**MANUAL DE LA APLICACIÓN**

**DOCUMENTACION TECNICA**

**INDICE**

1. Descripción general.
2. Pantalla principal – Sus componentes.
3. Manual de uso.
   1. Alta de un registro.
   2. Baja de un registro.
   3. Modificación de un registro.
   4. Limpieza de las cajas de entrada.
   5. Filtro por nombre de obra.
   6. Selección en listado.
4. Módulos de la aplicación.
5. Detalle de las funciones del modelo.
   1. Funciones para el manejo de la base de datos.
   2. Funciones de ABM.
   3. Funciones de consulta y treeview.
   4. Función de cierre de la aplicación.
   5. Funciones de manipulación de datos.
   6. Declaración de la variable “info”.
6. Vista y control.
   1. Raíz de la ventana.
   2. Frame de menú.
   3. Frame de datos.
   4. Frame de treeview.
   5. Label de status.

**1. Descripción general**

Se trata de una aplicación para el manejo de una nómina de empleados.

La visualización de la misma será con TKinter, que por medio de sus widgets podremos hacer las altas, bajas, modificaciones y consultas (CRUD). La información a manipular será guardada en una base de datos a través de SQLite3.

**2. Pantalla principal – Sus componentes**

Captura con referencia en números y letras.

A Área de menú de comandos.

A.1 Botón para dar de alta al registro.

A.2 Botón para la baja del registro previamente seleccionado.

A.3 Botón para la modificación del registro previamente seleccionado.

A.4 Botón para limpiar los campos de ingreso de datos.

A.5 Botón para salir de la aplicación.

B Área para manejo de datos.

B.1 DNI (será controlado el tipo y longitud de dato) dato obligatorio.

B.2 CUIL (será controlado el tipo y longitud de dato) dato obligatorio.

B.3 Nombres de la persona, dato obligatorio.

B.4 Apellidos de la persona, dato obligatorio.

B.5 Domicilio en el cual reside actualmente.

B.6 Fecha de nacimiento, se puede ingresar directamente o con un almanaque.

B.7 Fecha de alta, por defecto aparece la fecha actual, pero se puede modificar con la misma metodología que el punto anterior.

B.8 Obra a la cual fue asignado a trabajar.

b.9 El jornal que percibe por día de trabajo.

B.10 Botón de búsqueda.

C Área Listado

C.1 Ingresar nombre de obra por la que se quiere filtrar el listado.

C.2 Botón para aplicar el filtro.

C.3 Listado de todas las obras o el contenido filtrado.

D Área de notificación o status de las actividades.

**3. Manual de uso**

3.a. Alta de un registro:

Para poder dar de alta a un empleado es necesario completar todos los campos, en especial los obligatorios (DNI (B.1), CUIL (B.2), Nombres (B.3) y Apellidos (B.4)) Para el DNI es necesario que sean solo números y una cantidad mínima de dígitos de 7 y máxima de 8 sin colocar puntos. Para CUIL son 11 dígitos y sin colocar los guiones.

Para las fechas se pueden ingresar directamente o a través de los almanaques que se despliegan.

Una vez completados todos los datos se debe dar al botón de ALTA (A.1) y será agregado al listado de abajo (C.3) y grabado en la base de datos. En caso de errores será alertado con una ventana emergente y en todos los casos se le informará sobre el proceso en el área de notificación.

3.b. Baja de un registro:

Hay dos caminos distintos para dar de baja a algún registro.

* Una manera es seleccionando el mismo desde el listado en la parte inferior y luego se completarán todos los campos (B) con los datos correspondientes a ese registro.
* La otra alternativa es ingresar el número de DNI en el campo (B.1) con dicho nombre y presionar el botón de Buscar (B.10). Este traerá a los campos los datos correspondientes de ese registro. En casa que no exista en la base de datos ese número de DNI se dará una advertencia.

