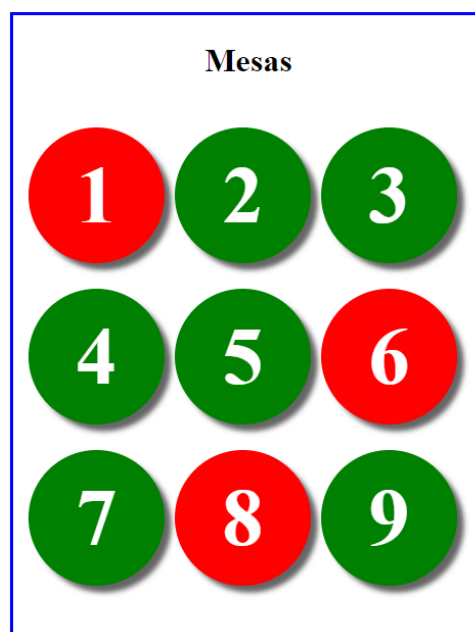
	<p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Bloque II: Práctica Obligatoria</p>	<p align="center">DEPARTAMENTO DE INFORMÁTICA</p>
---	--	--

Una cafetería con más de veinte años de historia en Montequinto decide digitalizar la gestión de las consumiciones de las mesas que tiene en el local. Para ello nos solicita el desarrollo de una aplicación web parecida a la que ya tienen muchos bares y cafeterías actualmente. Es muy importante el concepto de usabilidad ya que el personal de hostelería suele tener una alta rotación laboral. Uno de los objetivos del proyecto es que la curva de aprendizaje de uso de la aplicación no debe ser muy pronunciada, de manera que si se incorpora personal nuevo no tarde mucho tiempo en familiarizarse con el uso de la misma.

La aplicación consta de tres paneles fundamentales:

1. Panel de Mesas: donde se podrá ver gráficamente el estado de todas las mesas del local. En nuestro caso debemos contemplar que el local tendrá nueve mesas. Las mesas de color rojo estarán ocupadas y tendrán una cuenta abierta. Las de color verde estarán libres a la espera de ser ocupadas por nuevos clientes.



2. Panel de Controles: donde se podrá visualizar todos los productos disponibles que podemos ofrecer a los clientes categorizados por el tipo de alimento. Además tendrá una botonera para indicar el número de unidades del producto seleccionado que se desea añadir a la cuenta de la mesa seleccionada.

Controles

Categoría:

Tostadas ▾

Productos:

Media con aceite y jamón ▾

Unidades:

1	2	3
4	5	6
7	8	9

3. Panel de Cuenta: donde se mostrarán las consumiciones de la mesa seleccionada, con el importe total de la mesa. Además habrá un botón que permitirá cerrar la cuenta, cobrarle al cliente y liberar la mesa para que pueda ser ocupada de nuevo. En el caso de que la mesa seleccionada estuviera libre, solo aparecerá un mensaje de la mesa actual.

<h3>Cuenta</h3> <p>Mesa 6</p> <p>TOTAL: 26.80€</p> <div style="border: 1px solid blue; padding: 2px; display: inline-block; margin: 5px auto; width: 100px;">PAGAR Y LIBERAR MESA</div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">Modificar</th> <th style="width: 5%;">Uds.</th> <th style="width: 5%;">Id.</th> <th style="width: 50%;">Producto</th> <th style="width: 10%;">Precio</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">+ -</td> <td style="text-align: center;">9</td> <td style="text-align: center;">1</td> <td>Café con leche (ud: 0.95€)</td> <td style="text-align: right;">8.55</td> </tr> <tr> <td style="text-align: center;">+ -</td> <td style="text-align: center;">4</td> <td style="text-align: center;">8</td> <td>Entera con aceite y jamón (ud: 2.85€)</td> <td style="text-align: right;">11.40</td> </tr> <tr> <td style="text-align: center;">+ -</td> <td style="text-align: center;">3</td> <td style="text-align: center;">11</td> <td>Media con manteca colorá (ud: 1.45€)</td> <td style="text-align: right;">4.35</td> </tr> <tr> <td style="text-align: center;">+ -</td> <td style="text-align: center;">2</td> <td style="text-align: center;">5</td> <td>Media con aceite (ud: 1.25€)</td> <td style="text-align: right;">2.50</td> </tr> </tbody> </table>	Modificar	Uds.	Id.	Producto	Precio	+ -	9	1	Café con leche (ud: 0.95€)	8.55	+ -	4	8	Entera con aceite y jamón (ud: 2.85€)	11.40	+ -	3	11	Media con manteca colorá (ud: 1.45€)	4.35	+ -	2	5	Media con aceite (ud: 1.25€)	2.50	<h3>Cuenta</h3> <p>Mesa 3</p>
Modificar	Uds.	Id.	Producto	Precio																						
+ -	9	1	Café con leche (ud: 0.95€)	8.55																						
+ -	4	8	Entera con aceite y jamón (ud: 2.85€)	11.40																						
+ -	3	11	Media con manteca colorá (ud: 1.45€)	4.35																						
+ -	2	5	Media con aceite (ud: 1.25€)	2.50																						

