

# JS

AJAX



2º DAW

Alberto Saboya – Anabella Aceto– Cristina Tropotel – David Rodríguez

## Tabla de contenido

Introducción .....	3
Window.onload.....	4
Petición al servidor.....	4
Procesamiento de la petición .....	5
Creamos los inputs de los tamaños de pizza.....	5
Creamos los inputs de los ingredientes .....	5
Validaciones .....	6
Cajas de texto y teléfono .....	6
Tamaño .....	6
Ingredientes .....	7
Mostrar total.....	7
Tamaño .....	7
Ingredientes .....	8
Archivo .Json .....	8
TEST DEL CÓDIGO .....	9
No hemos marcado nada.....	9
No hemos marcado el nombre .....	9
No hemos marcado tamaños.....	9
No hemos marcado ingredientes.....	10
Todo correcto .....	10
Probamos el refrescar .....	10
Conclusión.....	11

## Introducción

El presente trabajo es el resultado de un trabajo tanto individual como grupal, donde la colaboración y el consenso fueron elementos clave para la resolución de las actividades. En esta propuesta utilizamos tecnologías como AJAX, JSON y JavaScript.

La implementación de AJAX nos permitió realizar solicitudes asincrónicas al servidor. El intercambio de datos se llevó a cabo en formato JSON, facilitando la manipulación de la información y su presentación dinámica en el lado del cliente mediante JavaScript.

Las sesiones de grupo no solo sirvieron como ocasiones para compartir progresos, sino también como instancias para abordar desafíos técnicos, deliberar sobre mejoras y coordinar esfuerzos. La colaboración continua y la retroalimentación mutua constituyeron elementos esenciales de nuestra estrategia, posibilitándonos superar obstáculos de manera eficaz.

Cabe mencionar que, otra vez, github ha sido una herramienta valiosa para la gestión de versiones. Su capacidad para crear ramas facilitó la coordinación entre los miembros del equipo, posibilitando el trabajo conjunto y asincrónico.

Por último, mencionamos que el trabajo elegido para la presentación es el de David Rodríguez. Tanto los proyectos individuales como el seleccionado pueden consultarse en el siguiente link de github:

<https://github.com/Xukialb/AE-2.-AJAX.git>

En este caso hemos extraído el código de la actividad anterior, y hemos realizado ciertas modificaciones ya que este era muy adaptable a lo que se solicitaba en la actividad, hemos sido capaces de mantener la función de validación de datos.

Hemos comentado el código de la actividad al detalle de que hace paso a paso, por ello vamos a adjuntar capturas de todo el código completo y al final del documento pondremos a prueba nuestro código.

## Window.onload

```
1
2 //Cargamos todo lo que hay dentro una vez se haya cargado la web.
3 window.onload = function () {
4
5     //creamos los diferentes div y botones que tenemos en el html
6     var bottomform = document.getElementById("bottomform");
7     var contenedor = document.getElementById("contenedor");
8     //div de los dos botones (total - refrescar)
9     var botones = document.getElementById("botones");
10    //botón refrescar
11    var refrescar = document.getElementById("refrescar");
12
13    //Creamos el EventListener para que al hacer click en el botón con refrescar
14    //nos deje el div (bottomform) en blanco y posteriormente procese de nuevo la petición
15    //al json para volver a traer los tamaños de pizzas y sus ingredientes sin tocar nada más
16    refrescar.addEventListener("click", function () {
17        //con las dobles comillas en blanco dejamos limpio el div y posteriormente llamamos
18        //a la función de petición para solicitar de nuevo el json
19        bottomform.innerHTML = "";
20        enviarPeticiónAsíncrona();
21    });
22
23    //Llamamos a la función de petición
24    enviarPeticiónAsíncrona();
25
26    //Hacemos hijos el bottomform y los botones
27    contenedor.appendChild(bottomform);
28    contenedor.appendChild(botones);
29 }
30
```

