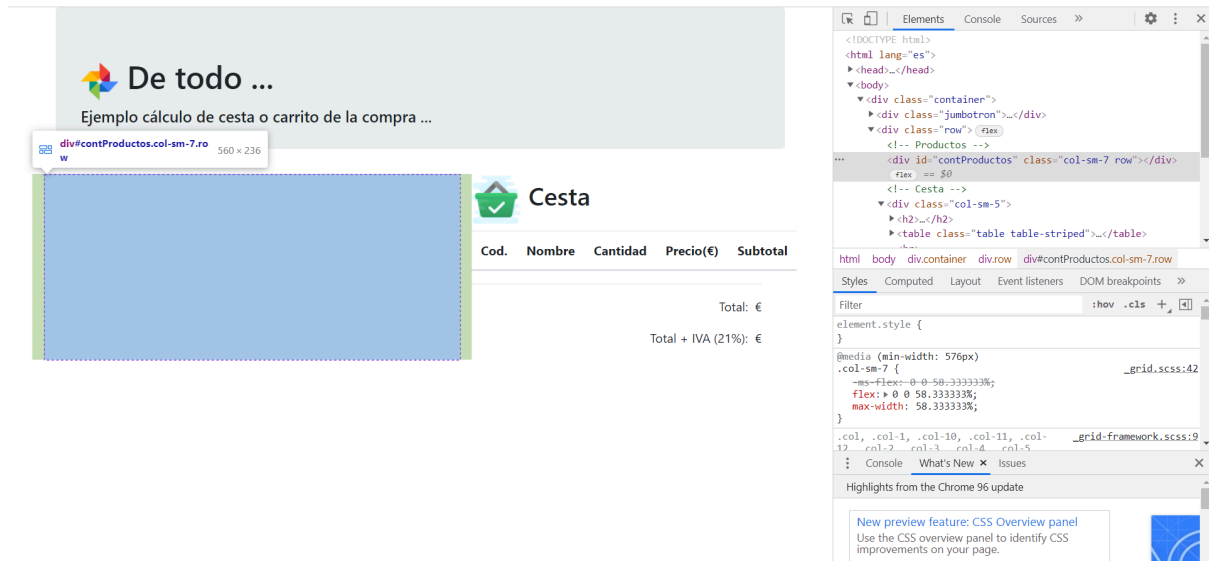


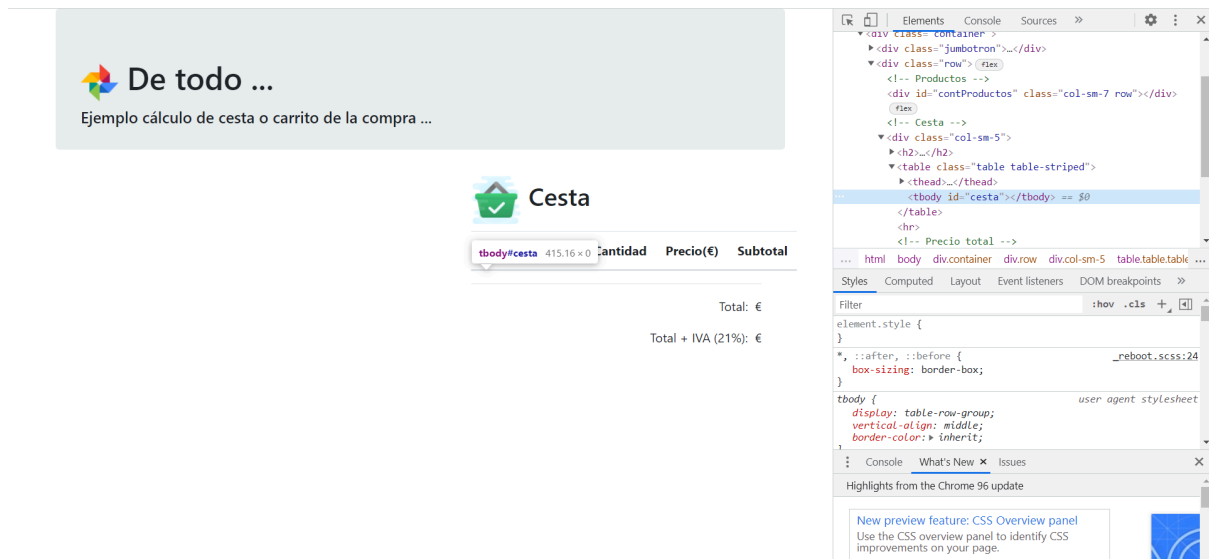
Guión de ayuda para Tarea1T7

1. Creamos con ayuda de Bootstrap la estructura base y estática con elementos HTML.

En la siguiente captura, podemos ver sólo los elementos estáticos y el contenedor **id="contProductos"** en el cual se va a renderizar dinámicamente los artículos o productos disponibles. Este contenedor mantiene la estructura, solo recibe el renderizado de cada grupo/ficha de elementos html (img, nombre, precio, botón ..) de cada producto, creados con los métodos **createElement()**, **setAttribute()**, **appendChild()** (y otros métodos estudiados en el tema ...) , recorriendo todos ellos con el bucle correspondiente.



En esta otra captura se puede ver el identificador **id=cesta** para el **tbody** de una tabla que recibirá dinámicamente los productos que se vayan añadiendo a la cesta.



Guión de ayuda para Tarea1T7

Una cosa importante a tener en cuenta es que durante la creación de los elementos html, éstos, deberán contener también los atributos para identificarlos, para cuando se haga clic sobre ellos, ó al leer el valor del campo en el caso de los **input**. (Ver imagen a continuación, especial atención en los atributos personalizados **data-iduni** del input y **data-idbot** del botón, que poseen el mismo valor identificador para que queden asociados).

The screenshot shows a shopping cart interface. On the left, a grid of product cards is displayed. Each card contains an image, a name, a price, a quantity input field, and an 'Añadir' button. The products are: bolso (20€), Móvil (120€), Taza (10€), Zapatos (50.5€), Vestido (30€), and Cámara (100€). On the right, a 'Cesta' (Cart) section shows a table with the following data:

Cod.	Nombre	Cantidad	Precio(€)	Subtotal
4	Zapatos	5	50.5	252.5
2	Móvil	1	120	120
1	bolso	1	20	20
3	Taza	1	10	10

The total is 402.50 €, and the total with IVA (21%) is 487.02 €.

The Chrome DevTools overlay on the right shows the HTML structure of a product card. The HTML code is as follows:

```
<!-- Productos -->
<div id="contProductos" class="col-sm-7 row" (flex)
  <div class="card col-sm-4" (flex == $0)
    <div class="card-body">
      
      <h5 class="card-title">bolso/h5>
