Desarrollo Web en Entorno Cliente



Formularios

|  |
| --- |
| VERÓNICA BONIS MARTÍN |
| MARIA CARMEN CORREA HERAS |
| ÁNGEL SÁNCHEZ-SIERRA CRUZ  JOSÉ MARÍA TENREIRO EIRANOVA |
| **REPOSITORIO GITHUB**  <https://github.com/ITTDAW/AE-3.-Formularios.git> |
|  |

**Requerimiento 1**

En esta actividad debes desarrollar un formulario completo, sin envío al servidor, que sirva para dar de alta un pedido de una pizza dentro de una web de una pizzería.

Los campos del formulario son los siguientes:

* Nombre
* Dirección
* Teléfono
* Email
* Un radio button con el tamaño de la pizza, pudiendo ser pequeña, mediana o grande
* 4 Checkbox con los diferentes ingredientes de la pizza
* Un botón de procesar el pedido

Todos los campos tienen que estar rellenos de tipo texto deben de estar rellenos para que sean válidos, además debe de elegir obligatoriamente un tamaño de la pizza y al menos un ingrediente para ella.

**Requerimiento 2**

El campo teléfono y email deben de tener un formato adecuado, deben de hacerse las validaciones oportunas con JavaScript (Expresiones regulares).

Además, el nombre no puede empezar por minúsculas.

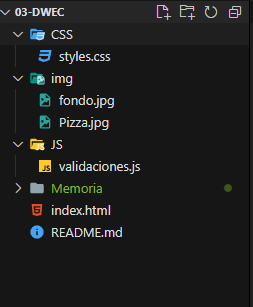
**Requerimiento 3**

Al pulsar el botón de procesar el pedido, se mostrará el precio total del pedido calculándolo en base a los siguientes parámetros:

* 5€ para la pizza pequeña
* 10€ para la pizza mediana
* 15€ para la pizza grande
* Cada ingrediente elegido tendrá un valor de 1€

**ORGANIZACIÓN DE DIRECTORIOS**

La organización de los directorios en nuestro proyecto del formulario de pedido de una pizzería será el siguiente:



* Tenemos dos carpetas llamadas CSS e img en donde almacenaremos nuestra hoja de estilos y nuestras imágenes respectivamente.
* Después tenemos una carpeta llamada JS que es donde almacenamos nuestro archivo Javascript para las validaciones.
* En la raíz del proyecto se encuentra nuestro archivo html index. En el index implementamos nuestro formulario de la pizzería con los requisitos exigidos en el enunciado.

**CUESTIONES CLAVE DE LA ORGANIZACIÓN DE DIRECTORIOS**

Hay dos cuestiones clave siempre en todo archivo html:

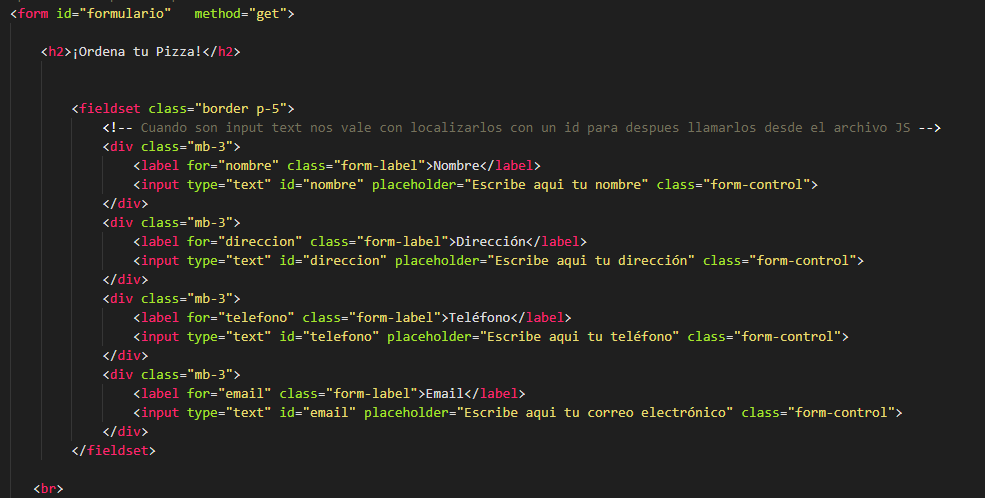
-La primera, hay que intentar que nuestro código html sea limpio, es decir, que no encontremos rastro de código javascript en éste. Esto debe de hacerse así, porque el cliente podría copiar nuestro código desde el navegador. Es cierto, que aun con esta medida de seguridad sigue siendo posible copiarlo, pero al menos se añade un nivel de dificultad.

