**Focaccia**

**Link:** <https://js-modulo-02.web.app/index.html>

**JavaScript**: Aplicado en secciones:

<https://js-modulo-02.web.app/dia.html> 🡪 Manipulación DOM y vectores

<https://js-modulo-02.web.app/aleatorio.html> 🡪 JQuery, manipulación DOM y vectores

<https://js-modulo-02.web.app/contacto.html> --> Validación de formulario

**En la sección día:**



**Vectores y manipulación del DOM:**

Presenta el plato del día, uno por día cargado para una semana.

Un vector llamado “menu” contiene 7 elementos cada uno correspondiente a un día de la semana. Para conocer el día de la semana actual se utiliza el constructor Date():

var fecha = new Date();

*// Obtiene el dia representado en un entero. 0-->Domingo 6-->Sabado*

indiceMenuSeleccionado = fecha.getDay();

Una vez determinado el día se obtiene el plato del día, almacenado en el vector “menú”.

Con esta información se obtiene la foto, titulo e ingredientes para informar al cliente por si es alérgico a alguno, de los objetos fotoMenu, tituloMenu e ingredientes.

En este punto se hace uso del DOM para manipular el código HTML y el condicional do while para iterar la lista de ingredientes del plato del día:

*// Imprime foto en el id fotoAleatorio*

function imprimirFoto(idFoto, indiceMenuSeleccionado){

    var FotoAImprimir = fotoMenuSeleccionado(indiceMenuSeleccionado);

    var Foto = document.getElementById(idFoto);

    Foto.setAttribute('style','background-image: url(' + FotoAImprimir + ')');

}

*// Imprime titulo en el id tituloAleatorio*

function imprimirTitulo(idTitulo, indiceMenuSeleccionado){

    var tituloAImprimir = tituloMenuSeleccionado(indiceMenuSeleccionado);

    var titulo = document.getElementById(idTitulo);

    titulo.innerHTML = tituloAImprimir;

}

*// Imprime ingredientes en el id listaIngredientesAleatorio*

function imprimirIngredientes(idLista, indiceMenuSeleccionado){

    var ingredientesAImprimir = ingredientesMenuSeleccionado(indiceMenuSeleccionado);

    var lista = document.getElementById(idLista);

    var i = 0;

*do*{

        var listaItem = document.createElement('li');

        listaItem.innerHTML = ingredientesAImprimir[i];

        lista.appendChild(listaItem);

    i++;

    }*while*(i<ingredientesAImprimir.length)

**En la sección aleatorio:**

**Vectores y manipulación del DOM:**

Muestra un plato aleatorio cargado en el vector menú, para cada ingreso del usuario.

La lógica de funcionamiento en la manipulación del HTML con uso del DOM y uso del condicional do while es el mismo. Solo cambia que la obtención del plato ya no está ligada al día de la semana. Sino que se utiliza del módulo Math para obtener un número aleatorio y luego acotar el resultado, mínimo valor retornado 0 y máximo longitud del vector menu-1 ya que los vectores inician su índice en 0.

function seleccionMenuAleatorio() {

    var indice = Math.floor(Math.random() \* (menu.length-1));

*return* indice;

}

Para evitar que al manipular el DOM solo se muestre el ultimo ingrediente, ya que se “pisan”, se añade a cada ingrediente como un hijo mediante appendChild en el condicional do while.

*do*{

        var listaItem = document.createElement('li');

        listaItem.innerHTML = ingredientesAImprimir[i];

        lista.appendChild(listaItem);

    i++;

    }*while*(i<ingredientesAImprimir.length)

**JQuery y manipulación del DOM:**

****

Utilizando el método ready de JQuery, al terminar de cargar la pagina se ejecuta la función “jQueryFuncion” y “jQueryAleatorio”.

$(document).ready(jQueryFuncion);

$(document).ready(jQueryAleatorio);

jQueryFuncion asocia un handler a un evento tipo click sobre el botón “sorpréndeme”, dicho handler es jQueryAleatorio.

function jQueryFuncion(){

    $("#botonAleatorio").click(jQueryAleatorio);

    console.log("jQueryFuncion");

}

jQueryAleatorio se dispara al ocurrir un evento tipo click sobre el boton Sorpréndeme, toma un elemento aleatorio del vector menú utilizando la función “seleccionMenuAleatorio()”. En caso de recibir como nuevo elemento uno que es igual al ya mostrado en pantalla realiza una nueva petición de otro elemento aleatorio. Se utiliza el selector de id “#id” de JQuery y se lo oculta en primer lugar utilizando el método hide para poder luego mostrarlo con el efecto fadeIn, con un tiempo de difuminado de un segundo.

function jQueryAleatorio(){

    indiceMenuSeleccionado = seleccionMenuAleatorio();

*if*(tituloMenu[menu[indiceMenuSeleccionado]]==document.getElementById("tituloAleatorio").innerText){

        console.log("Repetido");

        jQueryAleatorio();

*return*;

        }

    imprimirFoto('fotoAleatorio', indiceMenuSeleccionado);

    $('#fotoAleatorio').hide();

    $('#fotoAleatorio').fadeIn(1000);

    imprimirTitulo('tituloAleatorio', indiceMenuSeleccionado);

    imprimirIngredientes('listaIngredientesAleatorio', indiceMenuSeleccionado);

    console.log("jQueryAleatorio");

}

**En la sección contacto:**

**Validación de formulario:**

Todos los campos son requeridos, para el campo teléfono se utiliza un patrón de validación:

pattern="[0-9]{8,16} 🡪 Indica que los caracteres ingresados deben ser números del 0 al 9, deben ser como mínimo 8 números y máximo 16.