En ambos métodos luego se debe presionar el botón de BAJA (A.2) se consultará si está seguro de ejecutar la sentencia y con la afirmación se borrará la información en el listado de abajo y de la base de datos. En caso de errores será alertado con una ventana emergente y en todos los casos se le informará sobre el proceso en el área de notificación.

3.c. Modificación de un registro:

Los procedimientos para la modificación de algún dato correspondiente a un registro son:

* Una manera es seleccionando el mismo desde el listado en la parte inferior y luego se completarán todos los campos (B) con los datos correspondientes a ese registro.
* La otra alternativa es ingresar el número de DNI en el campo (B.1) con dicho nombre y presionar el botón de Buscar (B.10). Este traerá a los campos los datos correspondientes de ese registro. En casa que no exista en la base de datos ese número de DNI se dará una advertencia.

En ambos métodos se bloquearán los campos de DNI (B.1) y CUIL (B.2), los cuales no podrán ser modificados. Luego se debe presionar el botón de MODIFICAR (A.3) se consultará si está seguro de ejecutar la sentencia y con la afirmación se actualizará la información del listado de abajo y de la base de datos. En caso de errores será alertado con una ventana emergente y en todos los casos se le informará sobre el proceso en el área de notificación.

3.d. Limpieza de las cajas de entrada:

Luego de hacer una consulta de la información de algún registro se puede hacer el borrado de todos los campos (B) presionando en el botón de LIMPIEZA (A.4). Al vaciar los campos se volverán a habilitar las cajas de DNI (B.1) y CUIL (B.2) para poder ingresar datos.

3.e. Filtro por nombre de obra:

En el área del listado se propone la posibilidad de hacer un filtrado del mismo por el nombre de la obra asignada, y de esa manera tener la nómina correspondiente a cada obra.

Se debe completar el campo con el nombre de la obra (C.1) y luego presionar el botón de Filtrar (C.2).

Si se deja vacío el campo de filtrado y se presiona el botón Filtrar se volverá a mostrar todos los registros. En caso de errores será alertado con una ventana emergente y en todos los casos se le informará sobre el proceso en el área de notificación.

3.f. Selección en listado:

Ya sea en el listado (C.3) completo o filtrado, al seleccionar un registro automáticamente se completarán todos los datos disponibles en el área de carga de datos. Para a posteriori poder ejecutar una BAJA o MODIFICACION.

**Detallado del código de la aplicación.**

**4. Módulos de la aplicación**

Se cargan los módulos de “tkinter” para el manejo de la interfaz gráfica, “Tkinter.messagebox” para las ventanas emergentes de alertas o decisiones, “Tkcalendar” para facilitar la carga de las fechas de nacimiento y alta, “os” para la obtención de las rutas dependiendo del sistema en el que se ejecute, “sqlite3” para el manejo de la base de datos y “re” para poder verificar la información con expresiones regulares.

**5. Detalle de las funciones del modelo**

5.a Funciones para manejo de la base de datos

*5.a.1 Creación de la base de datos.*

En primer lugar, se busca la ruta en la que se está ejecutando la aplicación y luego se concatena con la separación entre carpetas correspondiente al sistema sobre el cual se esta ejecutando y con el nombre de la carpeta donde se guardara la base de datos.

Se crea la base de datos o, si esta creada, se establece la conexión con la base dándole el nombre “nomina\_database.db”

*5.a.2 Creación de la tabla.*

Se genera el cursor para apuntar a la base de datos, se asigna a la variable sql la instrucción de creación de la tabla, ejecuta el cursor y se cierra con el commit.

*5.a.3 Consulta a la tabla de la base de datos.*

Creo una función para las consultas a la base de datos, ya que es una tarea que se repite varias veces en el desarrollo de la aplicación. Recibe como parámetro un string con la instrucción de sqlite3 y retorna una lista con lo encontrado en la base se acuerdo a los parámetros de búsqueda establecidos por cada funcion.

*5.a.4 Modificación de la tabla de la base de datos.*

Es una función que es compartida entre el alta, la baja y las modificaciones. Por ello recibe un string con la instrucción de sqlite3 y una lista con los datos a grabar, modificar o eliminar en la tabla.