Por ello el conjunto de los tres paneles que se visualizan simultáneamente sería así:

Gestor de Cafetería

Mesas

1

2

3

4

5

6

7

8

9

Controles

Categoría:

Productos:

Unidades:

1

2

3

4

5

6

7

8

9

Cuenta

Mesa 8


TOTAL: 27.60€

PAGAR Y LIBERAR MESA

Modificar	Uds.	Id.	Producto	Precio
+ -	6	1	Café con leche (ud: 0.95€)	5.70
+ -	1	3	Cola-caó (ud: 1.35€)	1.35
+ -	2	4	Chocolate a la taza (ud: 1.95€)	3.90
+ -	2	16	Caña de chocolate (ud: 1.35€)	2.70
+ -	1	13	Croissant (ud: 0.95€)	0.95
+ -	3	8	Entera con aceite y jamón (ud: 2.85€)	8.55
+ -	2	9	Media con mantequilla (ud: 1.15€)	2.30
+ -	1	12	Entera con manteca colorá (ud: 2.15€)	2.15

El modelo de datos tendrá la siguientes especificaciones con la explicación de atributos y métodos más significativos:


1. Array de String de categorías donde especificaremos las diferentes categorías de alimentos que ofreceremos a nuestros clientes.
2. Clase Producto. Cada objeto de esta clase representará un producto distinto que le ofrecemos a los clientes.
 - a. idProducto: entero que identifica de forma única al producto.
 - b. nombreProducto: cadena de texto con el nombre del producto.
 - c. precioUnidad: número real con el precio de una unidad del producto.
 - d. idCategoria: entero con el índice de la categoría del producto. Coincidirá con la posición en el array de categorías.
3. Clase Catálogo. Con un objeto de esta clase se representa el conjunto de productos que le ofrecemos a los clientes de nuestra cafetería.
 - a. productos: array de objetos de la clase Producto.
 - b. addProducto(idProducto, nombreProducto, precioUnidad, idCategoria).
Método que permite la inserción de un nuevo producto en el catálogo.
4. Clase LineaCuenta. Cada objeto de esta clase representará una línea del pedido de la cuenta de una determinada mesa.
 - a. unidades: entero con el número de unidades solicitadas del producto representado en el atributo siguiente.
 - b. idProducto: entero con el identificador del producto solicitado.

	<p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Bloque II: Práctica Obligatoria</p>	<p align="center">DEPARTAMENTO DE INFORMÁTICA</p>
---	--	--

5. Clase Cuenta. Cada objeto de esta clase representará la cuenta abierta de una determinada mesa de la cafetería.
 - a. mesa: entero que representa la mesa de la cuenta.
 - b. lineasDeCuenta: array de objetos de la clase LineaCuenta donde se almacenan las distintas peticiones realizadas por los clientes de la mesa. Hay que tener en cuenta que no se permitirán líneas de cuenta distintas con un mismo producto. Para añadir más unidades de una línea que contiene un producto ya pedido, se deberá incrementar las unidades de la línea introducida previamente.
 - c. pagada: booleano que representa si la mesa tiene cuenta abierta y no pagada (false) o está libre (true).
6. Clase Gestor. Se creará un objeto de esta clase que servirá para llevar las cuentas abiertas de las distintas mesas de forma simultánea, permitiendo cambiar de una mesa a la otra de forma inmediata.
 - a. cuentas: array de objetos de la clase Cuenta. La posición dentro del array determinará a qué mesa pertenece cada cuenta abierta.
 - b. mesaActual: entero que representa la mesa seleccionada y que será la que estará operativa en el panel Cuenta y se podrá modificar a través del panel Controles.