## Petición al servidor

```
31 //Insertamos la URL donde vamos a tener el recurso .json
32 const URL_DESTINO = "http://localhost:5500/Act1/AE-2.-AJAX/David/"
33 //indicamos el nombre dle recurso
34 const RECURSO = "pizzas.json"
35
36 //Creamos la petición, en nuestro caso es Asíncrona
37 function enviarPeticiónAsíncrona() {
38
39     //Creamos el objeto XMLHttpRequest - Esto es obligatorio en cada petición que hagamos
40     let xmlhttp = new XMLHttpRequest()
41
42     // Solicitamos el estado de la petición con .onreadystatechange
43     xmlhttp.onreadystatechange = function () {
44         //Si el estado que devuelve la petición es 4 comprobamos si su estado es 200
45         if (this.readyState == 4) {
46             //Si el estado de la petición es 200, todo ha salido bien, tendremos nuestro JSON
47             if (this.status == 200) {
48                 //llamamos a la función que va a procesar el .Json, y le pasamos le .Json con
49                 //this.responseText.
50                 procesarRespuesta(this.responseText);
51             } else { // En caso de no responder con status 200 va a generar un error cualquiera
52                 //Si esto se da, le decimos que nos saque un alert con un "ZASCA!"
53                 alert("ZASCA!")
54             }
55         }
56     }
57
58     //Al final tenemos que recuperar la url y el recurso usando GET, tenemos que marcar al final
59     //como true la petición y que sino no será una petición asíncrona
60     xmlhttp.open('GET', URL_DESTINO + RECURSO, true)
61     //En el envío de datos marcamos null porque no vamos a mandar datos al .Json, solo vamos
62     //a sacar datos de este.
63     xmlhttp.send(null)
64 }
```

# Procesamiento de la petición

## Creamos los inputs de los tamaños de pizza

```
64 //En esta función vamos a procesar el texto que nos devuelve el .Json
65 //Indicamos que le tenemos que pasar el jsonDoc que más arriba hemos indicado como (this.responseText);
66 function procesarRespuesta(jsonDoc) {
67     //Metemos lo que nos devuelve la consulta del .Json en una variable llamada "objetoJson"
68     //A la vez parseamos lo que contiene.
69     var objetoJson = JSON.parse(jsonDoc);
70
71     //Dentro del Json tenemos 2 objetos, las Pizzas y los ingredientes, nosotros vamos a meter
72     //El Objeto de las pizzas dentro de una variable, para llegar hasta las pizzas tenemos que
73     //Entrar al objetoJson. e indicar a que queremos acceder, en nuestro caso a las Pizzas
74     // objetoJson.Pizzas
75     var arraypizzas = objetoJson.Pizzas;
76
77     //Ya tenemos las pizzas en una variable, si nos vamos a nuestro Json, vemos que tenemos un
78     //Array con las pizzas y ya dentro de cada una encontramos su precio y tamaño.
79     //Ahora lo que aremos será recorrer este array de 3 lugares para ir creando cada botón
80     for (var i = 0; i < arraypizzas.length; i++) {
81         //Creamos esta variable para meter ahí el nombre de las pizzas del Json
82         var nompizza = arraypizzas[i].nombre;
83
84         //En esta variable metemos le asignamos la etiqueta span para meter en una misma línea
85         //todos tamaños de las pizzas, y entre nombre y nombre metemos un espacio
86         var namepizza = document.createElement('span');
87         namepizza.textContent = " " + nompizza + " ";
88
89         //Lo hacemos como hijo del div al que pertenece
90         bottomform.appendChild(namepizza);
91
92         //Creamos el input de las pizzas y le asignamos sus atributos correspondientes
93         var pizza = document.createElement("input");
94         pizza.setAttribute("name", "tamano");
95         pizza.setAttribute("type", "radio");
96         //Tanto el id como el value los extraemos de nuestro arraypizzas(que sería el objeto Pizzas dentro del Json) y ya
97         //Dentro de este array extraemos el nombre, en este caso "i" corresponde a una posición del array y esto corresponde
98         //a un tamaño de pizza de los 3 que tenemos, lo mismo aplica con el precio
99         pizza.setAttribute("id", arraypizzas[i].nombre)
100         pizza.setAttribute("value", arraypizzas[i].precio);
101         bottomform.appendChild(pizza);
102     }
```

