TRABAJO INTEGRADOR

Módulo Programador

- Profesores: Kessler Kevin – Lanfranco Lisandro.

- Integrantes Grupo 8
  - Birge, Adolfo Federico.
  - Ferreyra, María Luciana.
  - Gutiérrez, Emma Vilma.
  - Paz Rodolfo.

- Repositorio de entrega: https://github.com/FedeBirge/TRABAJO-INTEGRADOR

La inmobiliaria "Bienes Raíces Future", nos ha pedido ayuda con la informatización de sus propiedades. Necesitan poder almacenar las mismas diferenciadas por tipología, y saber si están disponibles o no para su ofrecimiento, ya sea a la venta o alquiler. Si bien es sencilla su necesidad, nos menciona que los datos de las propiedades suelen cambiar con rapidez.

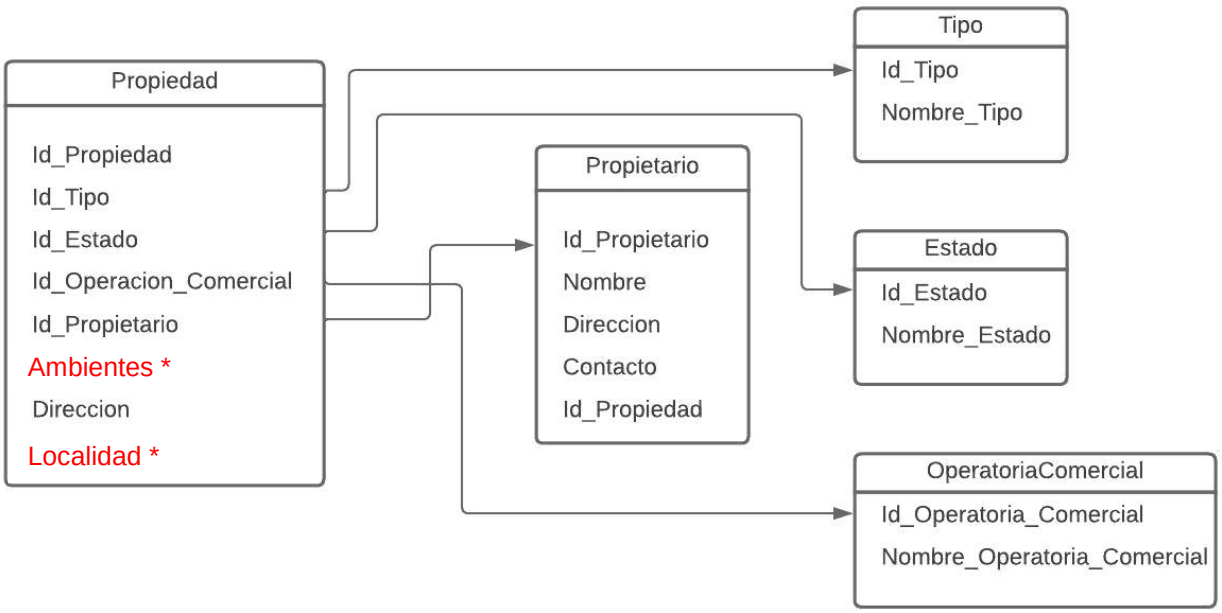
Los datos expuestos por la inmobiliaria fueron trabajados y resultó el siguiente diagrama, que nos servirá para la creación de la base de datos:

Modificación del diagrama:

Luego de analizar el diagrama, las tablas y sus campos, y teniendo en cuenta que la creación de la base de datos es de manera local, decidimos cambiar dos campos de la tabla Propiedad, Nombre y Contacto no tienen sentido en ésta tabla, los mismos se encuentran en la tabla Propietario (una propiedad no tiene nombre ni contacto, un propietario sí).

