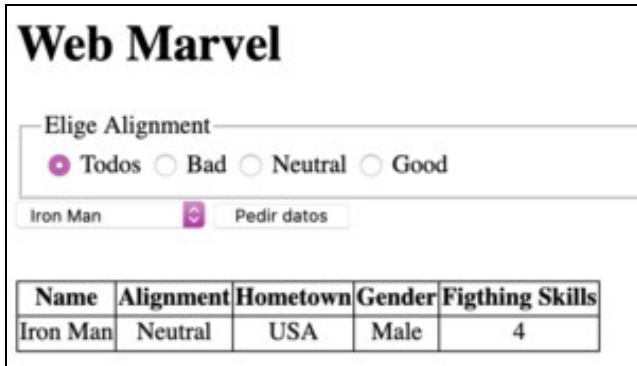


1 Solución a la mejora de la "Práctica 1 XMLHttpRequest ? GET + PHP ? MySQL"

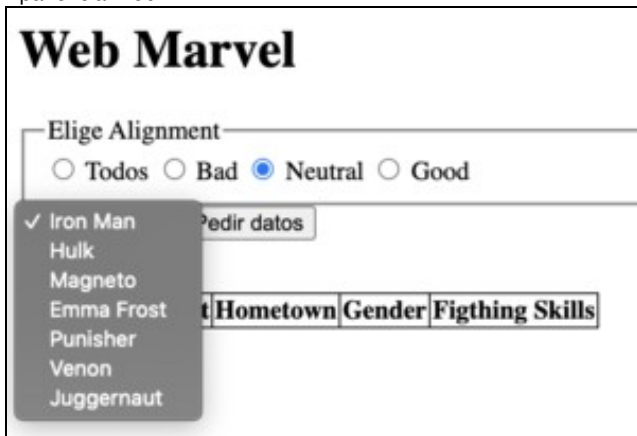
• Apariencia



The screenshot shows a web application titled "Web Marvel". It features a form with the heading "Elige Alignment" and four radio buttons: "Todos" (selected), "Bad", "Neutral", and "Good". Below the form is a dropdown menu with "Iron Man" selected and a "Pedir datos" button. At the bottom, there is a table with the following data:

Name	Alignment	Hometown	Gender	Fighting Skills
Iron Man	Neutral	USA	Male	4

Apariencia Web



This screenshot shows the same "Web Marvel" application, but with the "Neutral" radio button selected. A dropdown menu is open, listing several characters: Iron Man (checked), Hulk, Magneto, Emma Frost, Punisher, Venon, and Juggernaut. The "Pedir datos" button is visible to the right of the dropdown.

Apariencia Web

• HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="estilos.css">
  <title>Web Marvel</title>
</head>
<body>
  <h1>Web Marvel</h1>
  <form id="formulario">

    <fieldset id="alignments">
      <legend>Elige Alignment</legend>
      <label>
        <input type="radio" name="ralignment" value="Todos"> Todos
      </label>
      <label>
        <input type="radio" name="ralignment" value="Bad"> Bad
      </label>
      <label>
        <input type="radio" name="ralignment" value="Neutral"> Neutral
      </label>
      <label>
        <input type="radio" name="ralignment" value="Good"> Good
      </label>
    </fieldset>

    <select name="personajes" id="personajes"></select>
```

```

        <input type="button" value="Pedir datos" id="pedirDatos">
        <input type="button" value="Añadir datos" id="anadir">

    <table id="tabla">
        <thead>
            <th>Name</th>
            <th>Alignment</th>
            <th>Hometown</th>
            <th>Gender</th>
            <th>Figthing Skills</th>
        </thead>
    </table>
</form>

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

</body>
</html>

```

• PHP

```

//conexion.php
<?php
// Cabecera de resultado en JSON.
header('Content-Type: application/json');

// Constantes de configuración de la aplicación.

define('DB_SERVIDOR', 'localhost');
define('DB_PUERTO', '3306');
define('DB_BASEDATOS', 'marvel');
define('DB_USUARIO', 'marvel');
define('DB_PASSWORD', 'abc123');

try {
    $cadenaConexion = "mysql:host=" . DB_SERVIDOR . ";port=" . DB_PUERTO . ";dbname=" . DB_BASEDATOS . ";charset=utf8";
    $pdo = new PDO($cadenaConexion, DB_USUARIO, DB_PASSWORD);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    die("Error conectando a servidor de base de datos: " . $e->getMessage());
}