## Creamos los inputs de los ingredientes

```
103
104 //Ahora vamos a crear los ingredientes
105
106 //Creamos esta variable para guardar ahí el array de los ingredientes que tenemos
107 var ingredient = objetoJson.Ingredientes.lista
108
109 //Metemos varios enter para ajustarlo bien
110 var br = document.createElement("br");
111 bottomform.appendChild(br);
112 var br = document.createElement("br");
113 bottomform.appendChild(br);
114
115 //Al igual que con los tamaños de las pizzas repetimos el procedimiento pero con los ingredientes, es exactamente igual
116 for (var i = 0; i < ingredient.length; i++) {
117     var textingredient = document.createElement('span');
118     textingredient.textContent = " " + ingredient[i].Nombre;
119     bottomform.appendChild(textingredient);
120
121     var ingredientes = document.createElement("input");
122     ingredientes.setAttribute("name", "ingrediente");
123     ingredientes.setAttribute("type", "checkbox");
124     ingredientes.setAttribute("id", ingredient[i].Nombre);
125     ingredientes.setAttribute("value", ingredient[i].Precio);
126     bottomform.appendChild(ingredientes);
127 }
128
129 //Una vez hemos creado los tamaños de las pizzas y sus ingredientes
130 //vamos a crear la acción que va a ejecutar el botón del total al ser pulsado
131
132 //vamos a extraer el botón total por su ID
133 var totalboton = document.getElementById("total");
134
135 //Ya podemos manejar este botón, con el onclick le indicamos que cuando se pulse haga x acción que nosotros indiquemos
136 //en este caso va a llamar a la función validarForm.
137 totalboton.onclick = validarForm;
138
```

# Validaciones

## Cajas de texto y teléfono

```
138
139
140 //Tenemos que validar nuestro formulario
141 function validarForm(){
142
143     //Extraemos el ID de cada caja de texto que vamos a comprobar y lo pasamos a una variable
144     var nombre = document.getElementById("nombre");
145     var apellidos = document.getElementById("apellidos");
146     var direccion = document.getElementById("direccion");
147
148     // Validar campos de datos obligatorios mediante este if lo que hacemos es comprobar que ninguna de estas variables
149     //estén en blanco, con .trim lo que hacemos es eliminar los espacios al principio y final
150     if (nombre.value.trim() === "" || apellidos.value.trim() === "" || direccion.value.trim() === "") {
151
152         //En caso de estar alguno de estos 3 campos vacío nos tirará este alert
153         alert("Los campos de datos son obligatorios");
154         return false;
155     }
156
157     //Eliminamos los espacios del número de teléfono para posteriormente validarlo
158     var numeroTelefono = telefono.value.trim();
159
160     // Validamos el teléfono de la misma forma que el texto solo que aquí hemos añadido una doble validación
161     //No solo comprobamos que no esté vacío, sino que también comprobamos que tiene por lo menos 9 dígitos y que sean números
162     if (numeroTelefono === "") {
163         alert("El campo de teléfono es obligatorio");
164         return false;
165     }
166
167     // Validar que el número de teléfono tiene exactamente 9 dígitos
168     if (numeroTelefono.length !== 9 || isNaN(numeroTelefono)) {
169         alert("El número de teléfono debe tener nueve dígitos");
170         return false;
171     }
172 }
```

## Tamaño

```
171 }
172
173 // Ahora vamos a validar si el usuario a seleccionado algún tamaño de pizza
174 var tamano = document.getElementsByName("tamano");
175 //Ponemos la variable seleccionTam en false
176 var seleccionTam = false;
177
178     //Con este bucle vamos a recorrer nuestro array de tamaños de pizza (3 lugares), y comprobaremos
179     //uno a uno si hay alguno seleccionado, si hay alguno seleccionado se cambiará a true la variable
180     //seleccionTam y por tanto romperemos el bucle y nos saldremos de este.
181     for (var i = 0; i < tamano.length; i++) {
182         if (tamano[i].checked) {
183             seleccionTam = true;
184             break;
185         }
186     }
187
188     //Si la selección del tamaño no devuelve TRUE lo que haremos será sacar un alert indicando error
189     if (!seleccionTam) {
190         alert("Debe seleccionar un tamaño de pizza");
191         return false;
192     }
193 }
```

