



## 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