild(iconoEliminar);

            btnEliminar.setAttribute("data-id", respuesta.datos.id\_insertado);

            btnEliminar.classList.add("btn", "btn-borrar");

            //Agregarlo al padre

            contenedorAcciones.appendChild(btnEliminar);

            //Agregarlo al tr

            nuevoContacto.appendChild(contenedorAcciones);

            //Agregarlo con los contactos

            listadoContactos.appendChild(nuevoContacto);

            //Resetar el Form

            document.querySelector("form").reset();

            //Mostrar notificacion

            mostrarNotificacion("Contacto Creado Correctamente", "correcto");

        } else {

            console.log("error");

        }

    };

    //Enviar los datos

    xhr.send(datos);

}

function mostrarNotificacion(mensaje, clase) {

    const notificacion = document.createElement("div");

    notificacion.classList.add(clase, "notificacion", "sombra");

    notificacion.textContent = mensaje;

    // Formulario

    formularioContactos.insertBefore(notificacion, document.querySelector("form legend"));

    // Ocultar y Mostrar la notificacion

    setTimeout(() => {

        notificacion.classList.add("visible");

        setTimeout(() => {

            notificacion.classList.remove("visible");

            setTimeout(() => {

                notificacion.remove();

            }, 500);

        }, 3000);

    }, 100);

}

* Se utiliza un FormData antes de llamar a Ajax dado de que este es mas común y mejor aplicable cuando se trata de un formulario.
* Dado a que el FormData es un objeto, debemos colocar los valores del formulario en el con una llave y un valor. Siendo el valor el **.value** de cada input.
* Para acceder a los valores dentro del objeto en un **console.log()** debemos hacer: **console.log(…varN)**

Archivo PHP JSON encode

<?php

 if($\_POST["accion"] == "crear") {

     //Crear un nuevo registro en la base de datos

     require\_once("../funciones/bd.php");

     $nombre = filter\_var($\_POST["nombre"], FILTER\_SANITIZE\_STRING);

     $empresa = filter\_var($\_POST["empresa"], FILTER\_SANITIZE\_STRING);

     $telefono = filter\_var($\_POST["telefono"], FILTER\_SANITIZE\_STRING);

     try {

        $stmt = $conn->prepare("INSERT INTO contactos (nombre, empresa, telefono) VALUES (?, ?, ?)");

        $stmt->bind\_param("sss", $nombre, $empresa, $telefono);

        $stmt->execute();

        if($stmt->affected\_rows == 1) {

            $respuesta = array(

                "respuesta" => "correcto",

                "datos" => array(

                    "nombre" => $nombre,

                    "empresa" => $empresa,

                    "telefono" => $telefono,

                    "id\_insertado" => $stmt->insert\_id

                )

            );

        }

        $stmt->close();

        $conn->close();

     } catch (Exception $e) {

        $respuesta = array(

             "error" => $e->getMessage()

        );

     }

     echo json\_encode($respuesta);

 }

?>

Archivo PHP Conexión Base de Datos

<?php

//Credenciales de las Bases de Datos

define("DB\_USUARIO", "root");

define("DB\_PASSWORD", "root");

define("DB\_HOST", "localhost");

define("DB\_HOST", "agendaphp");

$conn = new mysqli(DB\_HOST, DB\_USUARIO, DB\_PASSWORD, DB\_NOMBRE);

?>

Explicación Resumida:

Lo que estamos haciendo es primero en el HTML recibir los datos del usuario, luego PHP los capta con el POST y los manda a la base de datos, como a Ajax (con un json\_encode). Este último permite que estos se vean sin ser necesario que se recargue la página.

Pero el HTML debe también recibir el contenido a través de la base de datos ya que estos se borran solo con Ajax.