- La segunda cuestión clave es que las validaciones siempre deben de hacerse tanto en el cliente como en el servidor. En la parte del servidor es donde el cliente no va a poder alterar nada de nuestro código. Es por eso que con unas validaciones solo en el cliente, nos exponemos a un problema de seguridad.

**ARCHIVO INDEX.HTML**

En la primera parte del formulario nos encontramos con los inputs de tipo text. Para poder llamarlos desde el archivo JS, nos vale definirlos con un id.

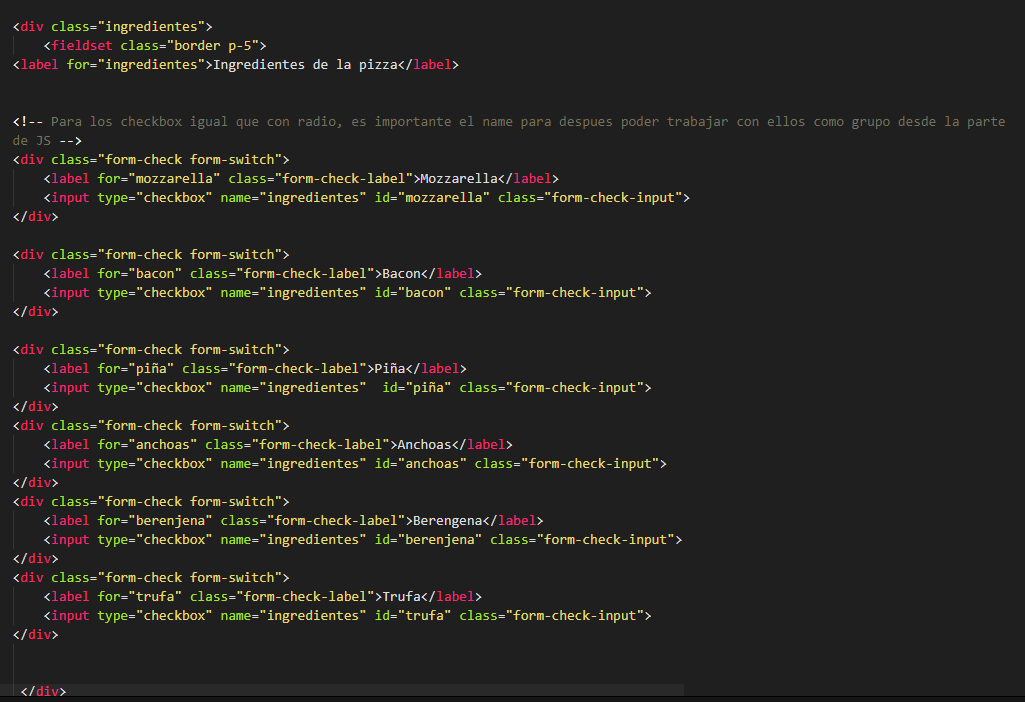
Si fueramos a trabajar con servidor, hay que acordarse también de definir un name.



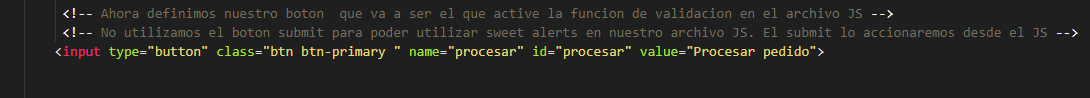
En nuestra segunda parte del formulario, definimos el grupo de input tipo radio. Aquí si debemos de definir un name para llamar a todo el grupo y que formen uno solo. Además, podemos utilizar el name para seleccionarlo desde JS para recorrer los elementos.



