

DESARROLLO WEB EN ENTORNO CLIENTE

PRÁCTICA 1

GRADO SUPERIOR DE DESARROLLO DE APLICACIONES WEB

UNIR FP - CURSO 2023-2024

Grupo 15

Alfredo Jose Hernandez- Rovati Bisso, Daniel Gómez Pérez,

David Ceniceros Aransay, y Diego García Rodríguez.

1. Organización de la tarea	4
2 .Requerimiento 1	5
3 .Requerimiento 2	7

1. ORGANIZACIÓN DE LA TAREA

La tarea se puede consultar en github en el siguiente enlace:
<https://github.com/alfredd/AE1-DWEC/tree/main>

El trabajo se ha desarrollado a través de github para fomentar la colaboración y la interacción entre los miembros del equipo y esto nos ha permitido trabajar de una forma ágil y organizada.

En el repositorio podrás encontrar 2 carpetas, una para cada requerimiento de la tarea con sus archivos correspondientes. Además se podrá encontrar este informe en formato PDF para facilitar la revisión de la tarea y su correcto funcionamiento.

2 .REQUERIMIENTO 1

La primera tarea cuenta con el siguiente enunciado:

Se quiere generar una página web completamente por JavaScript, modificando el árbol DOM. La página web solo tendrá un contenedor (DIV) y sobre dicho contenedor y por medio de JavaScript se añadirán elementos a dicho contenedor. La temática de la web se deja libre al alumno. No es necesario usar CSS. Se valorará usar funciones propias de modificación del árbol DOM y no usar innerHTML.

- *Se añadirá al menos los siguientes elementos HTML a un formulario*
- *Campos de texto donde poner el DNI, el nombre y los apellidos, la dirección y el teléfono.*
- *Dos radio button, con 4 opciones a elegir*
- *Cinco checkbox diferentes*
- *2 imágenes*
- *Un campo select*
- *Un text area*

Para ello, se ha trabajado con JavaScript y se ha evitado el uso de innerHTML que como se puede comprobar en el código no se ha utilizado en ninguna ocasión.

Para resolver las cuestiones planteadas se han utilizado las distintas estrategias que se han visto en los recursos de la asignatura: Como se puede ver en el archivo JavaScript Requerimientos1.js, hay tanto tareas realizadas con el código escrito línea a línea de manera organizada como utilizando funciones que facilitan el trabajo.

Para la generación de los campos de texto se ha utilizado una estructura común de anidamiento similar a la vista en la asignatura en la que ir creando de manera ordenada los campos, sus atributos, el texto que contiene y posteriormente se han agregado.

Para los radio buttons, hemos optado por el uso de la función listarCompanias(). Tras generar un array de operadores de telefonía con dos atributos: nombre y color, hemos invocado esa función en dos bucle for donde se encontraban las estructuras html del tipo de input "radio".

Como se ve esta forma es más ágil y permite escalar de una forma más sencilla si aumenta el volumen de campos a generar.

Para los checkbox se ha utilizado la misma estrategia con una función listarDispositivos() que devuelve un array de dispositivos tecnológicos con dos atributos: nombre y tamaño. Solo hemos usado el nombre y lo hemos transformado en el código html para los checkbox.

Finalmente para las imágenes el select y el textarea hemos vuelto al modelo original de escribir el código de forma ordenada y posteriormente vincular los nodos debido al bajo volumen de oposiciones que teníamos que listar.

El HTML original, como se puede ver en el archivo Requerimiento1.html, solo contiene un div vacío sobre el cual el JS pinta el código, para verificar su funcionamiento adjuntamos una captura.

FORMULARIO DE ALTA

Textfields

DNI
Nombre
Apellidos
Dirección
Teléfono

Radius buttons

Elige tu compañía telefónica

☐ Movistar
☐ Vodafone
☐ Orange
☐ Yingo

Elige tu color favorito

☐ Azul
☐ Rojo
☐ Naranja
☐ Morado

Checkbox buttons

Selecciona los dispositivos que poses

☐ Ordenador
☐ Portátil
☐ Tablet
☐ Smartphone
☐ Smartwatch

Imágenes en formulario

Selecciona tu guerrero tecnológico favorito



Tipo Select

¿Qué te va pareciendo hasta ahora la tarea, Félix?

