



El programa que se describe, pretende la gestión de una librería y corresponde al primer entregable de la diplomatura en Python dictada durante diciembre de 2023 por el Ing. Juan Barreto.

Este programa utiliza Tkinter para crear una interfaz gráfica básica y SQLite para trabajar con una base de datos. A continuación, se presenta un resumen de lo que hace:

1. **Importaciones:**

- Importa los módulos necesarios de Tkinter para la interfaz gráfica y de SQLite para la base de datos.

2. **Funciones:**

- `limpiar_treeview(tree)`: Borra todos los elementos de un widget Treeview, probablemente utilizado para limpiar la visualización de datos.
- `cargar_libros(tree)`: Carga información de libros desde la base de datos a un widget Treeview, un componente de visualización tabular en Tkinter.
- `conexion()`: Establece la conexión con la base de datos SQLite llamada "base_librero.db".
- Funciones para crear tablas en la base de datos (`crear_tabla_libros()`, `crear_tabla_categorias()`, `crear_tabla_autores()`, `crear_tabla_editorial()`).
- Funciones para agregar datos predefinidos a las tablas (`agregar_generos_literarios()`, `agregar_autores()`, `agregar_editoriales()`).

3. **Manejo de Errores:**

- Utiliza bloques `try-except` para controlar errores en la creación de tablas y la inserción de datos predefinidos en la base de datos. Imprime mensajes de error si ocurre algún problema.

Los siguientes son los CRUD de los diferentes datos que maneja el programa.

Funciones CRUD de Categorías:

1. `buscar_categoria(nombre_categoria)`: Busca una categoría en la base de datos según el nombre proporcionado.
2. `buscar_categorias()`: Recupera todas las categorías existentes en orden alfabético.

3. `guardar_categoria(nombre_categoria)`: Agrega una nueva categoría a la base de datos.
4. `guardar_mod_categoria(nombre, categoriamod)`: Actualiza el nombre de una categoría existente en la base de datos.
5. `nueva_categoria()`: Abre una ventana para agregar una nueva categoría. Valida la entrada de datos y muestra mensajes de error si es necesario.
6. `modificar_categoria()`: Abre una ventana para modificar el nombre de una categoría existente. Valida la entrada de datos y muestra mensajes de error si es necesario.
7. `borrar_categoria(nombre)`: Elimina una categoría de la base de datos.
8. `eliminar_categoria()`: Abre una ventana de confirmación para eliminar una categoría seleccionada. Muestra mensajes de éxito o error después de la operación.

Funciones CRUD de Editoriales:

1. `buscar_editorial(nombre_editorial)`: Busca una editorial en la base de datos según el nombre proporcionado.
2. `buscar_editoriales()`: Recupera todas las editoriales existentes en orden alfabético.
3. `guardar_editorial(nombre_editorial)`: Agrega una nueva editorial a la base de datos.
4. `guardar_mod_editorial(nombre, editorialmod)`: Actualiza el nombre de una editorial existente en la base de datos.
5. `nueva_editorial()`: Abre una ventana para agregar una nueva editorial. Valida la entrada de datos y muestra mensajes de error si es necesario.
6. `modificar_editorial()`: Abre una ventana para modificar el nombre de una editorial existente. Valida la entrada de datos y muestra mensajes de error si es necesario.
7. `borrar_editorial(nombre)`: Elimina una editorial de la base de datos.
8. `eliminar_editorial()`: Abre una ventana de confirmación para eliminar una editorial seleccionada. Muestra mensajes de éxito o error después de la operación.

Funciones CRUD de Autores:

1. `buscar_autor(nombre_autor)`: Busca un autor en la base de datos según el nombre proporcionado.
2. `buscar_autores()`: Recupera todos los autores existentes en orden alfabético.
3. `guardar_autor(nombre_autor)`: Agrega un nuevo autor a la base de datos.
4. `guardar_mod_autor(nombre, autormod)`: Actualiza el nombre de un autor existente en la base de datos.
5. `nuevo_autor()`: Abre una ventana para agregar un nuevo autor. Valida la entrada de datos y muestra mensajes de error si es necesario.
6. `modificar_autor()`: Abre una ventana para modificar el nombre de un autor existente. Valida la entrada de datos y muestra mensajes de error si es necesario.
7. `borrar_autor(nombre)`: Elimina un autor de la base de datos.
8. `eliminar_autor()`: Abre una ventana de confirmación para eliminar un autor seleccionado. Muestra mensajes de éxito o error después de la operación.

Funciones CRUD de Libros:

- `buscar_libro(nombre_libro)`: Busca un libro por su título en la base de datos y devuelve los detalles si se encuentra.
- `buscar_libros()`: Obtiene todos los libros de la base de datos y los formatea para mostrarlos en una lista.
- `obtener_id_autor(autor), obtener_id_editorial(editorial), obtener_id_categoria(categoria)`: Estas funciones obtienen el ID correspondiente para el autor, la editorial y la categoría de un libro.
- `guardar_libro(nombre_libro, autor, editorial, anio, genero, comentario)`: Guarda un nuevo libro en la base de datos con todos sus detalles.
- `guardar_mod_libro(nombre, libromod)`: Actualiza un libro existente en la base de datos.
- `nuevo_libro()`: Abre una ventana para agregar un nuevo libro con información como título, autor, género, editorial, año y comentarios.
- `eliminar_libro()`: Abre una ventana para eliminar un libro seleccionado en el treeview.
- `eliminar_libro_db(libro_id)`: Elimina un libro de la base de datos según su ID.

Luego, en la mainloop (`Tkinter`), se crea una aplicación con un menú que tiene opciones para libros, categorías, editoriales y autores. Además, utiliza un `Treeview` para mostrar los libros con sus detalles y permite operaciones CRUD sobre ellos.



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**
Secretaría de Cultura y Extensión Universitaria

PRIMER ENTREGABLE