5.b Funciones de ABM

*5.b.1 Alta de registro.*

La información que recibe es una lista con todos los campos que se cargaron en el formulario de interfaz gráfica. Lo primero que se hace es comprobar que no estén vacíos los campos obligatorios. Luego se establecen los patrones de expresiones regulares para comprobar que el campo dni (B.1) solo tenga números entre 7 u 8 dígitos y que el campo de cuil (B.2) solo tenga números de 11 dígitos. Con esto correcto se establece la instrucción de sql que será enviada a la función de modificación de la tabla (5.a.4) junto con la lista de datos. Se vacían los campos del formulario (5.e.2), se informa el estado del proceso en la etiqueta al pie de la ventana y se actualiza el treeview (5.c.3). En el caso que no se carguen todos los datos obligatorios, se ingresen incorrectamente el dni y/o cuil o que el dni ya exista en la base de datos; se advierte al usuario y se especifica en la etiqueta al pie de la ventana.

*5.b.2 Baja de registro.*

Esta función recibe una lista con la información del registro a eliminar y el treeview. Se verifica que el campo de dni (B.1) no este vacío, se establece en la variable sql la instrucción de sqlite3 y se consulta al usuario si esta seguro de la eliminación. En caso afirmativo se envía a la función de modificación de la tabla (5.a.4) la instrucción de sql y el dato del dni en una tupla. A posteriori se actualiza el treeview (5.c.3), se vacían los campos del formulario (5.e.2) en la interfaz grafica y se notifica en la etiqueta al pie de la ventana del éxito de la operación. En el caso que el dni (B.1) este vacío o que se cancele la eliminación se alerta al usuario y se informa en la etiqueta al pie de la ventana.

*5.b.3 Modificación de un registro.*

Al llamar a esta función se le envía la lista con los datos de los campos consultados y el treeview. Se verifica que el campo dni (B.1) no este vacío y de no estarlo se carga la instrucción de sqlite3 en la variable sql y se agrega el valor del dni al final de la lista data. Se consulta al usuario si esta de acuerdo con continuar el proceso y se de ser afirmativo a través de un bucle se eliminan los datos de dni y cuil, que no se pueden modificar, de la lista; así de esta manera queda ordenada para enviar a la función de modificación de la tabla (5.a.4) junto con la instrucción de sql. Al regreso se vacían los campos del formulario (5.e.2), se actualiza el treeview (5.c.3) y se notifica al usuario, en la etiqueta al pie de la ventana, que la operación se concretó. Si el campo dni (B.1) está vacío o se cancela la modificación se alerta al usuario y se informa en la etiqueta al pie de la ventana.

5.c. Funciones de consulta y treeview.

*5.c.1 Consulta a través del treeview.*

Al seleccionar una fila del listado (C.3) se toma el dato del id del treeview y se le establece a la variable sql un string con la instrucción de sqlite3 para consultar ese id a la base de datos. Para ello se envia la variable sql y el id a la función de consulta de la base (5.a.3) retornándonos una lista con toda la información para enviar a la función de seteo de los campos del formulario (5.e.2).

*5.c.2 Búsqueda desde el botón del formulario.*

Esta función recibe un string que funcionara como índice para poder hacer la búsqueda de la información en la base de datos. Para ello se establece en sql la instrucción de sqlite3 para luego enviarla junto con el índice a la función de consulta de la base (5.a.3), recibiendo una lista con todos los campos a cargar a través de la función de seteo de los campos (5.e.2). En caso que no se reciba información se le alerta al usuario que no esta registrado ese dni. En ambos casos se le informa de lo acontecido en la etiqueta al pie de la ventana.