Textarea

Escribe lo que quieras por aquí

3 .REQUERIMIENTO 2

La segunda tarea cuenta con el siguiente enunciado:

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.

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€

En esta tarea se ha realizado el trabajo en JS en el documento main.js para facilitar su lectura y análisis. El código HTML se ha incluido en el archivo Requerimiento2B.html

El código html como se puede ver, muestra de forma ordenada los campos solicitados, primero los campos de texto, a continuación los radio buttons y finalmente los checkbox. En la siguiente imagen se puede ver cómo queda el HTML antes de comenzar a validar.

← → ↻ 🏠 ⓘ 127.0.0.1:5501/Requerimiento2/Requerimiento2B.html ☆ 🛡️ 📄 🖨️ 🌐 ⋮

Pizzería DAW

Datos de contacto:

Nombre y apellidos:

Dirección:

Teléfono de contacto:

Email:

Tamaño de la pizza:

Pequeña ☐ Mediana ☐ Grande ☐

Ingredientes a añadir:

Jamón ☐ Bacon ☐ Champiñón ☐ Piña ☒

Finalizar pedido

Una vez el formulario está generado, el archivo main.js ejecuta la validación. Para ello primero tenemos la función que evita que salte como erróneo un formulario sin rellenar, esto lo hace la función procesarPedido a través de la "preventDefault".

A continuación, la función validacion() va validando los distintos campos. Para los campos de texto se ha utilizado la estrategia de validar si tras usar trim el campo de texto quedaba vacío, de ser así, se lanza un alert avisando del error y no se ejecuta la orden. Se adjuntan capturas, en la primera con nombre vacío y en la segunda con dirección vacía.

Pizzería DAW

Datos de contacto:

Nombre y apellidos:

Dirección:

Teléfono de contacto:

Email:

Tamaño de la pizza:

Pequeña ☐ Mediana ☐ Grande ☐

Ingredientes a añadir:

Jamón ☐ Bacon ☐ Champiñón ☐ Pata ☒

Finalizar pedido

127.0.0.1:5501 dice

Refleja la dirección

Pizzería DAW

Datos de contacto:

Nombre y apellidos:

Dirección:

Teléfono de contacto:

Email:

Tamaño de la pizza:

Pequeña ☐ Mediana ☐ Grande ☐

Ingredientes a añadir:

Jamón ☐ Bacon ☐ Champiñón ☐ Pata ☒

Finalizar pedido

127.0.0.1:5501 dice

Refleja la dirección

A continuación se valida que se elija un tamaño y se empieza a calcular el precio de la pizza. Para ello se ha utilizado un bucle for que en el caso de que encuentre una casilla marcada, añade a través de un switch su coste asociado a la variable total donde se calcula el precio. Si no encuentra ninguno marcado, lo indica vía alert y corta el proceso de validación.

Pizzería DAW

Datos de contacto:

Nombre y apellidos:
asdf

Dirección:
666 Rue de Bouvines Rd., 3

Teléfono de contacto:
+3666666666

Email:
asdf@a.com

Tamaño de la pizza:

Pequeña ☐ Mediana ☐ Grande ☐

Ingredientes a añadir:

