

## DOCUMENTACIÓN AE-2. AJAX (DWECE)

### HTML

Para el HTML nos hemos basado en el formulario de la actividad 1 realizada este trimestre. Hemos adaptado el formato del body utilizando divs para posteriormente poder hacer un CSS más organizado.

A parte de esto, los primeros cuatro inputs que son de tipo texto los hemos introducido en el body manualmente. En el `<div class="options">` hemos creado dos `<div>` más, uno para tamaños y otro para ingredientes. Estos están vacíos ya que serán posteriormente rellenados con una llamada AJAX al archivo `pizzas.json`.

### JAVASCRIPT

Cuando abrimos el archivo JS lo primero que vemos es el "módulo" de carga.

Primero hemos creado el evento `window.onload = function()`, dentro de él está la función `enviarPeticiónAsíncrona()` y así cuando carga la página es lo primero que ejecuta (función explicada posteriormente en la documentación).

Para el botón de refrescar hemos creado un evento `refresh.onclick = () => { location.reload();}` que recarga la página al clicar el botón.

Finalmente, en este módulo encontramos una EventListener que cuando detecta el submit del formulario evita ejecutar la acción con el `e.preventDefault()` y lanza dos funciones, `validación` y `obtenerTotal`.

En el segundo "módulo" encontramos la validación del formulario. En esta hemos creado una función que se encarga de comprobar que cada campo esté rellenado.

Para las de tipo texto lo hace mediante el `value.trim() == ""` para detectar si está vacío. Para el tamaño le metemos la información del `name="tamaño"` en una variable, posteriormente es recorrida con un for para ver si ese campo está seleccionado o no. Si lo está la variable creada "seleccionado" es true. Si no lo está (comprobándolo con un if) salta una alerta con un mensaje de error.

Para los ingredientes hace exactamente el mismo proceso.

En el tercer "módulo" hacemos la llamada a través de AJAX a la base de datos (json). Creamos una constante con la URL\_DESTINO a la cual le metemos la dirección localhost y otra constante llamada RECURSO con el archivo json.

Creamos la función `enviarPeticiónAsíncrona()`, dentro de ésta encontramos:

Primero creamos un objeto XMLHttpRequest, después ejecutamos la función open y send.

#### ESTADOS DEL READystate:

- open = 1
- send = 2
- interactive = 3
- completed = 4

Llamamos al evento `xmlHttpRequest.onreadystatechange = function()` y dentro de él creamos dos if's, uno nos detecta si el readyState es 4, por lo tanto completado y el otro nos detecta si el estado es 200 que asegura que es correcto, si los dos son true ejecutamos las funciones `procesarTamaño()` y `procesarIngredientes()` a las que se les pasa por parámetro el json parseado para poder acceder a éste.

Creamos las funciones `procesarTamaño()` y `procesarIngredientes()` las cuales se les pasa por parámetro el (`jsonDoc`) ya que con estas funciones a través de DOM creamos los elementos necesarios como los label, inputs o textos y los atributos de éstos.

Alguno de los valores como los tamaños y los ingredientes son traídos del `jsonDoc`.

El cuarto y último "módulo" nos calcula el precio de la pizza.

Volvemos a crear una función `obtenerTotal()` para hacer una llamada a AJAX y ejecutar la función `calcularPrecio()` volviéndole a pasar el json parseado por parámetro.

En la función `calcularPrecio()` la cual tiene por parámetro el `jsonDoc` creamos dos variables para los precios de tamaño y ingredientes.

A través de dos for recorreremos dentro del json los tamaños e ingredientes, comprobando si estos han sido seleccionados desde el formulario (para comprobar si han sido seleccionados volvemos a traer la variable tamaño y ingredientes seleccionadas en la validación) y le decimos que si ha sido seleccionada, la busque en el json y nos traiga el precio.

Al final le asociamos a través de dos atributos id un texto con `innerHTML` para el `estadoPedido` y un texto más la suma de dos variables para el `precio`.


## JSON

En el archivo JSON básicamente hemos creado un primer objeto llamado PIZZAS, dentro de éste dos objetos más llamados INGREDIENTES y TAMAÑOS, y dentro de estos un array con objetos que tienen las propiedades *nombre* y *precio*.

## CSS


Para esta actividad hemos querido darle estilo y hemos implementado un archivo CSS en el cual le hemos cambiado la fuente a toda la página y accediendo al HTML mediante las etiquetas, id's y class hemos ido dándole formato a todos los elementos.

## RESULTADO DE LA APP (IMG)


Página principal:  Image text


Validación de los campos:

*Validación de los campos texto*

 Image text

*Validación del campo tamaño*

 Image text

*Validación de los ingredientes* Image text

Resultado obtenido al hacer el submit:

 Image text

Actividad realizada por:

- Adrián Fernández
- Lierni Beltrán de Guevara
- Isaac Calderón
- Marcos Sandá

Esta aplicación ha sido realizada con el IDE VSCode y comprobada en tiempo real a través del plugin Live Server. Aunque también se ha comprobado que pueda accederse a ella a través del archivo HTML. La actividad se ha ido realizando a lo largo de jornadas los 4 integrantes del equipo juntos.

**GitHub:**

<https://github.com/addryx/Desarrollo-web-entorno-cliente-group/tree/main/AE-2-AJAX>