*5.c.3 Actualización de la información en el treeview.*

Esta función puede ser llamada desde otras funciones o desde el botón de filtrar por obra, por eso recibe el tree y un parámetro que establece desde donde surge la consulta y así poder establecer correctamente la instrucción de sql en la variable con el mismo nombre. Se envía la instrucción a la función de consulta de la base de datos (5.a.3), retornándonos una lista con todos los campos que se guarda en data\_list. Se borran todos los datos que hay en el treeview a través de un bucle for y con la misma metodología se cargan los datos recuperados de la base de datos. Para finalizar de vacía la entrada del filtro.

5.d Función de cierre de la aplicación.

Al presionar el botón de salida (A.5) se dispara esta función que consulta al usuario si esta seguro de continuar, y de ser así se cierra la aplicación.

5.e Funciones de manipulación de datos.

*5.e.1 Crear lista.*

Diseñe esta función para crear una lista con todos los campos del formulario (B) de la interfaz gráfica y sea más sencilla la manipulación de los datos entre función y función. En los datos como nombres, apellidos, dirección, obra asignada y art o seguro; se le aplica un método “.capitalize” para estandarizar el dato y que sea más efectiva la coincidencia a la hora de efectuar una búsqueda.

*5.e.2 Seteo de los entrys del formulario.*

Creo esta función debido a que esta tarea se usa en varias funciones y evita tener que repetir estas líneas varias veces en el código. En ella se puede cargar datos o vaciar los entrys. A su vez dependiendo si está vacío o con información el campo dni lo habilita o deshabilita respectivamente.

5.f Declaración de la variable info.

Se le da un texto multilinea para mostrar en la ventana que se abre a través del menubar de “Acerca de …”

**6. Vista y control**

6.a Raíz de la ventana.

Se declara la ventana principal de la aplicación determinando el título, el icono y la configuración que restringe el cambio de las dimensiones de la misma.

Para la búsqueda del icono en la carpeta “img”, uso el módulo os para establecer la ruta y el símbolo de separación de carpetas que corresponde al sistema operativo sobre el que se corre la aplicación.

Se dibuja el menú con solo la opción de “Acerca de …” que dispara una ventana de información con el texto antes declarado (5.f).

Se determina la etiqueta para el título de la aplicación y se dibuja a través del método de grid.

Declaro todas las variables que serán utilizadas en la interfaz gráfica.

La ventana principal se va a dividir en tres frames que contendrán las distintas áreas de la pantalla.

6.b Frame de menú. (A)

Este va a contener los botones para el alta (A.1), baja (A.2), modificación (A.3), limpieza (A.4) y salida (A.5) de la aplicación. Cada uno llamara a la función correspondiente a través de una función lambda que permite él envío de la información. Todos los botones son dibujados en el frame con el método de grid.

6.c Frame de datos. (B)

Aquí se dibujan con el método de grid todas las cajas de entrada de datos del formulario con sus respectivas etiquetas y el botón para poder disparar la función de búsqueda (B.10) del dni cargado.

6.d Frame treeview. (C)

Por último, en este contendrá el listado (C.3) con toda la información de la base de datos y una caja de entrada de datos (C.1) para poder hacer un filtrado del listado por el nombre de la obra asignada. Al cual le corresponde un botón (C.2) que dispara la actualización del treeview (5.c.3) filtrando la búsqueda por el dato ingresado, y si este está vacío se cancela el filtrado. Al dato cargado se le aplica un metodo “.capitalize” para que haya mejores correspondencias con los campos de la base de datos. Todos estos widgets se ubican con el método de grid.

6.e Label de status. (D)

Se inicializa la etiqueta que luego se utilizara continuamente para informar al usuario de los que va pasando en el proceso de utilización de la aplicación.

Se intenta (try:) conectar a la base de datos y crear la tabla, llamando a las funciones correspondientes en cada caso. En el caso de no poder llevar a cabo estas sentencias se hace una excepción (except:) para controlar el error. En ambos casos se configura con la leyenda correspondiente la etiqueta de status al pie de la aplicación.

Por ultimo y muy importante se efectúa el mainloop para poder mantener la ventana activa hasta salir de la aplicación.