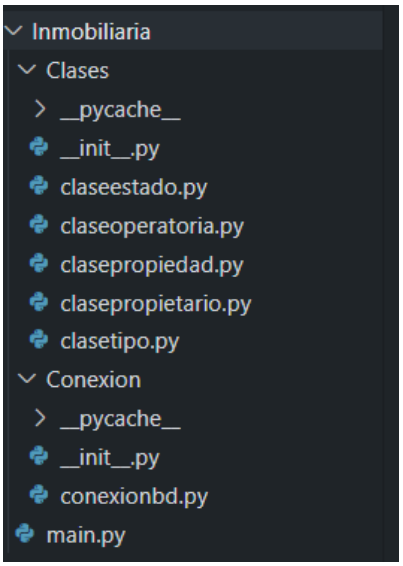
Cambiamos dichos campos y su tipo por Ambientes (cantidad de ambientes de la propiedad) y Localidad (ubicación de la propiedad) respectivamente, como se aprecia en el diagrama siguiente marcado en rojo y un asterisco (*).

Por lo tanto, para cada tabla se implementó una clase con sus atributos y métodos propios.

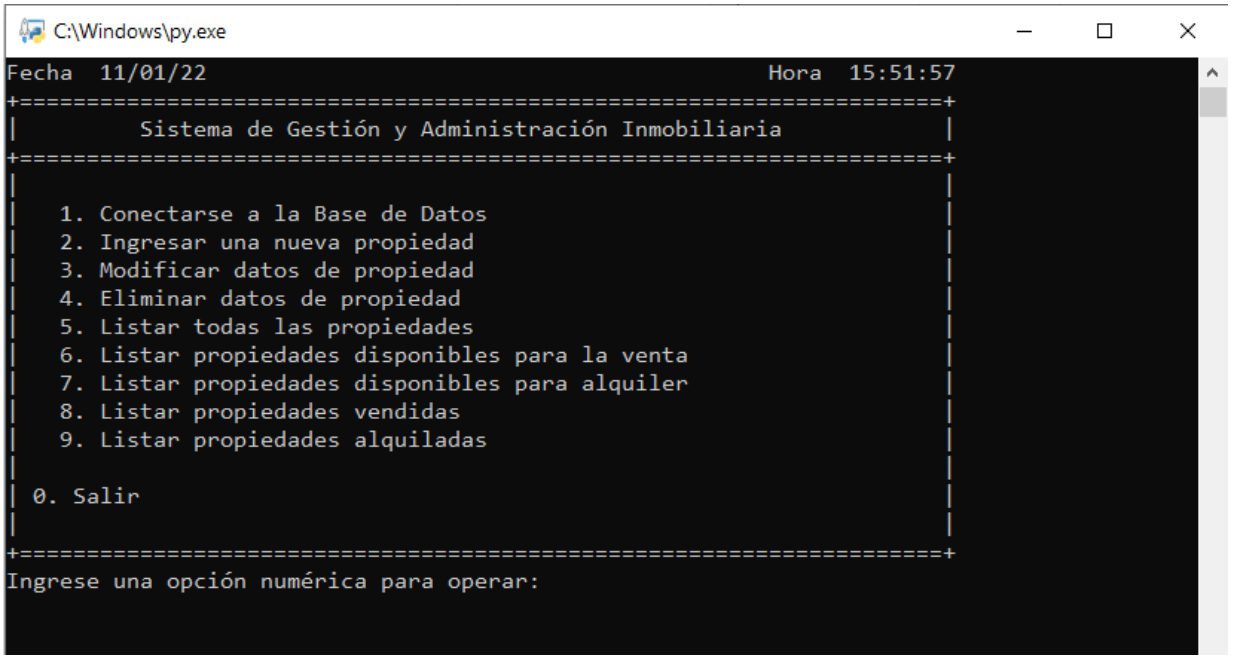


Deberán realizar un ABM que permita:

Estructura y ABM en Python:



