Módulo 6 - Algoritmos I - Laboratorio

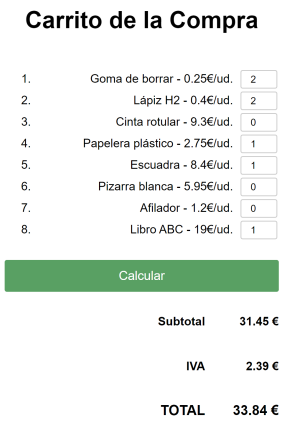
Carrito de la Compra

Introducción

Vamos a implementar un carrito de la compra donde el usuario pueda elegir las unidades que desea de cada producto, a partir de una lista de productos dada.

Tendremos por tanto, una sección describiendo la lista de productos y selectores para elegir las unidades deseadas, un botón para calcular la factura, y una zona con la factura desglosada mostrando el subtotal, impuestos y el importe total.

Un ejemplo de interfaz para que puedas orientarte podría ser el siguiente:



Datos de entrada

Vamos a proporcionarte un ejemplo de lista de productos. Utilízala en tu ejercicio como punto de partida:

```
// Constantes.
const REGULAR_TYPE = 21;
const LOWER_TYPE = 4;
const EXEMPT_TYPE = 0;

// Entrada.
const products = [
  {
    description: "Goma de borrar",
    price: 0.25,
    tax: LOWER_TYPE,
    stock: 2,
    units: 0,
  },
  {
    description: "Lápiz H2",
    price: 0.4,
    tax: LOWER_TYPE,
    stock: 5,
    units: 0,
  },
  {
    description: "Cinta rotular",
    price: 9.3,
    tax: REGULAR_TYPE,
    stock: 2,
    units: 0,
  },
  {
    description: "Papelera plástico",
    price: 2.75,
```

```
    tax: REGULAR_TYPE,
    stock: 5,
    units: 0,
  },
  {
    description: "Escuadra",
    price: 8.4,
    tax: REGULAR_TYPE,
    stock: 3,
    units: 0,
  },
  {
    description: "Pizarra blanca",
    price: 5.95,
    tax: REGULAR_TYPE,
    stock: 2,
    units: 0,
  },
  {
    description: "Afilador",
    price: 1.2,
    tax: LOWER_TYPE,
    stock: 10,
    units: 0,
  },
  {
    description: "Libro ABC",
    price: 19,
    tax: EXEMPT_TYPE,
    stock: 2,
    units: 0,
  },
];
```

Fíjate que cada producto consta de:

- Descripción.
- Precio unitario.
- IVA. Podrá ser normal (21%), reducido (4%) o exento.
- Stock disponible.
- Unidades. Inicialmente 0 para todos los productos.

HTML Dinámico

Hasta ahora, hemos hecho los ejercicios anteriores con HTML estático. Es decir, dada la lista anterior habríais creado elementos html en vuestro fichero `index.html` para todos esos productos. Pero esto es algo poco flexible...

Un carrito de la compra cambia para cada compra y usuario, el contenido de la página (HTML) se genera de forma dinámica para cada lista de la compra. De modo que si tuviésemos una lista totalmente distinta a la planteada anteriormente, nuestra página mostraría dicha nueva lista sin necesidad de tocar HTML. Para ello, será nuestro código Javascript quien genere los elementos HTML en base a la lista dada.

A esto se le llama generar HTML de forma dinámica y va a representar una parte fundamental de nuestro algoritmo en este ejercicio.

Para generar HTML de forma dinámica te explicamos a continuación unas nociones a través del siguiente ejemplo:

```
var input = document.createElement("input");
input.setAttribute("class", "product-unit");
input.setAttribute("type", "number");
input.setAttribute("value", 5);
input.addEventListener("change", event => console.log(event.target.value));

var main = document.getElementById("main");
main.appendChild(input);
```

En dicho código hemos creado un elemento `<input>`, hemos modificado ciertas propiedades, registrado un `eventListener` para el evento `change` y finalmente lo hemos añadido como hijo de otro elemento HTML ya existente cuyo id es `main`.

Hemos usado:

- `createElement` para crear el elemento HTML. Le pasamos como argumento el tipo de elemento que queremos crear (div, input, etc). IMPORTANTE: sólo por crear un elemento no vamos a tenerlo visible en nuestra página, para eso hemos de añadirlo como hijo de algún otro elemento que ya exista en nuestro HTML previamente.
- `appendChild`, para añadir como hijo cualquier elemento HTML.
- `setAttribute` para modificar atributos de un elemento HTML.
- `addEventListener` para registrar event listeners ante cualquier evento que se produzca en el elemento HTML en cuestión.

Cálculo de factura

Al pulsar el botón de calcular, el usuario deberá recibir el desglose de la factura a través de 3 campos: `subtotal`, `impuestos` y `total`.

Para calcular la factura, recorre tu lista de productos con un bucle y haz los cálculos pertinentes para cada producto:

- El precio total de un producto será el `precio unitario * unidades`.
- El IVA a pagar por un producto será `precio total producto * IVA / 100`.

Resumen de pasos

1. Prepara un HTML con la cabecera, el botón de calcular, y los campos de la factura `subtotal`, `impuestos` y `total`. Sin embargo, no hagas ningún elemento para los productos, eso lo debes hacer desde JS. Si necesitaras crear algún `<div>`, identifícalo con algún id que te servirá como punto de entrada para "enganchar" tu lista de productos.
2. Implementa un algoritmo que genere desde Javascript los elementos HTML necesarios para mostrar una lista de productos con su descripción, precio unitario y un input de unidades para cada uno.
3. Prepara un algoritmo que calcule la factura cuando pulsemos el botón calcular y muestre el resultado en los campos `subtotal`, `impuestos` y `total`.

Extra

Intenta hacer que el botón Calcular se habilite o deshabilite en función de si el usuario ha elegido al menos 1 unidad de algún producto o no. Es decir:

- Si las unidades de todos los productos están a 0, el botón calcular se deshabilita.
- Si existe al menos 1 producto con 1 unidad seleccionada, el botón calcular se deberá habilitar.

Para habilitar o deshabilitar un elemento HTML puedes hacer lo siguiente:

```
document.getElementById("button-calculate").disabled = true; // Disabled
document.getElementById("button-calculate").disabled = false; // Enabled
```