Jamón ☐ Bacon ☐ Champiñón ☐ Pila ☐

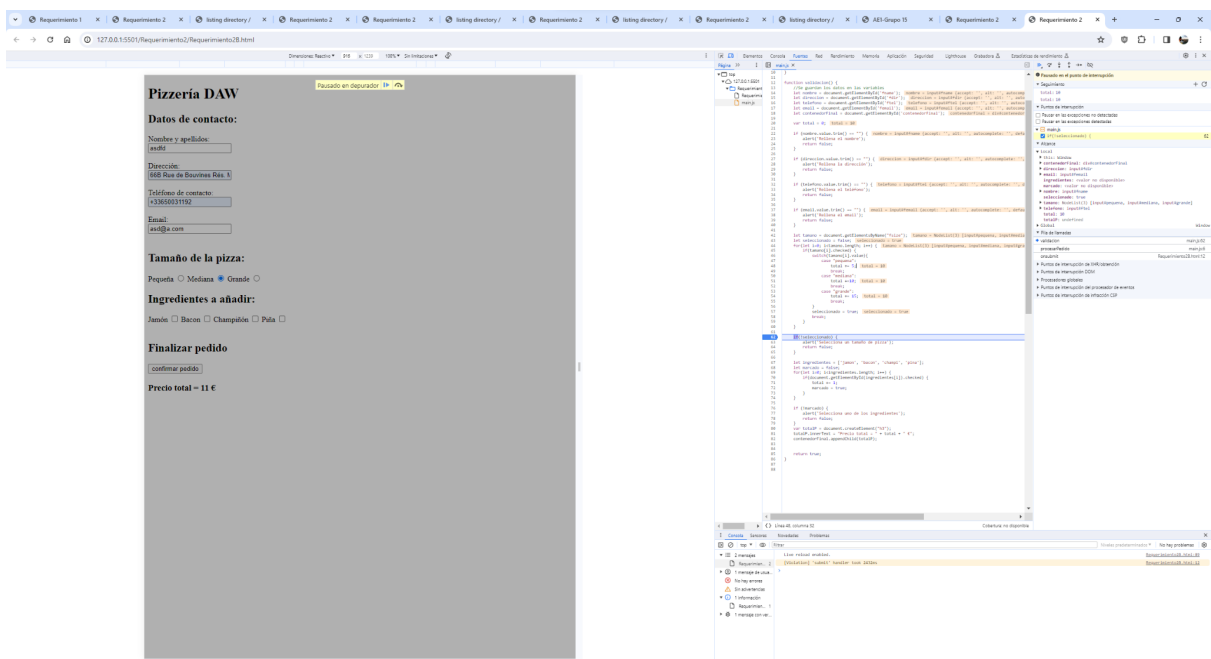
Finalizar pedido

confirmar pedido

127.0.0.1:5501 dice
Selecciona un tamaño de pizza

Aceptar

Aquí podemos ver mediante un punto de interrupción cómo asigna el valor a la variable total si encuentra el valor marcado al validar.



Por último se valida que al menos un ingrediente ha sido seleccionado y se suma el coste del total de los elegidos para mostrarlo en un div situado en la parte inferior. La estructura del código es similar al caso anterior, en un bucle validar si los valores están checkados, en caso de que ninguno lo esté y en caso de que haya alguno sumar el precio a la variable total.

A continuación dos capturas, del alert o del precio ya indicado correctamente.

Requisitante 1

Requisitante 2

listing directory

Requisito 2

Requisitante 2

listing directory

Requisitante 2

listing directory

AEI-Grupo 15

Requisitante 2

Requisitante 2

127.0.0.1:5501/Requisitante2/Requisitante228.html

☆

127.0.0.1:5501 dice

Repetir el nombre

Aceptar

Datos de contacto:

Nombre y apellidos:

Dirección:

Teléfono de contacto:

Email:

Tamaño de la pizza:

Pequeña ☐ Mediana ☒ Grande ☐

Ingredientes a añadir:

Jamón ☐ Bacon ☐ Champiñón ☐ Pera ☐

Finalizar pedido

confirmar pedido

Y aquí la validación completa del proceso y el aviso del precio total.

A screenshot of a web browser displaying the contact information form for Pizzeria DAW. The browser's address bar shows the URL "127.0.0.1:5501/Requisimiento2/Requisimiento28.html". The page has a light blue header with the text "Pizzeria DAW" in a bold, dark font. Below the header, the form is titled "Datos de contacto:" in a bold, dark font. The form contains several input fields: "Nombre y apellidos:" with the value "David Centeno", "Dirección:" with the value "608 Rue de Bonnettes Rue 1", "Teléfono de contacto:" with the value "+33605031192", and "Email:" with the value "asd@a.com". Below the form, there is a section titled "Tamaño de la pizza:" with three radio buttons: "Pequeña", "Mediana", and "Grande". The "Mediana" button is selected. Below this, there is a section titled "Ingredientes a añadir:" with three checkboxes: "Jamón", "Bacon", and "Champignons". The "Bacon" checkbox is checked. Below the checkboxes, there is a section titled "Finalizar pedido" with a button labeled "confirmar pedido". At the bottom of the page, the text "Precio total = 12 €" is displayed.