Desarrollar un sistema que integre una base de datos con SQLite para una concesionaria.

Mediante un menú de opciones llevar a cabo las siguientes posibilidades.

**REGULARIDAD**   
**1-** Cargar Automoviles.

De cada auto almacenar marca, modelo, precio, cantidadDisponibles.

La marca y modelo declararlo como Único e irrepetible. Llevar también a cabo un id como primary key autoincremental.

**2-** Modificar datos de un auto.

Dado la marca y el modelo del auto, permitir modificar su precio.

**3-** Borrar un auto.

Dado la marca y modelo del auto, permitir eliminarlo de la base datos **4-** Cargar disponibilidad.

Cada vez que ingresa un nuevo automovil de una marca y modelo en especifico, incrementar la cantidad disponible. Solicitar al usuario la marca y modelo e incrementar la cantidad disponible en 1.

**5-** Listado de Automoviles.

Mostrar ordenadamente todos los automoviles cargados.

0- Salir del menú.

**APROBACIÓN DIRECTA**

**6-** Crear una tabla llamada motocicleta que contenga los siguientes atributos: modelo (Varchar(30)), marca (Varchar(30)), cilindrada (Varchar(30)), precio(Integer), color (Varchar(30)), fechaUltimoPrecio (datetime). Llevar a cabo un id   
autoincremental denominado id\_moto. Para esta tabla los atributos marca y modelo

**NO** son unicos e irrepetibles.

**7-** Por el aumento del dolar se decidio actualizar los precios de todas las motos en un 10%. Se desea mantener el historial de registros de precios actuales. Insertar los registros viejos en una tabla llamada “historico\_motocicletas” y actualizar el precio y fecha en la tabla motocicletas .

Para este punto se debe de crear la tabla historico\_motocicletas que tendra exactamente las mismas caracteristicas que la tabla motocicletas. Previo a actualizar los precios en la tabla motocicletas, se debera de insertar los datos actuales en la tabla historico\_motocicletas   
**8-** Mostrar todos los registros anteriores a una fecha en especifico de la tabla motocicletas.

0- Salir del menú.

**CONSIDERACIONES A LA HORA DE CORRECCIÓN:**-Utilización de Programación Orientada a objetos  
-Uso de estructuras de manejo de errores ( try, except, finally)  
-Claridad a la hora de mostrar los mensajes  
-Simplicidad del codigo.

-Optimización del codigo  
-Para este trabajo UN solo integrante del grupo debera de forkear el repositorio y todos los integrantes clonar dicho fork. Deberan de subir los cambios a dicho fork y se debera de visualizar claramente los commits realizados por cada integrante del grupo.

-Luego de la entrega del TPI se deberá realizar una defensa del mismo donde se debera mostrar el sistema realizado y se podran realizar preguntas a cada   
integrante del grupo para visualizar su participación en el trabajo y conceptos de la materia.

-Enviar el link al repositorio al mail : [poncetomaseduardo@gmail.com](mailto:poncetomaseduardo@gmail.com)

**FECHA DE ENTREGA : 18/10/2022**