Nuestro grupo de ingredientes, formados por checkbox, los agruparemos también bajo un name, porque después nos servirá para contar cuantos hemos añadido y tratarlo como grupo.



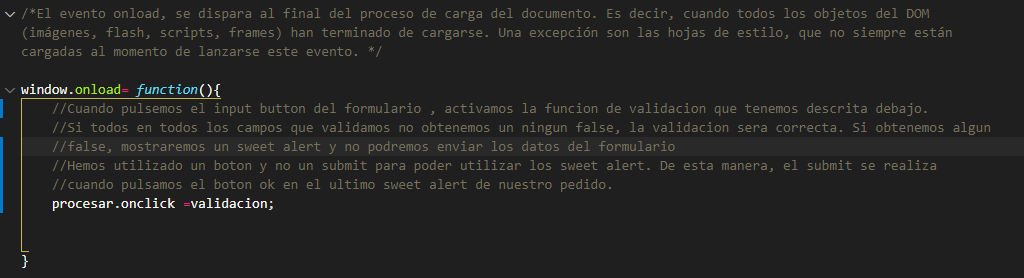
Por ultimo en nuestro formulario tenemos el input de tipo button que será el que desencadena nuestro evento en el archivo JS. Pero esto lo analizamos a continuación en este archivo. Hemos utilizado un tipo button para poder utilizar Sweet Alert en nuestro documento Javascript. Activaremos el submit del formulario al aceptar nuestro pedido con el sweet alert.



De esta manera, ya tenemos construido nuestro formulario mediante html, el problema es que por ahora no tiene ninguna validación y por lo tanto no se ajusta al enunciado. Mediante el archivo JS que definimos a continuación conseguiremos esta funcionalidad.

**ARCHIVO VALIDACION.JS**

Nuestro archivo JS el cual hemos enlazado en el head de nuestro html, está estructurado de la siguiente manera:

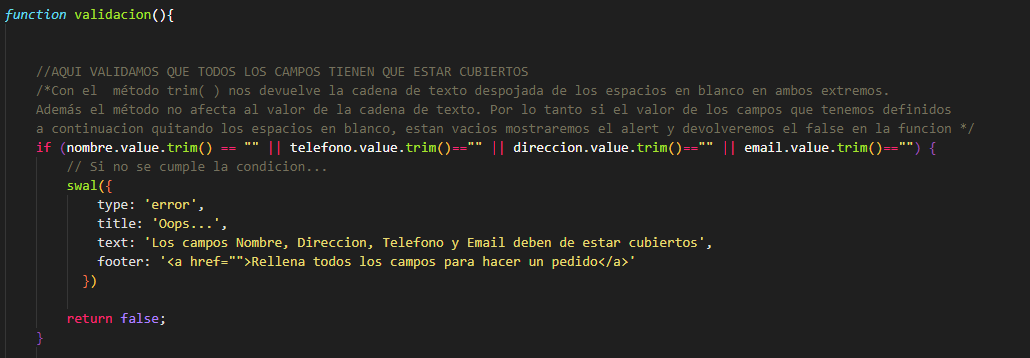


|  |
| --- |
|  |

Primero establecemos mediante el evento onload que cuando todos los objetos del DOM han terminado de cargarse, llamamos a la función que describimos.

Esta función determina que al hacer click en el botón del formulario, se lleva a cabo el análisis de la función validación que describimos abajo en el código. Por ahora sabemos que para que eso se lleve a cabo, esta función debe de devolver un true, si devuelve false, la validación será false y por lo tanto el submit al final de la función no se llevara a cabo.

