

Primer Parcial – Parte Práctica

Nombre y apellidos: _____

NOTA

Necesitamos una aplicación Web para gestionar los **coches** que hay en un **parking**. Aunque la tarea está dividida en ejercicios, toda esta tarea es un único gran ejercicio, y dividirla solo sirve para ayudaros a saber los pasos que debéis realizar para completarla. Se facilita una carpeta con los archivos principales:

- **coches.html**
- **estilos.css**
- **coche.js**: Contiene la clase coche **Coche** con las propiedades para almacenar los siguientes datos:
 - Modelo.
 - Marca (puede ser “Seat”, “Renault” o Citroen”) o Matrícula (única para cada coche que tengamos).
 - Color (puede ser “Rojo”, “Amarillo”, “Verde”, “Blanco” o “Gris”).

La puntuación de esta parte es **de 8 puntos sobre 10 puntos**. Para poder hacer media con la otra partes, se debe obtener al menos un 3,5 para superar el examen la suma de ambas partes debe ser igual o mayor a 5 puntos.

Ejercicio 1 [1,7 puntos]

- Se pide implementar una clase **Parking** en el fichero **parking.js** que contiene:
- Una propiedad **_coches** consistente en un array o un objeto Map (tu decisión) destinado a almacenar objetos de tipo Coche. Ten en cuenta que no se pueden repetir coches con la misma matrícula.
 - Una propiedad **_numMaxCoches** de tipo entero que establece el nº máximo de objetos de coches que podemos almacenar en el parking.
 - Una propiedad **_accesos** para almacenar la fecha y la hora de entrada de cada coche, usando la matrícula como clave. Ejemplo: {matricula: fecha}
 - Una propiedad **_costeMinuto** que establece el coste de un minuto de permanencia en el parking.
 - Al menos los siguientes métodos:

Método (2,5 puntos)	Descripción
Constructor(int coches, int costeMinuto) (0,15 puntos)	Recibe como parámetro el número máximo de coches que podemos almacenar. Y el coste de permanecer un minuto en el parking, puede crearse como una variable global.
get NumeroCoches(): entero (0,1 puntos)	Devuelve el número total de coches almacenados
actualizarImporte() (0,25 puntos)	Actualiza el texto Importe por minuto en función del valor que se pase en el constructor del parking.

<code>addCoche(marca,modelo,matricula,color): boolean</code> (Se evalúa junto con ejercicio 3)	Añade un coche. Previamente a insertar deberíamos comprobar si existe la matrícula del coche y si caben más coches (se ha alcanzado el máximo). En caso de que se pueda insertar el coche el método devuelve verdadero, en caso contrario devuelve falso. Además, debe añadir el coche a accesos.
<code>eliminaCoche(matricula):boolean</code> (Se evalúa junto con ejercicio 5)	Elimina el coche cuya matrícula coincide con la que se pasa como parámetro. En caso de existir el coche y eliminarse se devuelve true, en caso contrario se devuelve false. El método es independiente de mayúsculas/minúsculas. Recuerda eliminar también el coche de accesos.
<code>modificaCoche(matricula, modelo, marca, color):boolean</code> (0,6 puntos)	Modifica los datos de un coche filtrando por matrícula.
<code>calculaImporte(matricula): number</code> (Se evalúa junto con ejercicio 5)	Calcula el importe total en función del tiempo transcurrido desde la fecha de entrada hasta la fecha actual. El importe a cobrar debe redondearse al minuto más cercano. Si el coche se queda más de un segundo, se cobrará por el minuto completo.
<code>actualizarListado()</code> (Se evalúa junto con ejercicio 4)	Actualiza la zona principal del parking. Se llama desde addCoche, eliminaCoche, modificaCoche

Parking IES Julián Marías

Fecha actual: Lunes, 25 de noviembre de 2024, 12:38:08
Importe por minuto: 0.05 €/min

Marca

☐ Seat ☒ Renault ☐ Citroen

Modelo

Megane

Matrícula

8956 LKM

Color

Rojo

Limpiar

Guardar coche

Modificar coche

No hay coches en el parking.

Zona de SALIDA (Arrastra aquí para eliminar)

NOTA: Ten en cuenta que existe una zona información que aparece oculta.

Se debe crear un archivo **principal.js** que contendrá lo siguiente: **(0,6 puntos)**

- Una instancia de **Parking** (puede tratarse como una variable global) con un máximo de 6 coches, y con un coste de 0,05€ el minuto.
- Una función que actualice cada segundo la fecha y hora actual.
- Los manejadores de eventos para que se traten correctamente las pulsaciones de botones y los drag and drop.
- Además, podremos tener todas las funciones auxiliares que consideres necesarias.

Ejercicio 2 Validaciones [0,8 puntos]

Se debe crear un fichero denominado **utils.js** que contenga una función **mostrarMensaje** que debe recibir el mensaje y el tipo de mensaje (error,succes) y en función del tipo aplicará las clases **mensaje-success**, y **mensaje-error** facilitadas en el **estilos.css**.

El mensaje debe mostrarse durante 5 segundos.

