**Concesionaria de autos**

Este ejercicio es para que puedas poner en práctica los contenidos más importantes de JavaScript que vimos en este primer módulo de cursada. Al realizarlo, vas a poder asegurar que venís bien con los contenidos. La propuesta es ir en forma incremental con los contenidos vistos, por lo cual es normal que a medida que avances se vuelva más complejo y difícil. Si te resulta más sencillo, podés intentar resolver la ejercitación en VS Code y luego poner la respuesta en Playground. Si en algún momento no lográs avanzar y estás perdido, no dudes en consultar a tus compañeros para lograr comprender qué es lo que hay que hacer y qué es lo que hace el código.

Objetivos

* Tener feedback propio de ejercitaciones más complejas e integradoras.
* Probar tus conocimientos y autonomía a la hora de resolver un problema.
* Detectar puntos de mejora.
* Fortalecer las bases y conceptos de programación con JavaScript.

Metodología

Vamos a simular una situación de trabajo para situarnos en un contexto. Estás trabajando como desarrollador de CodeAR S.A., una reconocida software factory. En el equipo de trabajo contás con María y con Juan.

*Maria  
  
Es líder técnica del área. Es una persona muy ocupada y su rol consiste en traducir las necesidades del cliente a requerimientos técnicos.*

*Juan  
  
Es pasante y está empezando en el mundo de la programación. Si bien es algo tímido, le gusta aprender cosas nuevas y encarar nuevos desafíos.*

*Vos  
  
Si bien estás aprendiendo cosas como Juan, ya contás con algunos conocimientos previos.*

Primera semana de trabajo

María presenta al equipo un nuevo proyecto para una concesionaria de automóviles, cuya principal línea de negocios es la compra y venta de automóviles. La concesionaria necesita construir una lista con todos los vehículos que tiene registrados. Por cada vehículo necesita conocer la siguiente información:

* Marca (Ford, Fiat, etc). (marca)
* Modelo (Fiesta, Corolla, etc). (modelo)
* Color (Azul, Rojo, etc). (color)
* Año de fabricación. (año)
* Cantidad de kilómetros recorridos. (km)
* Precio final. (precio)
* Cantidad de cuotas. (cuotas)
* Patente. (patente)
* Vendido: para indicar si el auto está o no vendido. (vendido)

**María quiere saber con qué tipo de dato representarán a cada vehículo individual. Juan pensó en 3 opciones para el vehículo individual. ¿Podrías ayudarlo a elegir la correcta?**

Respuesta: Un array conteniendo objetos literales con las propiedades de cada auto.

**Aclarado el diseño del sistema. María tiene mucha experiencia, es súper profesional y está un paso adelante nuestro, por eso ella escribió en forma anticipada los tests de nuestro código, para asegurarse que el mismo funcione bien. Además, le pidió algunas tareas a Juan, quien te pregunta cómo él va a poder acceder a los objetos literales que vas a crear si el trabaja en otro archivo.**

Respuesta: Exportando el contenido de la variable mediante "module.exports" y requiriéndolo en los archivos que lo necesiten.

En base a las definiciones técnicas tomadas con el equipo deberás declarar la variable autos. Esta debe contener los siguientes vehículos:

1. El primer auto es un Ford Fiesta Azul, del 2019, con 200 kilómetros, cuyo precio es 150000, disponible en 12 cuotas, con la patente APL123 que no está vendido.
2. El segundo auto es un Toyota Corolla Blanco, del 2019, 0 kilómetros, cuyo precio es 100000, disponible en 14 cuotas, con la patente JJK116 que no está vendido.

Cada auto debe tener los siguientes atributos: marca, modelo, precio, km, color, cuotas, anio, patente, vendido.

EN VISUAL STUDIO CODE

A la semana, el proyecto avanza y María nos pide otro requerimiento, esta vez un poco más avanzado. Nos comenta que la concesionaria ahora necesita poder buscar los automóviles por patente y venderlos. Para esto, María nos encargó pensar la forma en que representaremos a la concesionaria.

Al terminar la reunión, Juan se acerca para comentarte tres ideas que tuvo. María, a unos metros, escuchó las opciones y te encargó definir cuál será la más eficaz.

Respuesta: Podríamos tener un objeto literal que represente a la concesionaria con los autos que creamos y que esta tenga las funcionalidades de buscar por patente y marcar como vendido un auto.

Ahora que tenemos creada nuestra lista de autos y definida la manera de representar nuestra concesionaria, necesitamos crearla. El objeto concesionaria va a conocer a todos los autos y va disponer de las funcionalidades de buscarlos y venderlos. Pero para no entrar en demasiado detalle, María propone trabajar en tres etapas y en cada una agregar valor. Respondé a las necesidades de cada etapa.