## Ingredientes

```
191 }
192
193 //Al igual que hemos hecho con el tamaño hacemos ocn los ingredientes
194 var ingredientes = document.getElementsByName("ingrediente");
195
196 //Creamos una variable con valor false
197 var seleccionIng = false;
198
199 //Creamos un bucle en el que recorremos el array de ingredientes
200
201 for (var i = 0; i < ingredientes.length; i++) {
202
203     //Con un if vamos comprobando si hay alguno marcado, en cuanto encuentre uno seleccionado nos cambia
204     //el valor de seleccionIng a True y nos sacará del for.
205     if (ingredientes[i].checked) {
206         seleccionIng = true;
207         break;
208     }
209
210     //En caso de que el if anterior no nos de TRUE deberá de sacarnos un mensaje de error
211     if (!seleccionIng) {
212         alert("Debe seleccionar al menos un ingrediente");
213         return false;
214     }
215
216
217 // Si todas las validaciones pasan, llama a la función para mostrar el total
218 mostrarTotal();
219
220 //marcamos return true para que se envíe el formulario
221 return true;
222 }
```

## Mostrar total

### Tamaño

```
222 }
223
224 //Mostraremos el total del coste de Pizza + ingrediente
225 function mostrarTotal() {
226
227     //Creamos una variable donde tendremos almacenado el precio del tamaño de la pizza y la iniciamos a 0
228     var precioPizza = 0;
229
230     // almacenamos todos los botones con atributo name "tamaño" en una variable llamada tamanio
231     var tamanio = document.getElementsByName("tamaño");
232
233     //Con este bucle for lo que hacemos es recorrer el array de tamanio
234     for (let i = 0; i < tamanio.length; i++) {
235         //Vamos en cada lugar del array tenemos un tamaño de pizza
236         //Vamos a comprobar si cada tamaño ha sido seleccionado, cuando haya uno seleccionado
237         //asignamos ese valor a la variable precioPizza
238         if (tamanio[i].checked) {
239             //Aquí lo que estamos haciendo es averiguar cual ha seleccionado el usuario, estamos dentro de un if
240             //que se ejecuta en bucle hasta encontrar un botón seleccionado, cuando lo encuentre, este va a ejecutar varios if
241             //En los que vamos a comprobar si el valor de ese botón seleccionado es 5,10,15, dependiendo del valor sabemos
242             //que nuestra pizza es pequeña, mediana o grande, de ser otro caso podríamos identificarlo no por su value sino por su id
243             //ya que este sería el nombre de cada pizza o por un número de referencia, en este caso era fácil sacarlo por el value
244             if (tamanio[i].value == 5) {
245                 precioPizza = 5;
246             }
247             else if (tamanio[i].value == 10) {
248                 precioPizza = 10;
249             }
250             else if (tamanio[i].value == 15) {
251                 precioPizza = 15;
252             }
253         }
254     }
255 }
```