El funcionamiento normal de la aplicación atenderá a las siguientes especificaciones:

1. Cuando se inicia la aplicación todas las mesas estarán libres, se habrán cargado todos los productos que se ofrecen en el catálogo y se pondrá la mesa 1 como actual.
2. En el panel de controles se podrá modificar la categoría de productos seleccionada, debiendo ofrecerse en el desplegable de productos solo aquellos que pertenecen a la categoría seleccionada.
3. Para abrir la cuenta de una mesa, y que esta pase al estado de ocupada, solo hay que añadir una línea de pedido a su cuenta. Permanecerá ocupada hasta que se pulse el botón “PAGAR Y LIBERAR MESA” que simulará el proceso de cobrarle al cliente pasando la mesa a estar libre y su cuenta a no contener ninguna línea de pedido.
4. En todo momento se podrá alternar de una mesa a la otra pulsando sobre cada una de ellas con el ratón. Cuando se seleccione una determinada mesa como activa, se deberá mostrar su cuenta abierta en el panel de Cuenta y se podrá seguir añadiendo productos en ella.
5. En cada línea de pedido de la cuenta de una mesa aparecerán dos botones que permitirán incrementar en uno y disminuir en uno las unidades seleccionadas del producto. Se deberá preguntar al usuario si realmente quiere eliminar la última

	<p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Bloque II: Práctica Obligatoria</p>	<p align="center">DEPARTAMENTO DE INFORMÁTICA</p>
---	--	--

unidad de un producto, eliminando la línea de la cuenta de la mesa en caso de confirmación.

6. En todo momento se deberá mostrar el total de la cuenta actualizado. En el caso de añadir nuevas líneas de pedido o modificar el número de unidades de las líneas existentes, se deberá refrescar el total adecuadamente.
7. Para añadir una línea de pedido a la cuenta de la mesa actual solo hay que pulsar el número de unidades del producto seleccionado que ha solicitado el cliente. Solo se podrán añadir líneas de 1 a 9 productos. Si se solicitan más, se deberá añadir nueve y añadir más usando los botones de la línea de pedido.
8. Cuando se intente añadir una nueva línea de un producto del que ya existe una línea anterior en la cuenta, se deberá informar de tal circunstancia al usuario y no se permitirá añadirla, sugiriendo que se usen los botones de modificación de unidades de la línea de pedido.

Técnicamente habrá que responder a los siguientes requisitos:

1. Se facilita una plantilla para poder empezar a trabajar con el proyecto. Esta contiene los siguientes ficheros:
 - a. **practicaObligatoria.html**. No se podrá modificar bajo ningún concepto.
 - b. **practicaObligatoria.css**. Hoja de estilo que se propone para facilitar el diseño de la aplicación. Se han preparado clases para visualizar correctamente las mesas libres y ocupadas en el panel de mesas.
 - c. **clases.js**. Donde se deberá incluir la definición de clases del modelo de datos de la aplicación.
 - d. **practicaObligatoria.js**. Donde se definirá el código javascript para desarrollar las diferentes funcionalidades descritas anteriormente.
2. Todo el proyecto se implementará usando el estándar ES6 de programación orientada a objetos.
3. Se valorará positivamente la claridad del código, aportación de comentarios y cuestiones técnicas pertenecientes al contenido del módulo profesional (delegación de eventos, recorrido del DOM, añadir contenido nuevo, ...)

Entrega:

- La implementación de esta práctica se debe realizar de forma individual.
- Cada estudiante será responsable de finalizar el desarrollo y realizar la entrega antes de la fecha y hora tope puesta en la tarea correspondiente del aula virtual.
- Comprimir la carpeta del proyecto en un zip cuyo nombre debe ser Apellido1Apellido2Nombre.zip (Ejemplo: Manuel Gómez Ramírez → GomezRamirezManuel.zip) y entregarlo en la tarea del aula virtual.