

AD + PSP

TEMA 2. JAVASCRIPT

PROYECTO CON JAVASCRIPT

El proyecto consiste en realizar una aplicación para gestionar un parking robotizado. Es necesario que el parking gestione los clientes que se abonan al parking, así como la información de usos de las plazas del parking por los distintos vehículos que entran y salen. El parking dispone de 45 plazas en total: 15 son específicas para turismos, 15 para aparcar motocicletas y 15 para caravanas. La tarificación por minutos es la siguiente:

- Turismos - 0,12 €
- Motocicletas - 0,08 €
- Caravanas - 0,45 €

La aplicación tiene dos subsistemas: **una zona cliente y una zona administrador.**

ZONA CLIENTE

En la zona cliente de la aplicación se pueden realizar básicamente las siguientes acciones:

Depositar vehículo

- El sistema informa en todo momento del número de plazas libres que existen de cada tipo.
- El cliente introduce la matrícula y el tipo (turismo, motocicleta o caravana). El sistema asigna una plaza de las posibles, si existen plazas libres. Automáticamente el montacargas del parking guarda el vehículo en la plaza asignada.
- El sistema genera un ticket donde aparece la matrícula del vehículo, la fecha de depósito, el identificador de la plaza asignada y un pin de seis dígitos numéricos que servirá para retirar el vehículo posteriormente. Este ticket aparecerá en la consola del sistema. Esta información se debe guardar **de forma persistente**¹ para poder ser consultada cuando el cliente proceda a la retirada del vehículo.
- No es necesario guardar información de los clientes si se hace uso del parking sin abono.

Retirar vehículo

- El cliente introduce la matrícula, el identificador de la plaza y el pin asociado. El sistema calcula el coste total a pagar e informa de la tarifa al cliente.
- Una vez realizado el pago, el sistema devuelve el vehículo y actualiza el número de plazas libres, así como la información relativa al coste final y fecha de salida del vehículo.
- Los cobros realizados se deben guardar en una colección para, posteriormente, poder consultarlos desde la zona de administración.

¹ Como por ahora no sabemos acceder a una base de datos desde Javascript, lo guardaremos en una colección del tipo que creas más conveniente.

Depositar abonados

- El cliente abonado introduce en el sistema la matrícula del vehículo y su DNI. Se supone que un cliente tiene un solo vehículo y un vehículo pertenece a un solo cliente.
- El sistema aparca el vehículo en la plaza asignada al abonado y actualiza el estado de la plaza para saber que el vehículo del abonado está en el parking. Asocia siempre el mismo pin para poder retirar el vehículo tantas veces como sea necesario. El PIN se genera al crear el abono del cliente.
- De los clientes abonados es necesario saber su DNI, nombre, apellidos, número de tarjeta de crédito, tipo de abono que tienen y su email.

Retirar abonados

- El cliente introduce la matrícula, el identificador de plaza asignada y el pin.
- El sistema actualiza el estado de la plaza del parking, que no queda libre, sigue estando reservada, pero el vehículo del abonado no está en el parking.

ZONA ADMINISTRADOR

La zona admin de la aplicación se encarga de:

Estado del parking

Controlar el estado del parking. Se debe mostrar por consola el estado de las plazas (libre, ocupada, abono ocupada y abono libre) y el identificador de cada plaza².

Facturación

Entre fechas. El sistema solicita dos fechas y horas concretas para saber los cobros realizados entre las mismas. Los abonos no se contemplan en esta opción.

Consulta de Abonados.

El sistema informa de los abonos anuales, con los cobros realizados.

Abonos

Alta. El sistema solicita datos personales del abonado y un número de tarjeta donde se realizan los cargos mensuales del abono. El cliente debe elegir entre los distintos abonos: mensual (25€), trimestral (70€), semestral (130€) y anual (200€). Todos los abonos tienen una fecha de activación y una fecha de cancelación. La fecha de activación se actualiza con la fecha en la que se da de alta y la fecha de cancelación se calcula en función del tipo de abono.

Modificación. Existirá la opción de cambiar los datos personales del abonado o bien cambiar la fecha de cancelación del abono, porque el abono ha sido renovado.

Baja. Se eliminará el registro del abonado pero no se podrán borrar los datos asociados a su facturación.

Caducidad de abonos

² ¿Eres capaz de pintar por consola el mapa del parking? Sólo para aquellos que hayan terminado el resto de funcionalidades.

Mes. El sistema solicita un mes y nos informa de los abonos que caducan en ese mes.

Consultar últimos 10 días. El programa informa por consola de los abonos que caducan en los siguientes 10 días a la fecha actual.

ELEMENTOS A UTILIZAR EN EL PROYECTO

- Preferentemente, se utilizarán siempre que se pueda la sintaxis o elementos de ES6
- Debe ser orientado a objetos. Allá donde se pueda utilizar una clase, debe ser la opción prioritaria.
- Se debe usar modularización, es decir, que tenemos que repartir nuestro código en diferentes directorios y ficheros para que quede correctamente estructurado. Puedes seguir pensando en términos de repositorios, servicios y controladores, aunque la aplicación no vaya a ser web.

MATERIAL

- Node.js no está especialmente pensado para trabajar en consola, pero existe alguna librería que nos permite hacerlo. Una de ellas es **readline-sync**. Más información en <https://www.npmjs.com/package/readline-sync>
- Como has podido comprobar, hay que manejar fechas. Quizás te interese utilizar la librería moment.js, que nos permite hacer cosas chulas. Echa un vistazo a
 - https://programacion.net/articulo/manejar_fechas_y_horas_como_un_profesional_con_moment_js_1301
 - <https://momentjs.com/>

PLAZOS, ENTREGA y CALIFICACIÓN

Este proyecto tendrá el valor de examen para el Tema 2, si bien complementariamente haremos una prueba escrita sobre el proyecto, para poder confrontar tu aprendizaje.

El proyecto se comienza a desarrollar el lunes 9 de Noviembre y se debe entregar antes del viernes 13 de Noviembre a las 19:00 horas.

Al igual que los demás proyectos de AD+PSP, se corregirá mediante un par de rúbricas, una asociada a AD y otra a PSP.

Como entrega, se debe adjuntar la URL de un repositorio en github donde esté incluido el proyecto.