## Ingredientes

```
254 }
255
256
257 // Obtener los ingredientes seleccionados
258
259 //Metemos todo los radio button de ingredientes en una variable llamada ingredientes
260 var ingredientes = document.getElementsByName("ingrediente");
261
262 //Creamos la variable para almacenar el precio de los ingredientes y la iniciamos a 0;
263 var ingredientesPrecio = 0;
264
265 //Es similar a las pizzas, vamos recorriendo el array de los ingredientes que tenemos
266 for (let i = 0; i < ingredientes.length; i++) {
267     //cuando encuentra un ingrediente seleccionado va a proceder a parsear su valor (para que lo sume y no lo concatene con el de la pizza)
268
269     if (ingredientes[i].checked) {
270         //Aquí estamos parseando el valor del ingrediente seleccionado
271         var vingredient=parseFloat(ingredientes[i].value);
272         //Indicamos que sume el ingrediente a nuestra variable que almacena el coste de los ingredientes
273         ingredientesPrecio += vingredient;
274     }
275 }
276
277 //Finalmente sumamos el precio del tamaño y de los ingredientes y lo sacamos por un alert
278 let precioTotal = precioPizza + ingredientesPrecio;
279 alert("El importe es: " + precioTotal + "€");
280
281 }
282
283 }
```

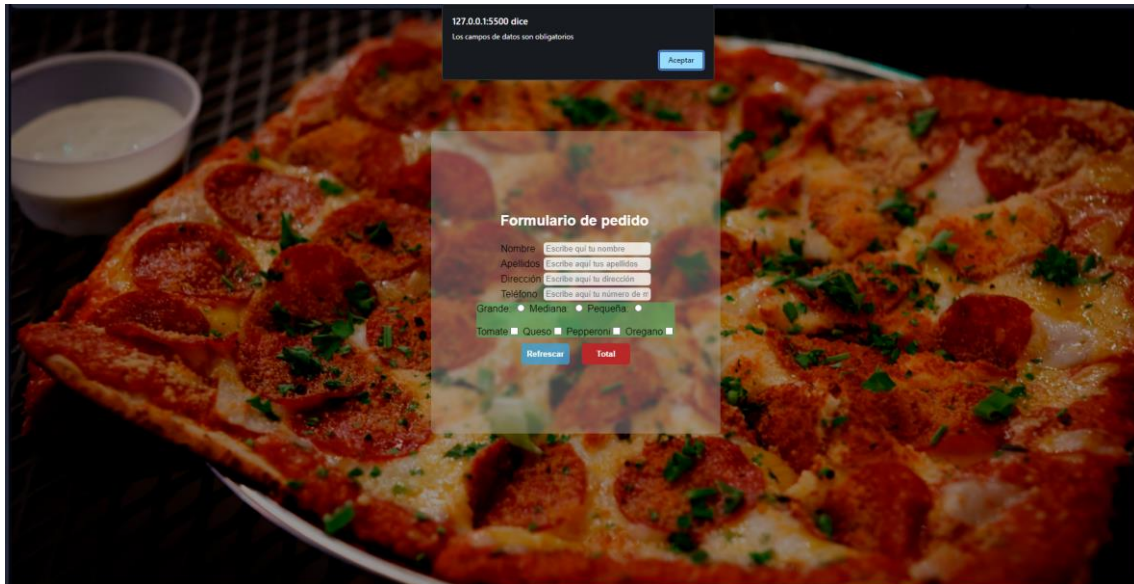
## Archivo .Json

```
unknown, ayer | 1 author (unknown)
1  [
2    "Pizzas": [
3      {
4        "nombre": "Grande",
5        "precio": 15
6      },
7      {
8        "nombre": "Mediana",
9        "precio": 10
10     },
11     {
12       "nombre": "Pequeña",
13       "precio": 5
14     }
15   ],
16   "Ingredientes": {
17     "lista": [
18       {"Nombre": "Tomate",
19       "Precio": 2},
20       {"Nombre": "Queso",
21       "Precio": 4},
22       {"Nombre": "Pepperoni",
23       "Precio": 6},
24       {"Nombre": "Oregano",
25       "Precio": 15}]
26   }
27 ]
28
29 ]
```