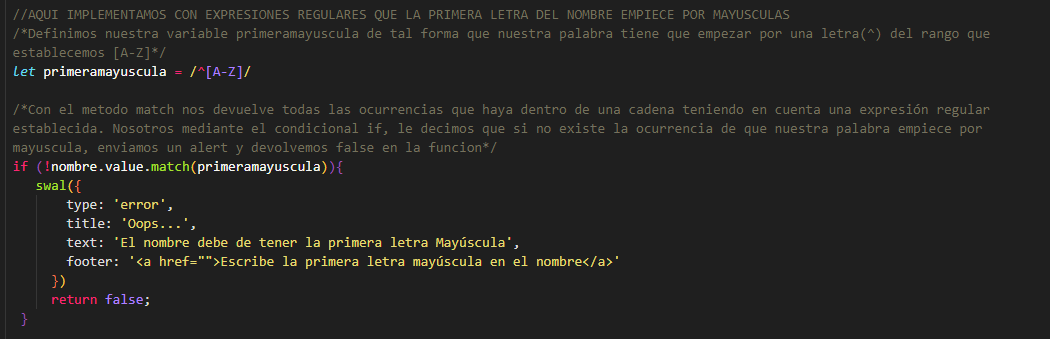
A continuación, analizamos la función validación:



Lo primero que analizamos en nuestra función es que todos los campos deben de estar completos, ninguno debe de encontrarse en blanco.

Para eso, utilizamos nuestra propiedad value con el método trim() el cual nos devuelve la cadena despojada de los espacios. Por lo tanto, si todos los campos están vacíos, devolvemos un false en la función para que no se lleve a cabo el submit, y se muestre una alerta (alert).

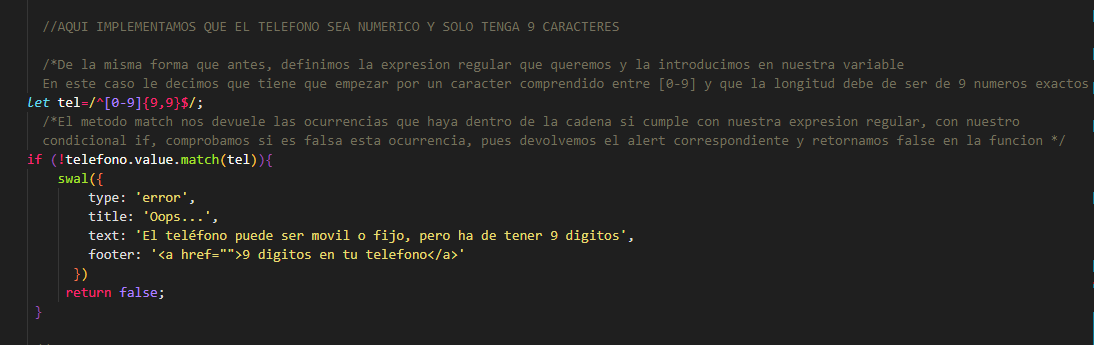
Lo segundo que analizamos es que el nombre empiece por letra mayúscula:



Para ello, utilizamos expresiones regulares y con el método match observamos las ocurrencias y lo analizamos con un condicional if para ver si se producen. Si no obtenemos el resultado que deseamos, mostramos un alert y devolvemos un false para que no se lleve a cabo el submit en el formulario.

