



Ikaskuntza Birtual eta Digitalizatuen LHII
CIFP de Aprendizajes Virtuales y Digitalizados

UD02 Tarea de Evaluación: Gestión de Reservas de Vehículos

Angela Alonso Moro
DAW DWEC

Curso: 2024/25

EUSKO JAURLARITZA



GOBIERNO VASCO

HEZKUNTZA SAILA
Lanbide Heziketako Sailburuordetza

DEPARTAMENTO DE EDUCACIÓN
Viceconsejería de Formación Profesional

Contenido

1. INTRODUCCIÓN	1
2. DIFERENTES APARTADOS DE LA TAREA	1
3. AUTOEVALUACIÓN	5
4. BIBLIOGRAFÍA	6

1. INTRODUCCIÓN

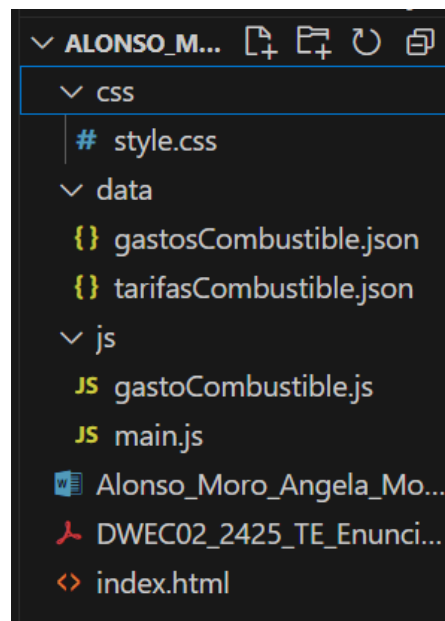
En esta tarea de evaluación he desarrollado una aplicación web para calcular los costos de combustible de una empresa de reparto. La aplicación permitirá introducir los gastos de combustible de diferentes tipos de vehículos (furgonetas, motos y camiones) y visualizar el total gastado en combustible ese año.

La interfaz se divide en dos partes: una para introducir gastos (tipo de vehículo, fecha y kilómetros), y otra para visualizar los últimos gastos y el gasto total de combustible por año. La aplicación solo procesará gastos entre 2010 y 2020.

2. DIFERENTES APARTADOS DE LA TAREA

Ejercicio 1: Estructura de directorios

He seguido la estructura del MVC.



Ejercicio 2: Interfaz de visualización del gasto total

- a) Modifica el calendario para que solo puedan añadirse fechas entre 2010 y 2020

Lo he hecho utilizando HTML

```
<!-- Restringir el calendario a las fechas entre 2010 y 2020 -->  
<input type="date" id="date" name="date" required min="2010-01-01" max="2020-12-31">
```

- b) Incluir en las variables tarifasJSONpath y gastosJSONpath la ruta de los ficheros de datos



```
let tarifasJSONpath = '../data/tarifasCombustible.json';
let gastosJSONpath = '../data/gastosCombustible.json';
```

c) Calcula los gastos entre 2010 y 2020 usando la función calcularGastoTotal().

He iterado sobre cada gasto almacenado en gastosJSON obteniendo el año de la fecha del gasto y sumando el precioViaje al total correspondiente en el array aniosArray.

```
function calcularGastoTotal() {
  // array asociativo con clave=año y valor=gasto total
  let aniosArray = {
    2010: 0,
    2011: 0,
    2012: 0,
    2013: 0,
    2014: 0,
    2015: 0,
    2016: 0,
    2017: 0,
    2018: 0,
    2019: 0,
    2020: 0
  };

  // Sumar gastos existentes
  gastosJSON.forEach(gasto => {
    const fecha = new Date(gasto.date);
    const year = fecha.getFullYear();
    if (aniosArray.hasOwnProperty(year)) {
      aniosArray[year] += gasto.precioViaje;
    }
  });
}
```

d) Muestra el importe del gasto total para cada año en el apartado “Gastos Totales:”