      <p class="card-text">20€/p>
      <input data-iduni="1" type="number" max="999" min="1" value="1" style="width:70px">
      <button class="btn btn-primary" data-idbot="1">
        Añadir</button>
    </div>
  </div>
</div>
```

Y en esta otra, se puede ver el resultado de generar las filas de los productos añadidos a la cesta ...

The screenshot shows the same shopping cart interface as before. The Chrome DevTools overlay on the right shows the HTML structure of the cart table. The HTML code is as follows:

```
<div class="col-sm-5">
  <h2></h2>
  <table class="table table-striped">
    <thead></thead>
    <tbody id="cesta">
      <tr>
        <td>Use $0 in the console to refer to this element.
        <td>Zapatos</td>
        <td>5</td>
        <td>50.5</td>
        <td>252.5</td>
        <button class="btn btn-danger" data-idelim="4">
          X</button>
      </tr>
    </tbody>
  </table>
</div>
```

2. Estructura de clases.

En este ejemplo se ha usado 3 clases, la clase **Producto** (que sólo instancia nuevos productos o artículos), la clase **ContProductos** (que va a instanciar un único objeto con las propiedades y métodos necesarios para renderizar los productos disponibles, podría contener métodos definidos como estáticos) y la clase **Cesta** (que va a instanciar un solo objeto que contiene las propiedades y métodos para renderizar y calcular la cesta, podría contener métodos definidos como estáticos).

A tener en cuenta, para renderizar los productos debe existir una array privado en la clase **ContProductos** que guarde todos los productos disponibles. Igualmente, existirá otro array privado en la clase **Cesta**, que vaya guardando los productos añadidos o comprados, en este caso sólo es necesario que los objetos que forman el array, sólo contengan los campos **id** y **cantidad**, ya que los demás campos se pueden extraer del array productos, que se pasará al crear la instancia de la **Cesta**.

En cualquier caso, cada renderización de productos disponibles y de los productos añadidos a la cesta, se hará desde sus arrays correspondientes recorridos por bucles.

Lógicamente, este guión podría ser más extenso y descriptivo, pero no debo hacerlo..., así os ponéis a prueba...