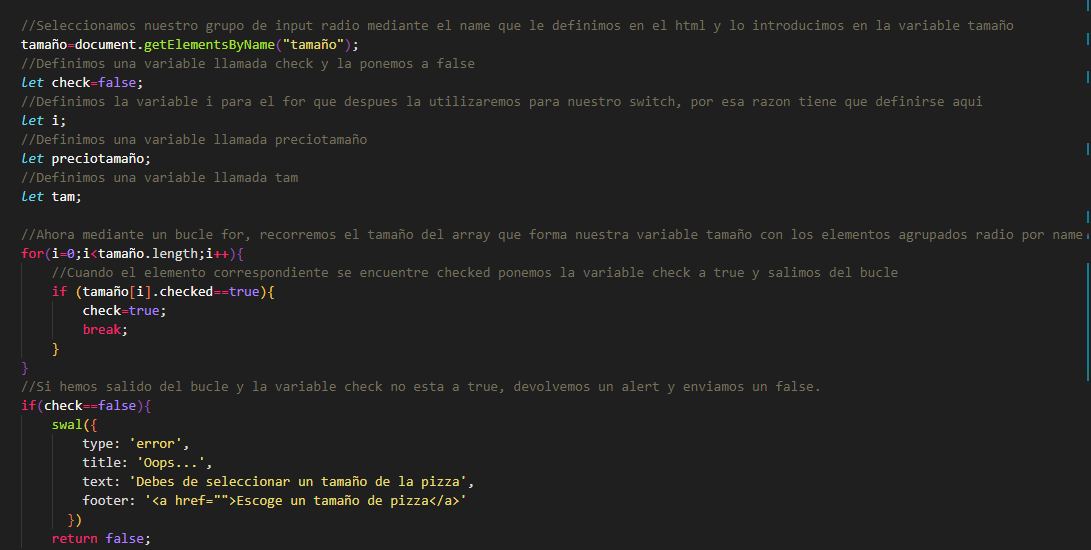
Para el teléfono y email utilizamos expresiones regulares de la misma manera que para el nombre.

Lo difícil en esta cuestión es dar con la expresión regular correcta para conseguir el objetivo:



Cuando ya hemos analizado los inputs tipo texto, el siguiente objetivo es analizar los campos input tipo radio e input tipo checkbox.

Lo primero que nos dice el enunciado es que hay que escoger un tamaño de pizza para que se envíen los datos:

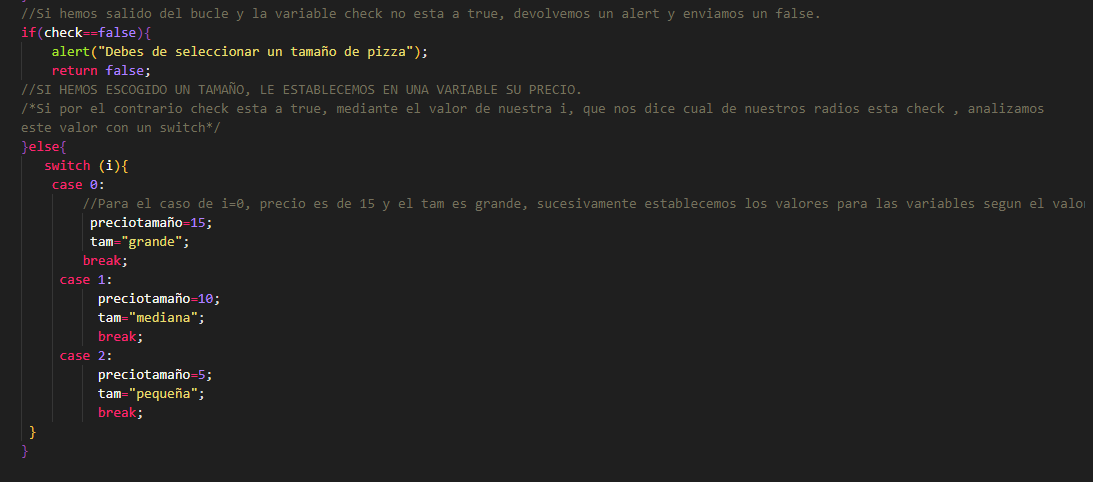


Para ello, definimos una variable llamada tamaño que es el grupo de input radio el cual seleccionamos mediante el name que habíamos definido en el html. Este grupo se va a comportar como un “array” y por lo tanto podemos recorrerlo mediante un bucle for como tal.

También definimos unas variables que después vamos a ir cargando con valores según los resultados.