Mensajes de Éxito (0,2 puntos)

Al pulsar el botón **Guardar coche** deberán hacerse las validaciones que se describen a continuación. En caso de que todos los datos sean correctos se añadirá un nuevo coche al parking y se mostrará el siguiente mensaje indicativo en la capa con id **información**:

El coche se ha añadido al parking correctamente.

Al pulsar el botón **Modificar Coche** se deberán hacer nuevamente las comprobaciones y en caso de que todos los datos sean correctos se modificará el coche por matrícula y se mostrará el siguiente mensaje:

El coche se ha modificado correctamente.

Mensajes de Validación (0,6 puntos)

- Todos los campos son obligatorios. Esto implica que si algún campo está vacío no se añadirá el coche al parking, y se mostrará un mensaje indicativo en el div **informacion**, por ejemplo:

Todos los campos son obligatorios.

Se considera un campo de texto vacío cuando no se ha introducido texto o se han introducido únicamente espacios.

- Se utilizará una expresión regular para comprobar la validez de la matrícula. Esta debe de obedecer al formato **NNNN LLL** (siendo N un número del 0 al 9 y L una letra). En caso de ser incorrecta la matrícula, se mostrará un mensaje indicativo, pasando el foco al campo de matrícula:

La matrícula no tiene un formato válido (NNNN LLL).

- Si los datos son correctos, pero ya existía un coche con la matrícula que se quiere insertar, también se mostrará un mensaje indicativo del error, pasando el foco al campo de matrícula:

El coche ya existe en el parking.

- Si intentamos modificar o eliminar un coche que no existe en el parking:

El coche no existe en el parking.

- Lo mismo ocurrirá si se ha alcanzado el máximo número de coches:

El parking está lleno.

Ejercicio 3 Inserción [1,5 punto]

Al rellenar correctamente todos los campos del formulario y pulsar en “Guardar coche” se debe almacenar el coche si no existe su matrícula, y establecer el color de fondo del div elegido en el desplegable. Revisa en el css las clases de coche que puedan ser de utilidad.

Parking IES Julián Marías

Fecha actual: lunes, 25 de noviembre de 2024, 15:45:29
Importe por minuto: 0.05 €/min

Marca

☒ Seat ☐ Renault ☐ Citroen

Modelo

Matrícula

Color

Limpiar

Guardar coche

Modificar coche

Modelo: ibiza
Marca: seat
Matrícula: 3256 HYM
Color: amarillo
Hora de entrada:
25/11/2024, 12:41:10

Zona de SALIDA (Arrastra aquí para eliminar)

Al pulsar sobre un coche se deben cargar en el formulario los datos de ese coche. El botón Limpiar debe dejar el formulario en su estado inicial.

Ejercicio 4 ListadoActualizado [1 punto]

Haz que la capa con id **listado** muestre en todo momento el listado de coches actual usando de color de fondo del div de cada coche el color del coche (puedes meter estilos en el JS).

Parking IES Julián Marías

Fecha actual: lunes, 25 de noviembre de 2024, 12:37:17
Importe por minuto: 0.05 €/min

Marca

○ Seat ● Renault ○ Citroen

Modelo

Megane

Matricula

8956 LKM

Color

Rojo

Limpiar

Guardar coche

Modificar coche

Modelo: Ibiza
Marca: seat
Matricula: 1254 KJM
Color: blanco
Hora de entrada: 25/11/2024, 12:36:26

Modelo: Megane
Marca: renault
Matricula: 8956 LKM
Color: rojo
Hora de entrada: 25/11/2024, 12:36:53

Zona de SALIDA (Arrastra aquí para eliminar)

Ejercicio 5 Eliminación [3 puntos]

Al arrastrar un coche a la zona de salida se debe eliminar el coche del parking.

Parking IES Julián Marías

Fecha actual: lunes, 25 de noviembre de 2024, 12:34:54
Importe por minuto: 0.05 €/min

Marca

● Seat ○ Renault ○ Citroen

Modelo

Tiguan

Matricula

1254 KJM

Color

Blanco

Limpiar

Guardar coche

Modificar coche

Modelo: Tiguan
Marca: seat
Matricula: 1254 KJM
Color: blanco
Hora de entrada: 25/11/2024, 12:34:45

Zona de SALIDA (Arrastra aquí para eliminar)

Parking IES Julián Marías

Fecha actual: lunes, 25 de noviembre de 2024, 12:35:19
Importe por minuto: 0.05 €/min

Marca

● Seat ○ Renault ○ Citroen

Modelo

Tiguan

Matricula

1254 KJM

Color

Blanco

Limpiar

Guardar coche

Modificar coche

No hay coches en el parking.

Zona de SALIDA (Arrastra aquí para eliminar)

Al eliminar el coche, se calcula el tiempo de permanencia y el importe correspondiente y se mostrará un mensaje indicativo como el que se muestra. El coche debe ser eliminado también de accesos.

El coche con matrícula 1234 FFF salió tras 7 minutos. Importe: 0.35€

Desarrollo Web en Entorno Cliente – Laura Isabel López Naves

CC BY NC SA

Curso 2024-2025