Buscalibre, quiere abrir su primera casa física en Rosario, para ello se debe desarrollar un sistema que integre una base de datos con SQLite.

**A-** Nos especifican que necesitan un menú de opciones que permita:

## REGULARIDAD

**1- Cargar Libros.**

De cada libro se quiere saber ID, ISBN, Título, Autor, Género, Precio, FechaUltimoPrecio y CantDisponible.

ISBN declararlo como Único e irrepetible. ID es una clave primaria y autoincremental

**2- Modificar precio de un libro**

Dado el ID, permite modificar el  precio de un libro. Deberá ser confirmado por el usuario antes de ejecutar la acción

**3- Borrar un libro.**

Dado el ID, permitir eliminarlo de la base datos. Deberá ser confirmado por el usuario antes de ejecutar la acción

**4- Cargar disponibilidad**.

Que permita, ingresando el ID, incrementar la CantDisponible.

**5- Listado de Libros**

Mostrar ordenadamente todos los libros, por id, autor, titulo

**0- Salir del menú.**

Recuerden utilizar sentencias adecuadas para ejecutar las consultas y actualizaciones en la base de datos. Además, tengan en cuenta manejar posibles errores y validaciones en cada opción del menú.

## APROBACIÓN DIRECTA

**B-** Luego de probar anterior, nos piden agregar, la siguiente mejora:

**6- Ventas**

Crear una tabla Ventas, registrar en esta tabla id del libro vendido, la cantidad y la fecha. Descontar de la tabla Libros la cantidad vendida.

**7- Actualizar Precios**

Por el aumento del dólar se necesita poder actualizar los precios de todas los libros en un porcentaje determinado. Se desea mantener el historial de registros de precios actuales. Insertar los registros viejos en una tabla llamada “historico_libros” y actualizar el precio y fecha en la tabla libros .

Para este punto se debe de crear la tabla historico_libros que tendrá exactamente las mismas características que la tabla libros. Previo a actualizar los precios en la tabla libros, se deberá de insertar los datos actuales en la tabla historico_libros

**8-** Mostrar todos los registros anteriores a una fecha en específico de la tabla libros.

## 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 y optimización del código.
- Para este trabajo UN solo integrante del grupo creará el repositorio y pondrá al resto de los integrantes del grupo como colaboradores. Deberán de subir los cambios y se debe poder 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 deberá mostrar el sistema realizado y se podrán 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 : gabycaliva@gmail.com

baumanpet@gmail.com

**FECHA DE ENTREGA : 14/06/2023**

```
import sqlite3


class ProgramaPrincipal:


    def menu(self):
        while True:
            print("Menu de opciones Concesionaria")
            print("2- Modificar Automovil")
            print("1- Cargar Automovil")
            print("0- Salir de menu")
            nro = int(input("Por favor ingrese un número"))
            if nro == 1:
                marca = input("Por favor ingrese la marca del automovil: ")
                modelo = input("Por favor ingrese el modelo del automovil: ")
                precio = input("Por favor ingrese el precio del automovil: ")
                cantidadDisponibles = input("Por favor ingrese la cantidad de
unidades disponibles: ")
                nuevo_automovil =
Automovil(marca,modelo,precio,cantidadDisponibles)
                nuevo_automovil.cargar_automovil()
```

```
            if nro ==2:
                marca = input("Por favor ingrese el nombre de la marca: ")
                modelo = input("Por favor ingrese el nombre del modelo: ")
                precio = input("Por favor ingrese el nuevo precio: ")
                automovil_a_modificar=Automovil(marca,modelo,precio)
                automovil_a_modificar.modificar_automoviles()
            if nro==0:
                break


    def crearTablas(self):
        conexion = Conexiones()
        conexion.abrirConexion()
        conexion.miCursor.execute("DROP TABLE IF EXISTS AUTOMOVILES")
        conexion.miCursor.execute("CREATE TABLE AUTOMOVILES
(id_automovil INTEGER PRIMARY KEY , marca  VARCHAR(30) ,modelo
VARCHAR(30),precio FLOAT NOT NULL, cantidadDisponibles INTEGER NOT
NULL,UNIQUE(marca,modelo))")
        conexion.miConexion.commit()
        conexion.cerrarConexion()


class Automovil:
    def __init__(self, marca, modelo,precio=None,cantidadDisponibles=None):
        self.marca = marca
        self.modelo = modelo
        self.precio=precio
        self.cantidadDisponibles=cantidadDisponibles
```

```
    def cargar_automovil(self):
        conexion = Conexiones()
        conexion.abrirConexion()
        try:
            conexion.miCursor.execute("INSERT INTO
AUTOMOVILES(marca,modelo,precio,cantidadDisponibles) VALUES('{}',
'{}','{}','{}')".format(self.marca,
self.modelo,self.precio,self.cantidadDisponibles))
            conexion.miConexion.commit()
            print("Automovil cargado exitosamente")
        except:
            print("Error al agregar un automovil")
        finally:
            conexion.cerrarConexion()



    def modificar_automoviles(self):
        conexion = Conexiones()
        conexion.abrirConexion()
        try:
            conexion.miCursor.execute("UPDATE AUTOMOVILES SET precio='{}'
where marca='{}' and modelo='{}' ".format(self.precio,self.marca,self.modelo))
            conexion.miConexion.commit()
            print("Automovil modificado correctamente")
        except:
```

```
            print('Error al actualizar un automovil')
        finally:
            conexion.cerrarConexion()



class Conexiones:


    def abrirConexion(self):
        self.miConexion = sqlite3.connect("Concesionaria")
        self.miCursor = self.miConexion.cursor()


    def cerrarConexion(self):
        self.miConexion.close()




programa = ProgramaPrincipal()
programa.crearTablas()
programa.menu()
```