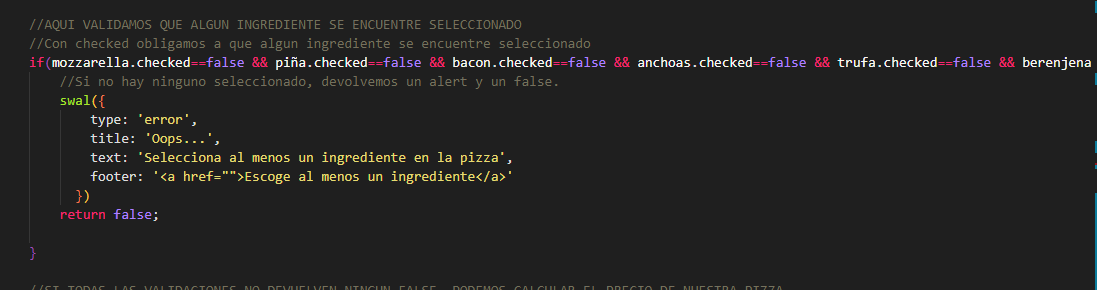
Con el bucle for recorremos nuestro grupo de input radio y con un condicional if, analizamos si alguno de los elementos de ese grupo se encuentra en estado checked. Si esto se cumple, ponemos el check a true y salimos del bucle. Analizamos esa variable check y si es false, mostramos un alert y devolvemos un false a la función.

Si el check es true:



Analizamos mediante un switch que elemento de nuestro grupo input radio se encontraba en checked e introducimos en las variables correspondientes sus valores para después presentar el precio correctamente.

El siguiente paso es validar que al menos un ingrediente se encuentra seleccionado. Este paso es muy sencillo con la propiedad checked que hemos analizado antes y con un condicional if:



Analizamos que, si ninguno de nuestros ingredientes están checked, mostramos un alert y devolvemos un false a la función.

Cuando tenemos todas nuestras validaciones comprobadas y ninguna devuelve un false, es el momento de analizar el precio de nuestra pizza con su tamaño e ingredientes correspondientes:

 /\*Para saber cuantos ingredientes hemos seleccionado, seleccionamos nuestro grupo checkbox con el name que le establecimos

    en el html y lo recorremos con un bucle for \*/

    ingredientes=document.getElementsByName("ingredientes");

*let* j;

*let* precioingredientes=0;

    for(j=0;j<ingredientes.length;j++){

        //Cada vez que un ingrediente se encuentra seleccionado sumamos un ingrediente a precioingredientes

        if(ingredientes[j].checked==true){

            precioingredientes++;

        }

    }

    //Y en el ultimo sweet alert devolvemos el valor total de la pizza desglosando sus apartados

    swal({

        title:"La pizza pedida es: \n"

         + "Tamaño : "+ tam.toUpperCase() + " " + preciotamaño + " euros",

        text: "Has añadido : \n"+

        precioingredientes + " ingredientes: " + precioingredientes + "euros.",

        type: 'info',

        confirmButtonColor: '#3085d6',

        confirmButtonText: 'OK'

      }).then((*result*) *=>* {

            swal(

                {

                    title              : "El precio total de la pizza es: \n" +

                    "TAMAÑO: "+ preciotamaño+ "EUROS\n" +

                    "INGREDIENTES: " + precioingredientes + "EUROS\n"+

                    "TOTAL: " +(preciotamaño+precioingredientes) + " EUROS.",

                    text               : "¿Deseas hacer el pedido?",

                    type               : "success",

                    allowEscapeKey     : false,

                    allowOutsideClick  : false,

                    showCancelButton   : true,

                    confirmButtonColor : "#DD6B55",

                    confirmButtonText  : "Yes",

                    showLoaderOnConfirm: true,

                    closeOnConfirm     : false,

                    //en sweet alert para el submit es necesario el parametro preconfirm, sino no envia formulario, ya que tiene que ser async

                    preConfirm:   *function* (*isConfirm*) {

                        if (isConfirm) {

                            //Aqui es donde hacemos el submit del formulario, cuando ya tenemos todo preparado

                            document.formulario.submit();

                            return true;

                        }else{

                            return false;

