



ANOTACIONES DEL CÓDIGO

1) Importaciones:

- a) **tkinter**: Para la interfaz gráfica.
- b) **messagebox, ttk**: Componentes adicionales de tkinter para mensajes y el Treeview.
- c) **mariadb (o mysql.connector)**: El conector para la base de datos.
- d) **bcrypt**: Para el cifrado seguro de contraseñas.
- e) **DateEntry de tkcalendar**: Un widget para seleccionar fechas de forma interactiva.
- f) **datetime**: Para manejar objetos de fecha.

2) Configuración de Base de Datos (DB_CONFIG):

- a) Un diccionario que contiene los parámetros de conexión a tu base de datos MariaDB.

3) Funciones de Base de Datos:

- a) **get_db_connection()**: Establece y devuelve una conexión a la base de datos. Maneja errores de conexión.
- b) **create_table()**: Crea la tabla users si no existe. La tabla tiene los campos id (autoincremental), nombre, fecha_nacimiento (tipo DATE) y password_hash (para la contraseña cifrada).
- c) **hash_password(password)**: Cifra la contraseña utilizando bcrypt.hashpw(). bcrypt genera automáticamente un "salt" y lo incluye en el hash, lo que es crucial para la seguridad.
- d) **verify_password(password, hashed_password)**: Verifica si una contraseña plana coincide con su versión cifrada almacenada.
- e) **insert_user(nombre, fecha_nacimiento, password)**: Inserta un nuevo usuario en la base de datos. La contraseña se cifra antes de ser almacenada.
- f) **get_users()**: Recupera todos los usuarios de la tabla (sin la contraseña hash para evitar exponerla).



- g) `get_user_by_id(user_id)`: Recupera un usuario específico por su ID.
 - h) `update_user(user_id, nombre, fecha_nacimiento, password=None)`: Actualiza los datos de un usuario. Si se proporciona una nueva contraseña, esta también se cifra y actualiza.
 - i) `delete_user(user_id)`: Elimina un usuario por su ID.
- 4) Clase `UserApp` (Interfaz Tkinter):
- a) `__init__(self, master)`: Inicializa la aplicación, configura la ventana principal y llama a `create_widgets()` y `load_users()`.
 - b) `create_widgets(self)`:
 - i) Crea un `LabelFrame` para agrupar los campos de entrada y botones.
 - ii) Campos de Entrada: `tk.Entry` para ID (solo lectura/visualización para edición), nombre y contraseña. `DateEntry` de `tkcalendar` para la fecha de nacimiento, lo que proporciona un calendario desplegable.
 - iii) La entrada de la contraseña (`self.password_entry`) usa `show="*"` para ocultar los caracteres mientras se escribe.
 - iv) Botones: "Crear Usuario", "Actualizar Usuario", "Eliminar Usuario" y "Limpiar Campos".
 - v) `Treeview`: Un widget de tabla para mostrar los usuarios.
 - (1) `show="headings"`: Muestra solo los encabezados de columna.
 - (2) `heading()`: Define el texto para los encabezados de las columnas.
 - (3) `column()`: Configura el ancho y la alineación de las columnas.
 - (4) `bind("<ButtonRelease-1>", self.on_tree_select)`: Asocia un evento para que, al hacer clic en una fila del Treeview, los datos se carguen en los campos de entrada.
- c) `load_users(self)`: Limpia el `Treeview` y lo vuelve a llenar con los datos actuales de la base de datos.



- d) `add_user(self)`: Obtiene los datos de los campos, valida que no estén vacíos y llama a `insert_user()`. Luego recarga la lista de usuarios y limpia los campos.
 - e) `update_selected_user(self)`: Obtiene el ID del usuario seleccionado en el Treeview, los nuevos datos de los campos de entrada, y llama a `update_user()`. Recarga y limpia.
 - f) `delete_selected_user(self)`: Obtiene el ID del usuario seleccionado, pide confirmación y llama a `delete_user()`. Recarga y limpia.
 - g) `on_tree_select(self, event)`: Esta función se ejecuta cuando se selecciona una fila en el Treeview. Carga los datos de la fila seleccionada en los campos de entrada para su edición. Nota: No se carga la contraseña por seguridad. El campo de ID se deshabilita para evitar modificaciones accidentales.
 - h) `clear_fields(self)`: Limpia todos los campos de entrada y restablece la fecha a la actual. Vuelve a habilitar el campo ID.
- 5) `if __name__ == "__main__":`
- a) Este bloque se ejecuta cuando el script se inicia directamente.
 - b) `create_table()`: Asegura que la tabla users exista en la base de datos antes de iniciar la aplicación.
 - c) `root = tk.Tk()`: Crea la ventana principal de tkinter.
 - d) `app = UserApp(root)`: Crea una instancia de nuestra aplicación.
 - e) `root.mainloop()`: Inicia el bucle de eventos de tkinter, haciendo que la ventana sea interactiva.

Darío Fernando Gómez Z.
Instructor - CEAI