La imagen arriba representa la estructura de la implementación en Python, la carpeta principal “Inmobiliaria” (paquete), que contiene la carpeta “Clases” (módulo) donde se encuentran las entidades del diagrama implementadas en clases con sus atributos y métodos respectivos, la carpeta “Conexión” (módulo) donde se encuentra la implementación para conectar con la base de datos desarrollada como una clase utilizando el módulo *mysql.connector.* Y por último, el archivo “main.py” (será nuestro ejecutable) donde se implementa el menú con el que interactúa el usuario entre las diferentes opciones solicitadas en la consigna, como se muestra en la figura siguiente:



★ *Ingresar una propiedad.*

Opción 2 del menú para cargar una propiedad, se irán solicitando al usuario primero los datos del propietario, luego de la propiedad y por ultimo las opciones (tipo, estado, operatoria comercial), y se mostrará un mensaje que la carga fue exitosa, de lo contrario, en casa de un error se mostrará un mensaje y se volverá al Menú principal.

★ *Modificar una propiedad.*

Opción 3 del menú para modificar una propiedad, primero se muestra una tabla con todas las propiedades disponibles en la base de datos, y se solicita al usuario ingrese el ID que identifica la propiedad, si la opción elegida no es correcta se muestra un mensaje y las opciones a seleccionar, de lo contrario, se muestran los datos de la propiedad a modificar y se van solicitando los datos nuevos al usuario. Para modificar el propietario asociado se le consulta al usuario si también desea modificarlo. Por último se muestra un mensaje que la carga fue exitosa, de lo contrario, en casa de un error se mostrará un mensaje y se volverá al Menú principal.

★ *Eliminar una propiedad.*

Opción 4 del menú para eliminar una propiedad, primero se muestra una tabla con todas las propiedades disponibles en la base de datos, y se solicita al usuario ingrese el ID que identifica la propiedad, si la opción elegida no es correcta se muestra un mensaje y las opciones a seleccionar, de lo contrario, se muestra un mensaje que la propiedad fue eliminada con éxito.

Además, se solicita la creación de los siguientes reportes:

★ *Listado de propiedades TOTALES, sin distinción de estados.*

Opción 5 del menú, se muestran en una tabla todas las propiedades con sus datos correspondientes.

★ *Listado de propiedades DISPONIBLES para la venta.*

Opción 6 del menú, se muestran en una tabla las propiedades sólo disponibles para la venta con sus datos correspondientes.

★ *Listado de propiedades DISPONIBLES para alquiler.*

Opción 7 del menú, se muestran en una tabla las propiedades sólo disponibles para alquiler con sus datos correspondientes.

★ *Listado de propiedades vendidas.*

Opción 8 del menú, se muestran en una tabla las propiedades vendidas con sus datos correspondientes.

★ *Listado de propiedades alquiladas.*

Opción 9 del menú, se muestran en una tabla las propiedades alquiladas con sus datos correspondientes.

También se necesita que se conecte la Base de Datos creada, con el ABM desarrollado para poder persistir los datos.

Opción 1 del menú, para conectar a la base de datos, se muestra un mensaje si la conexión fue exitosa con fecha y hora, de lo contrario un mensaje de error. Por otra parte, si se intenta ingresar a algunas de las restantes opciones sin estar la BD conectada, mostrará un mensaje al usuario para que primero establezca la conexión.

Nota 1: Opción 0, para salir del programa. Aquí además se realizará la desconexión a la base de datos mostrando un mensaje correspondiente. La conexión se realiza una vez en la opción 1, y mientras el usuario interactúe con el programa (lo tenga abierto) dicha conexión permanecerá (no se cerrará ni abrirá en ninguna sentencia interna, solo se chequeará si esta conectada) para facilitar el uso y evitar errores por una conexión cerrada.

Nota 2 IMPORTANTE: **para presentar de manera correcta, ordenada y estética la información de las propiedades en pantalla utilizamos e importamos el módulo *Tabulate* disponible en las librerías de Python. Por lo tanto, debe ser instalado previamente con el comando: *pip install tabulate***