                        }

                    }

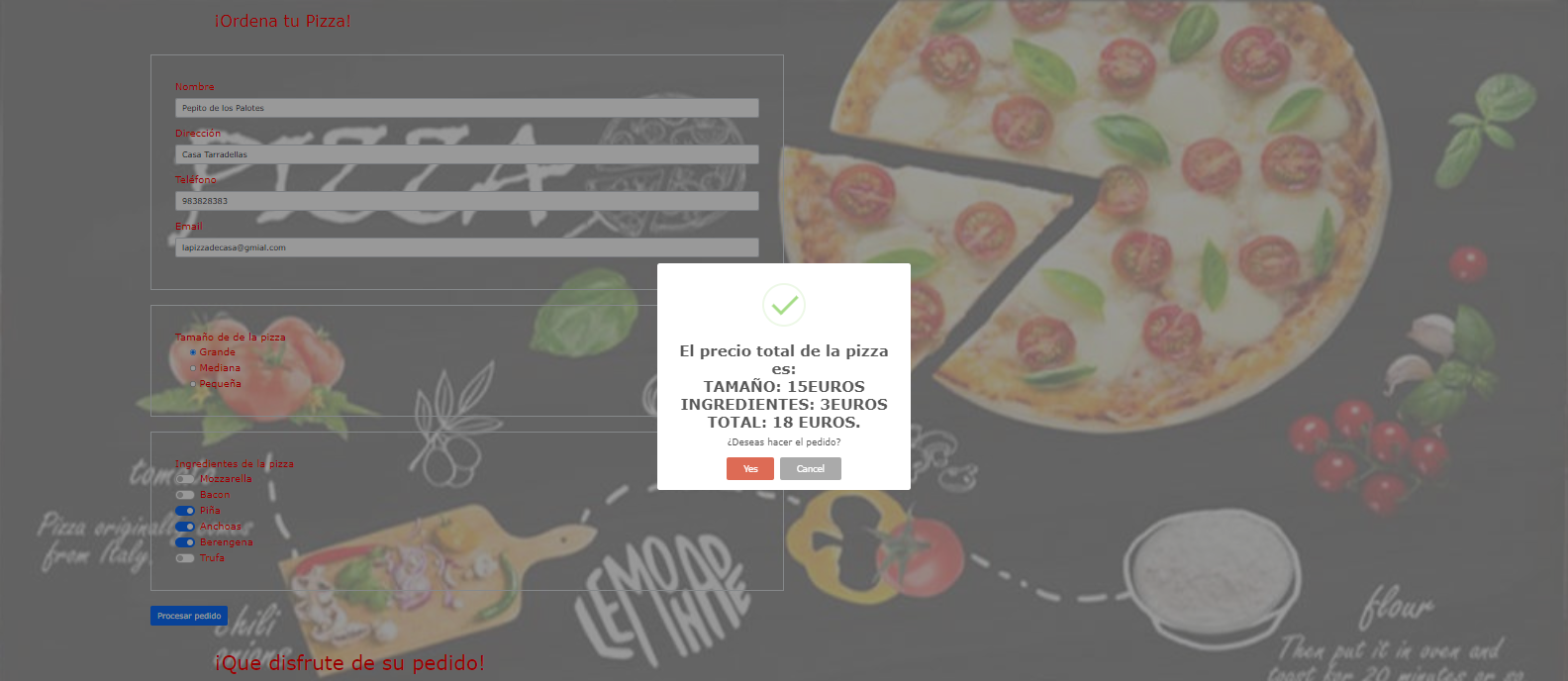
                }

            )

El precio según el tamaño ya lo habíamos guardado en una variable cuando lo analizamos con el bucle for y el switch. Para saber cuántos ingredientes hemos seleccionado, utilizamos un bucle for y según los checked que nos encontremos sumamos una unidad.

Después mediante alert solo nos queda presentar nuestro resultado del precio total.

Esta será nuestra imagen de nuestro formulario con la solución:



Todo el proyecto se encuentra comentado para explicar las decisiones que se toman en cada momento.

Esta memoria es un resumen de cómo funciona el proyecto.