?>

//buscar.php
<?php

require_once 'conexion.php';

try {
    // Creamos un objeto para componer el resultado luego en JSON
    $objeto = new stdClass();

    if (isset($_GET['id']) && $_GET['id'] != '') {
        // Preparamos la consulta de SELECT
        $stmt = $pdo->prepare('select * from marvels where id=?');
        // Vinculamos los parámetros.
        $stmt->bindParam(1, $_GET['id']);
        // Ejecutamos la consulta
        $stmt->execute();

        // Si encontró el registro de la consulta inicial
        if ($stmt->rowCount() == 1) {
            $objeto->total = $stmt->rowCount();
            $objeto->resultados = $stmt->fetchAll();
        } else {
            $objeto->total = 0;
            $objeto->resultados = [];
        }
    }

    echo json_encode($objeto);
}

```

```

}

if (isset($_GET['alignment']) && $_GET['alignment'] != '') {
    if ($_GET['alignment'] != 'Todos') {
        $stmt = $pdo->prepare('select * from marvels where alignment=?');
        $stmt->bindParam(1, $_GET['alignment']);
    } else {
        $stmt = $pdo->prepare('select * from marvels');
    }

    // Ejecutamos la consulta
    $stmt->execute();
    $objeto->total = $stmt->rowCount();
    $objeto->resultados = $stmt->fetchAll();

    echo json_encode($objeto);
}
} catch (PDOException $error) {
    echo "Error: " . $error->getMessage();
}
?>

//insertar.php
<?php

require_once 'conexion.php';

try {
    // Creamos un objeto para componer el resultado luego en JSON
    $objeto = new stdClass();

    if (isset($_POST) && !empty($_POST)) {

        $errores = [];
        foreach ($_POST as $campo => $valor) {
            if (isset($_POST[$campo]) && $_POST[$campo] == '') {
                $errores[] = "El campo $campo no puede estar vacío.";
            }
        }

        if (isset($_POST['fighting_Skills']) && !filter_input(INPUT_POST, "fighting_Skills", FILTER_VALIDATE_INT)) {
            $errores[] = "El campo fighting_Skills no es un entero.";
        }

        if (count($errores) != 0) {
            // Componemos el objeto a devolver en JSON
            $objeto->status = 'error';
            $objeto->msg = $errores;
        } else {

            // NO ha habido errores insertamos los datos en la tabla.
            // Preparamos la consulta de INSERT
            $stmt = $pdo->prepare('insert into marvels(name,alignment,gender,hometown,fighting_Skills) values (?,?,?,?,:)');

            // Vinculamos los parámetros.
            $stmt->bindParam(1, $_POST['name']);
            $stmt->bindParam(2, $_POST['alignment']);
            $stmt->bindParam(3, $_POST['gender']);
            $stmt->bindParam(4, $_POST['hometown']);
            $stmt->bindParam(5, $_POST['fighting_Skills']);

            // Ejecutamos la consulta
            $stmt->execute();

            // Componemos el objeto a devolver en JSON
            $objeto->status = 'ok';
            $objeto->msg = 'Registro insertado con éxito con el ID: ' . $pdo->lastInsertId();
        }
    }

    // Enviamos el objeto en formato JSON como respuesta: con status= error | ok.
    echo json_encode($objeto);
}

```

```

} catch (PDOException $error) {
    echo "Error: " . $error->getMessage();
}

?>

function getData(id,alignment) {

    const numAleatorio = (new Date()).getTime();

    if (id == undefined) {
        const url = `buscar.php?alignment=${alignment}&r=${numAleatorio}`;
        // console.log(url);

        fetch(url)
            .then(res => res.json())
            .then(dataJSON => {
                // console.log(dataJSON);
                //Creamos un fragmento de código
                const fragmento = document.createDocumentFragment();

                personajes.innerHTML = "";
                for (const heroes of dataJSON.resultados) {
                    //<option value="1">Spider Man</option>
                    const option = document.createElement('option');
                    option.setAttribute('value', heroes.ID); //'ID' como está en la BD
                    option.textContent = heroes.name;
                    fragmento.appendChild(option);
                }

                personajes.appendChild(fragmento);
            })
            .catch(err => {
                console.log("Error con id=undefined: " + err.message);
            });

    } else {
        const url = `buscar.php?id=${id}&r=${numAleatorio}`;
        // console.log(url);

        fetch(url)
            .then(res => res.json())
            .then(dataJSON => {
                // console.log(dataJSON);

