****

**Programación Imperativa**

# Segundo Parcial

# **Objetivo**

Integrar los conocimientos adquiridos hasta el momento en la cursada.

Practicar la lectura, comprensión e interpretación de consignas y producir código que resuelva los enunciados propuestos.

## **Buenas prácticas**

A la hora de la resolución no sólo es importante que el código funcione correctamente, sino también que sea legible y respete las buenas formas.

* Que las variables, métodos y funciones tengan nombres descriptivos
* Que se utilice camelCase donde corresponda
* Que se utilicen los métodos más adecuados para resolver cada problema
* Que el código produzca el resultado esperado a partir de los datos suministrados

# Introducción

Un emprendimiento de venta de libros necesita crear un pequeño sistema que le permita facilitar la gestión de datos y algunas otras funcionalidades.

El mismo tiene una base de información que podremos encontrar en un array de objetos literales, que representan a cada uno de los libros, al comienzo del archivo.



# Para comenzar

Deberán descargar la siguiente [PLANTILLA](https://drive.google.com/file/d/1sKBDA-lBri24uU20YuuZpsHa7Z7e7m_T/view?usp=sharing) donde encontrarán la estructura inicial que necesitan para una entrega exitosa.

* **app.js,** será donde escribiremos nuestro código para resolver las consignas.

# Consignas Partimos de un objeto ya creado con una propiedad libros, la cual será nuestra base de datos, en función de ello deberemos realizar las siguientes características de nuestra app:

1. Agregar un método **buscarPorTitulo** que permita buscar dentro de la propiedad de nuestro objeto bookstore un libro por su título, el cual deberá ser igual al titulos que se envía como argumento al momento de invocarlo.
   * Este método deberá retornar un libro.
2. Agregar un método **buscarPorAutor** que permita buscar a todos los libros que sean del autor indicado como parámetro
   * Este método deberá retornar un array de libros cuyo autor sea el mismo.
3. Agregar un método **filtrarPorPrecio** que permita filtrar los libros en función de su precio, es decir, todos los libros cuya propiedad precio sea menor o igual a la indicada por parámetro
   * Este método deberá retornar un array de libros.
4. Agregar un método **modificarStock** que permite cambiar el stock de un libro determinado. Este método recibe dos parámetros, el título del libro a modificar, y la nueva cantidad de stock
   * Reutiliza el método **buscarPorTitulo** para obtenerlo.
   * Una vez encontrado el libro deberá cambiar el valor de la propiedad *stock* a la indicada por parámetro.
   * Retorna el libro.
5. Agregar un método **precioPromedio** que permita consultar el promedio de precio de libros.
   * El método deberá recorrer el listado de libros y calcular el promedio de la propiedad precio.
   * El método deberá retornar un mensaje con el siguiente formato.  
     *ej: “El precio promedio de cada libro es de: $1800.”*