# Etapa 1

En esta primera etapa, necesitamos requerir tu módulo autos que se encuentra en la misma carpeta del archivo donde estás trabajando.

Además, necesitarás crear un objeto literal llamado concesionaria que contendrá todas las funcionalidades que el cliente solicita.

Por último, nuestro objeto literal debe tener un atributo llamado autos que contenga la lista de automóviles importada anteriormente.

# Etapa 2

Ahora que la concesionaria tiene los autos, es posible crear la funcionalidad buscarAuto que reciba por parámetro la patente y devuelva el auto al cual le corresponde. En caso de no encontrar el mismo, deberá retornar null.

# Etapa 3

Ahora, María les pide que agreguen la funcionalidad de venderAuto que recibe la patente y, en caso de encontrar al automóvil, le asigna el estado de vendido.

¿Cómo lo resuelven?

**Recordatorio**: Para comenzar tenés que agregar el código que escribiste en el ejercicio anterior. Para resolver esta nueva funcionalidad, tendrás que utilizar la función buscarAuto.

# Funcionalidad extra

La primera es poder contar, como concesionaria, con la habilidad de poder tener la lista de autos para la venta. A lo cual, María, cree que es una tarea sencilla que Juan y vos pueden encarar solos, usando la función autosParaLaVenta, aunque por las dudas ella les recuerda que no deberían de aparecer los autos que ya fueron vendidos.

Para comenzar, tenés que agregar el código que escribiste en el ejercicio anterior. Tené en cuenta que estamos optimizando nuestro código, por lo cual, deberíamos utilizar el método filter.

# Una nueva funcionalidad extra

María, contenta con el trabajo que realizaron, les pide otra funcionalidad extra. Resulta que a la concesionaria le suelen preguntar muy seguido cuáles de los autos **para la venta**son 0 km. Tené en cuenta que María considera que un auto 0 km es aquel que tenga un kilometraje menor a 100. Vas a tener que desarrollar la funcionalidad autosNuevos.

¿Cómo podés resolver esto reutilizando la función autosParaLaVenta?

# Más funcionalidades

El cliente le pidió saber cuánto dinero generaron las ventas.

María te pide que completes la función listaDeVentas que devuelve una lista que contiene el precio de venta de cada auto vendido. A esto, Juan, que está al lado tuyo, se le escapa la frase "mmm.....estoy seguro que alguna función de arrays nos va a servir, pero no me acuerdo".

# Total de ventas

Terminada esta función, María te pide que resuelvas la funcionalidad de totalDeVentas, que justamente nos devuelva la sumatoria del valor de todas las ventas realizadas. Acá el único requerimiento técnico explícito es que utilices la función reduce, ¡a codear!

# Agregando funcionalidades

Muy contento el equipo por cómo viene el desarrollo, por la tarde, María te comenta que se agrega una funcionalidad muy importante: la de verificar si una persona puede comprar o no un auto. Esta permite al sistema definir si una persona al consultar por un auto, puede comprarlo. Las personas solo sacan los autos en cuotas y tomando dos factores como condición de compra. Una es el **costo total**: si el total de un auto excede lo que la persona considera caro, no va a comprar el auto. Otra condición es su **capacidad de pago en cuotas**: si la capacidad de pago en cuotas supera al costo de la cuota, va a poder pagarlo. Si **ambas condiciones se cumplen**, se realiza la compra.

Es por esto que María te pide que desarrolles la función puedeComprarque reciba por parámetro un auto y una persona y devuelva true si la misma puede comprar el auto.

Una persona va a ser representada mediante un objeto literal de la siguiente forma:

{  
nombre: “Juan”,  
capacidadDePagoEnCuotas: 20000,  
capacidadDePagoTotal: 100000  
}

Al finalizar de desarrollar la nueva funcionalidad, le proponés al equipo que sería relativamente fácil agregar la funcionalidad que: dada una persona, se le pueda ofrecer fácilmente qué autos puede comprar así no tiene que estar preguntando uno por uno. A Juan y María les encanta la idea, María se la menciona al cliente y esperan con ansias esa nueva funcionalidad.

# Agregando funcionalidades

Ahora, te comprometiste a realizarla. Así que manos a la obra. Hay que escribir la función autosQuePuedeComprar, que recibe una persona y devuelve la lista de autos que puede comprar.

La función debe de realizar los siguientes pasos:

1) Obtener los autos para la venta

2) Por cada uno de los autos debe de probar si la persona puede comprarlo, ¿ya hay alguna funcionalidad que me permita hacer esto?.

3) Luego debemos retornar los que pueda comprar, ¿hay alguna manera de poder filtrar la lista de autos para la venta del punto 1 con el paso 2?