                //Creamos la fila de datos
                const aCampos = ['name', 'alignment', 'hometown', 'gender', 'fighting_Skills'];

                const tr = document.createElement('tr');
                for (const campo of aCampos) {
                    // console.log(campo);
                    // console.log(dataJSON.resultados[0][campo]);
                    const td = document.createElement('td');
                    td.textContent = dataJSON.resultados[0][campo];
                    tr.appendChild(td);
                }
                //Borramos el tr existente antes de poner el del elemento seleccionado
                if (tabla.children[1]) tabla.removeChild(tabla.children[1]);
                //Añadimos el elemento seleccionado a la tabla
                tabla.appendChild(tr);
            })
            .catch(err => {
                console.log('Error con id:' + err.message);
            });
    }
}

//Comprobamos cuál es el alignment y cambiamos contenido del select
function queAlignment() {
    let valueAlignment;
    for (const a of inputAlignment) {
        if (a.checked == true) valueAlignment = a.value;
    }
}

```

```

    }
    // console.log(valueAlignment);

    getData(undefined, valueAlignment);

}

// Seleccionamos elementos HTML
const formulario = document.getElementById('formulario');
const personajes = document.getElementById('personajes');
const botonPedirDatos = document.getElementById('pedirDatos');
const tabla = document.getElementById('tabla');
const mensaje = document.getElementById('mensaje');

//Pedimos los datos para rellenar la tabla
botonPedirDatos.addEventListener('click', pedirDatos, false);

function pedirDatos() {
    // console.log(personajes.selectedIndex); //Elemento seleccionado en el select
    const idSelect = personajes.children[personajes.selectedIndex].value;
    // console.log(idSelect);
    getData(idSelect);
}

//Seleccionamos el fieldset que tiene en su interior los elementos Radio
const fieldset = document.getElementById('alignments');
//Le añadimos un escuchador click que leerá cuál de los elementos esté checkeado
fieldset.addEventListener('click', queAlignment, false);

//Seleccionamos los elementos Radio con 'name' 'alignment'
const inputAlignment = document.getElementsByName('ralignment');
//Configuramos al inicio el 'input' 'alignment' 'Todos' en 'checked'
inputAlignment[0].checked = true;

//Al completar la carga del HTML
getData(undefined, inputAlignment[0].value);

////////////////////
// Modificación para añadir y editar registros
const botonAnadir = document.getElementById('anadir');
//Creamos el atributo estado= 'escribir' para escribir los datos a añadir
// estado='insertar' para ya insertarlos en la bd

botonAnadir.setAttribute('estado', 'escribir');
botonAnadir.addEventListener('click', anadirElemento, false);

// formulario.addEventListener('submit', e => {
//     e.preventDefault();

// }, false);

function anadirElemento() {
    if (this.getAttribute('estado') == 'escribir') {
        //Creamos la fila de datos
        const aCampos = ['name', 'alignment', 'hometown', 'gender', 'fighting_Skills'];

        //Creamos una fila de input text en la tabla
        const tr = document.createElement('tr');
        for (const campo of aCampos) {
            const td = document.createElement('td');
            const input = document.createElement('input');
            if (campo == 'fighting_Skills') {
                input.setAttribute('type', 'number');
            } else {
                input.setAttribute('type', 'text');
            }
            input.setAttribute('name', campo);
            input.setAttribute('class', 'cajaTexto'); //Para seleccionarlas todas juntas
            td.appendChild(input);
            tr.appendChild(td);
        }
    }
}

```

```

//Borramos el tr existente antes de poner el del elemento seleccionado
if (tabla.children[1]) tabla.removeChild(tabla.children[1]);
//Añadimos el elemento seleccionado a la tabla
tabla.appendChild(tr);

//Ponemos el foco en el primero de esos input
tabla.children[1].children[0].children[0].focus();

//Cambiamos el 'estado' a 'insertar'
this.setAttribute('estado', 'insertar')
} else {
  console.log('Escribimos en la bd');

  //Nuestro formulario se llama 'formulario'
  //Creamos un URLSearchParams() para configurar datos a enviar en formato
  const nuevosDatos = new URLSearchParams();
  for(const c of formulario.getElementsByClassName('cajaTexto')) {
    nuevosDatos.append(c.name, c.value);
  }
  // console.log(nuevosDatos.toString());
  fetch('insertar.php', {
    method: 'POST',
    body: nuevosDatos})
    .then(res => res.json())
    .then(res => {
      console.log(res);
      if(res.status == 'ok') {
        mensaje.innerText = 'Insertado con éxito';
        setTimeout(function () {
          mensaje.innerText = '';
        }, 1500); //No olvidar configurar el tiempo de pausa

      }
    })
    .catch(err => console.log(err));

  //Borramos el tr existente
  if (tabla.children[1]) tabla.removeChild(tabla.children[1]);

  //Cambios el 'estado' a 'escribir'
  this.setAttribute('estado', 'escribir');
}
}

```

[Volver](#)