Para el mensaje, posee una longitud de máximo 1000 caracteres.

Los campos son verificados y en caso de ser inválidos añaden al DOM un párrafo con un mensaje de error, cuando el usuario lo soluciona este desaparece automáticamente.



Los campos son validados al dispararse un evento tipo “change”:

*// Campo nombre*

nombre = document.getElementById("nombre");

nombre.addEventListener("change", validarNombre, false);

*// Campo apellido*

apellido = document.getElementById("apellido");

apellido.addEventListener("change", validarApellido, false);

*// Campo correo electrónico*

correo = document.getElementById("correo");

correo.addEventListener("change", validarCorreo, false);

*// Campo teléfono*

telefono  = document.getElementById("telefono");

telefono.addEventListener("change", validarTelefono, false);

*// Campo mensaje*

mensaje  = document.getElementById("mensaje");

mensaje.addEventListener("change", validarMensaje, false);

En caso de un intento de “submit” con campos invalidos se dispara el evento tipo “invalid”:

*// Formulario*

formulario  = document.getElementById("formularioCampos");

formulario.addEventListener("invalid", function(e){

    e.stopPropagation();

    validarNombre();

    validarApellido();

    validarCorreo();

    validarTelefono();

    validarMensaje();

}, true);

**Campos nombre y apellido siguen la misma lógica:**

*if*(document.getElementById("nombre").validity.valueMissing){}

Es verdadero cuando el campo nombre está vacío, esto sucede en todos los campos.

*if*(!document.getElementById("errorNombre")){}

Este condicional es utilizado para evitar que se añadan multiples mensajes de error, si no existe ningún mensaje de error crea uno. De lo contrario no modifica el DOM, esto sucede en todos los campos.

*if*(document.getElementById("errorNombre")){}

Al final de la validación si el campo nombre fue validado correctamente elimina el mensaje de error en el DOM, pero antes de realizar dicha acción verifica que exista dicho mensaje y así evitar errores, esto sucede en todos los campos.

function validarNombre (){

*if*(document.getElementById("nombre").validity.valueMissing){

*// Solo crea un elemento en el DOM si no existe el párrafo error*

*if*(!document.getElementById("errorNombre")){

            let mensajeError = document.createElement('p');

            mensajeError.setAttribute("class", "error");

            mensajeError.setAttribute("id", "errorNombre");

            mensajeError.innerHTML = "Debe indicar un nombre";

            document.getElementById("nombre").insertAdjacentElement("afterend", mensajeError);

            console.log("Campo errorNombre creado");

        }

        console.log("Campo nombre vacio");

    }

*else*{

        console.log("Campo nombre validado");

*// Elimino el párrafo de error del DOM*

*if*(document.getElementById("errorNombre")){

            document.getElementById("errorNombre").remove();

        }

    }

}

**Campos teléfono, correo y mensaje:**

Ambos poseen validación por ser requeridos y un segundo requerimiento extra.

Al no ser una única validación se utiliza mensajeErrorInnerHTML como un flag en el caso de ser invalido, en cualquier caso se llena la variable con el texto a mostrar al usuario. Al final de la validación si la variable tiene contenido distinto del null significará que no paso la validación y así se procede a modificar el DOM para mostrar el mensaje de error.

function validarTelefono (){

    let mensajeErrorInnerHTML = '';

*if*(document.getElementById("telefono").validity.valueMissing){

        mensajeErrorInnerHTML = "Debe indicar un teléfono";

        console.log("Campo telefono vacio");

        }

*else* *if*(document.getElementById("telefono").validity.patternMismatch){

        mensajeErrorInnerHTML = "Debe indicar un teléfono de mínimo 8 números" +

         " y 16 números como máximo";

        console.log("Campo teléfono no concuerda con el patron");

        }

*else*{

        console.log("Campo teléfono validado");

*// Elimino el párrafo de error del DOM*

*if*(document.getElementById("errorTelefono")){

            document.getElementById("errorTelefono").remove();

            }

        }

*// Existe error de validacion*

*if*(mensajeErrorInnerHTML){

*// Solo crea un elemento en el DOM si no existe el párrafo error*

*if*(!document.getElementById("errorTelefono")){

            mensajeError = document.createElement('p');

            mensajeError.setAttribute("class", "error");

            mensajeError.setAttribute("id", "errorTelefono");

            document.getElementById("telefono").insertAdjacentElement("afterend", mensajeError);

            console.log("Campo errorTelefono creado");

            }

*else*{

*// Si ya existe lo modifica*

            mensajeError = document.getElementById("errorTelefono");

        }

        mensajeError.innerHTML = mensajeErrorInnerHTML;

        console.log("Campo errorTelefono modificado");

        }

}