He introducido el resultado dentro del html a través de la clave gastoaño (ejemplo del html).



```
// Actualizar el HTML con los gastos totales
for (let anno in aniosArray) {
    document.getElementById(`gasto${anno}`).textContent = aniosArray[anno].toFixed(2);
}
```

Introducir Datos de Combustible

Tipo de Vehículo:

Furgoneta

Fecha del Gasto:

dd/mm/aaaa

Kilómetros Recorridos:

Guardar Gasto

Gastos recientes:

Gastos Totales:

- El gasto para el año 2020 es 71.96
- El gasto para el año 2019 es 100.95
- El gasto para el año 2018 es 67.85
- El gasto para el año 2017 es 123.10
- El gasto para el año 2016 es 79.82
- El gasto para el año 2015 es 91.08
- El gasto para el año 2014 es 117.50
- El gasto para el año 2013 es 77.40
- El gasto para el año 2012 es 122.55
- El gasto para el año 2011 es 76.70
- El gasto para el año 2010 es 116.18

Ejercicio 3: Interfaz de introducción de datos

- a) Almacena el gasto en un objeto de tipo GastoCombustible



```

< index.html JS main.js JS gastoCombustible.js X {} gastosCombustible.js
js > JS gastoCombustible.js > GastoCombustible > convertToJSON
1
2 //Creación de la clase
3 class GastoCombustible {
4     constructor(vehicleType, date, kilometers, precioViaje) {
5         //Creación de los atributos
6         this.vehicleType = vehicleType;
7         this.date = date;
8         this.kilometers = kilometers;
9         this.precioViaje = precioViaje;
10    }
11
12    // Método para convertir los atributos a formato JSON
13    convertToJSON() {
14        return JSON.stringify({
15            vehicleType: this.vehicleType,
16            date: this.date,
17            kilometers: this.kilometers,
18            precioViaje: this.precioViaje
19        });
20    }
21
22 }
23

```

- b) Calcula el precio del viaje y almacénalo en el atributo correspondiente del objeto
- c) En “Gastos recientes:”, muestra en una nueva fila el último gasto añadido usando convertToJSON().
- d) Actualizará el gasto total correspondiente en el apartado “Gastos Totales:”
- e) Dejar el formulario en blanco de nuevo

He creado una función a la que llamo al actualizar la visualización.

```

// Función para vaciar el formulario
function vaciarFormulario() {
    document.getElementById('vehicle-type').value = ''; // Vaciar el tipo de vehículo
    document.getElementById('date').value = ''; // Vaciar la fecha
    document.getElementById('kilometers').value = ''; // Vaciar los kilómetros
}

```



```
// Actualizar la visualización y recalcular el total
actualizarListaGastos(nuevoGasto);
calcularGastoTotal();
vaciarFormulario();
```

3. AUTOEVALUACIÓN

Ejercicio	Apartado	Puntuación	Justificación
1		10	He trabajado mucho con el MVC por lo que estoy familiarizada con la estructura y el acceso a los diferentes recursos de cada carpeta dentro del código
2	a	10	El manejo de restricciones en HTLM me resulta sencillo
	b	10	Estoy familiarizada con el uso de rutas
	c	9	La lógica de los array me cuesta
	d	9	Pasar datos de un documento a otro mediante claves me cuesta
3	a	9	La creación de objetos es igual que en servidor
	b	2	He tenido muchos problemas para desarrollar estos apartados y mostrar la lista de gasto total actualizada
	c	2	
	d	2	
	e	10	Estoy familiarizada con el uso de este tipo de funciones de la asignatura de DIW
Total		73	



4. BIBLIOGRAFÍA

Para este trabajo no he empleado más que los vídeos del profesor y consultas puntuales a Chat GPT para resolver dudas concretas, sobre todo respecto al apartado 3.