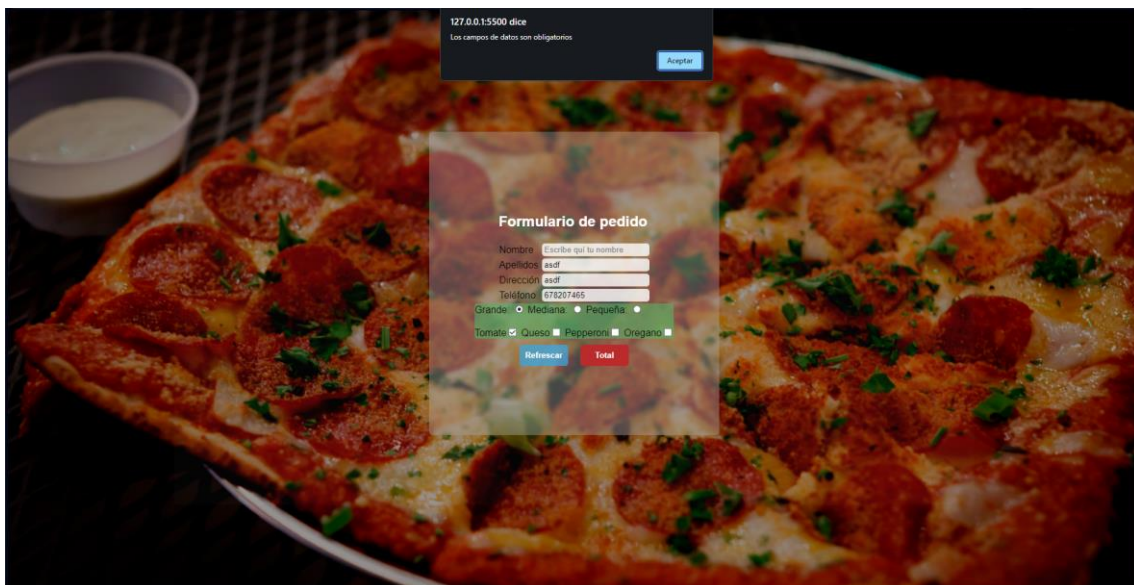


# TEST DEL CÓDIGO

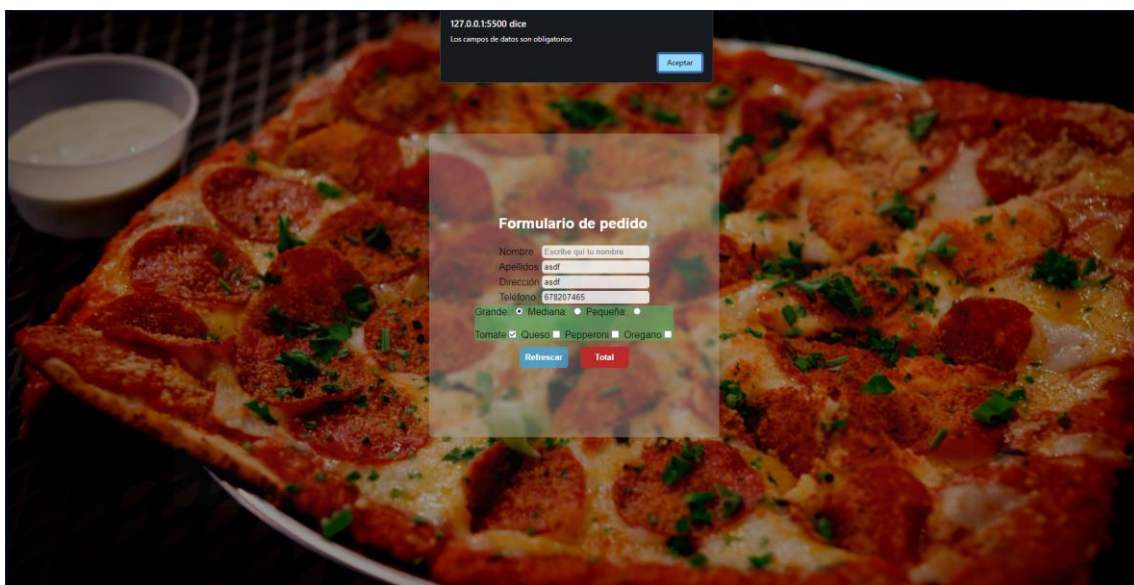
No hemos marcado nada



No hemos marcado el nombre

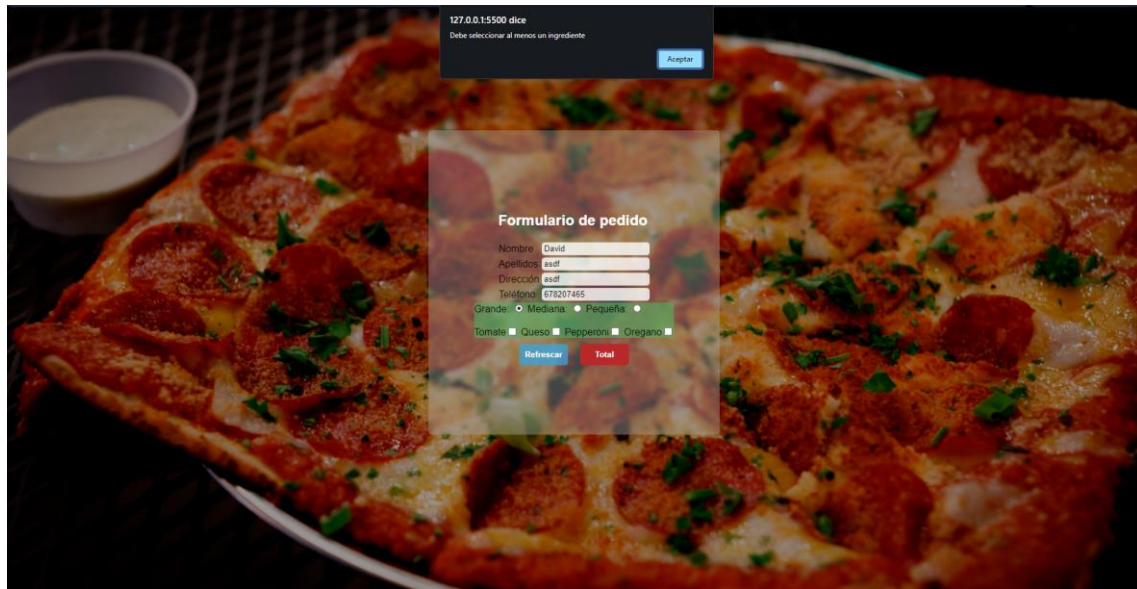


No hemos marcado tamaños

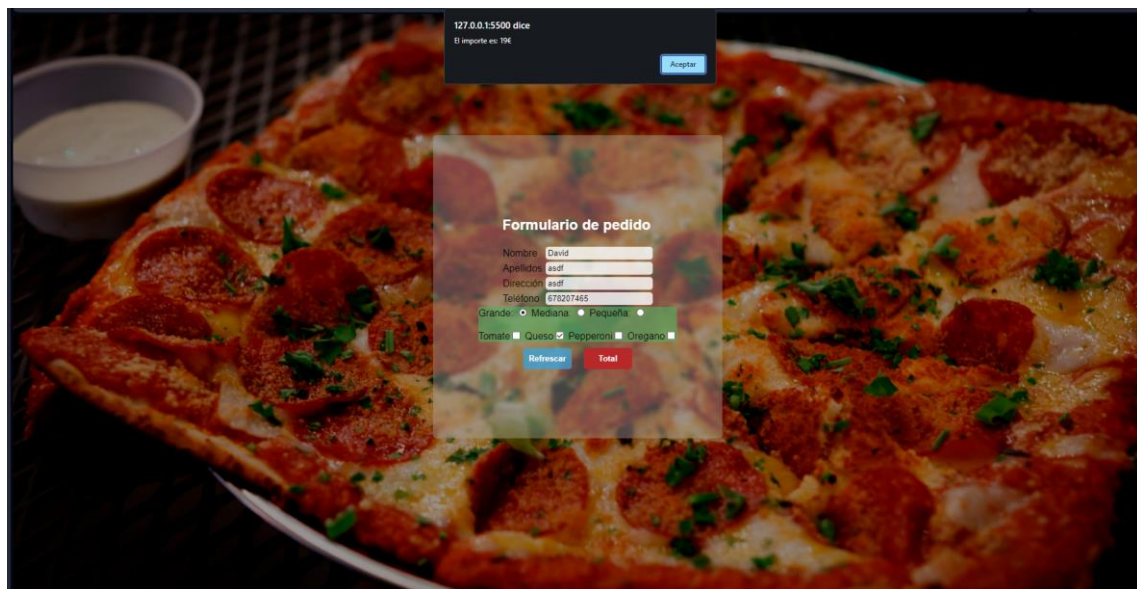




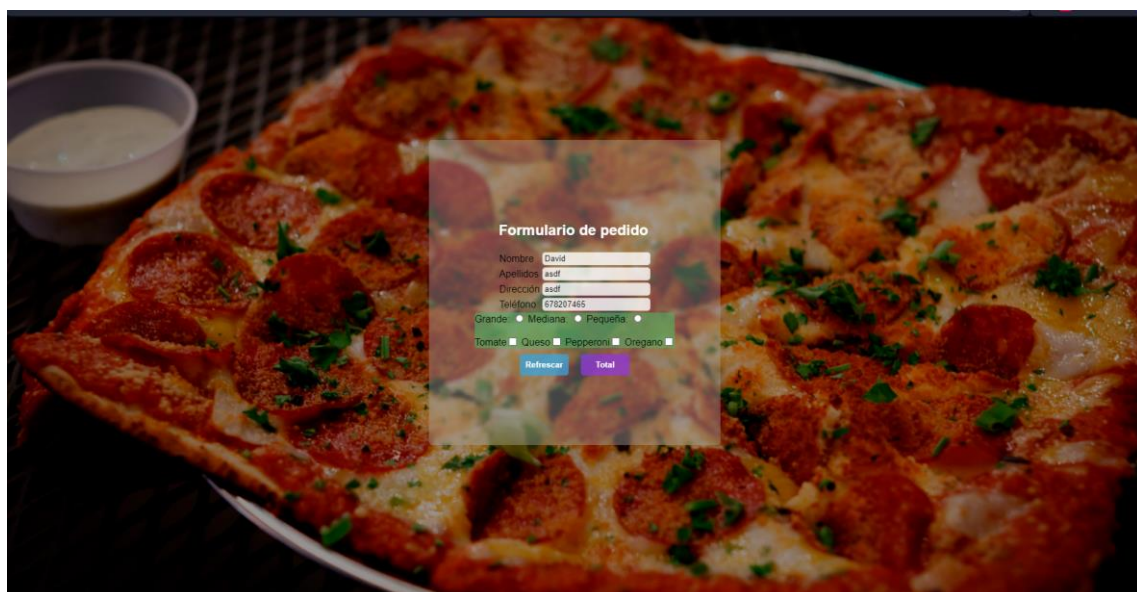
## No hemos marcado ingredientes



## Todo correcto



## Probamos el refrescar



## Conclusión

En conclusión, la implementación de AJAX y funciones asíncronas en JavaScript presentó desafíos significativos durante nuestro proyecto, lo que requirió de la colaboración de cada miembro del equipo. Enfrentamos obstáculos relacionados con la gestión de tiempos de respuesta del servidor y el manejo de datos asíncronos.

Sin embargo, mediante reuniones regulares de grupo y una comunicación fluida pudimos hacer frente a estas dificultades de manera efectiva. La resolución de problemas se basó en un enfoque colaborativo, donde cada miembro contribuyó con ideas y soluciones. GitHub se convirtió en una herramienta esencial para el seguimiento y la resolución de problemas.

A pesar de las complejidades propias de estos lenguajes, el equipo no solo superó los desafíos de AJAX y funciones asíncronas, sino que también fortaleció nuestra comprensión individual y colectiva de estas tecnologías. Esta experiencia no solo representa una revisión integral de nuestras habilidades en JavaScript, sino también una asimilación profunda de los aspectos cruciales de estas tecnologías en el desarrollo de aplicaciones web.