**Ventana Login**

Se creó un módulo o archivo independiente de Python para desarrollar el código del funcionamiento de la ventana “Login”, para posteriormente llamar este archivo desde un módulo principal.

Antes de importar el archivo de la interfaz grafica creada en Qt designer para la ventana de “Login” se procede a hacer la conversión del archivo con terminación .ui a .py con la finalidad de tener un fácil acceso a los botones y sus propiedades. Si se hiciera desde la versión .ui no se autocompleta el código al llamar a estos elementos.

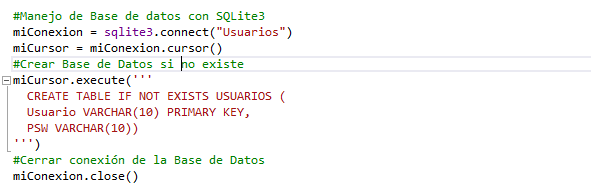


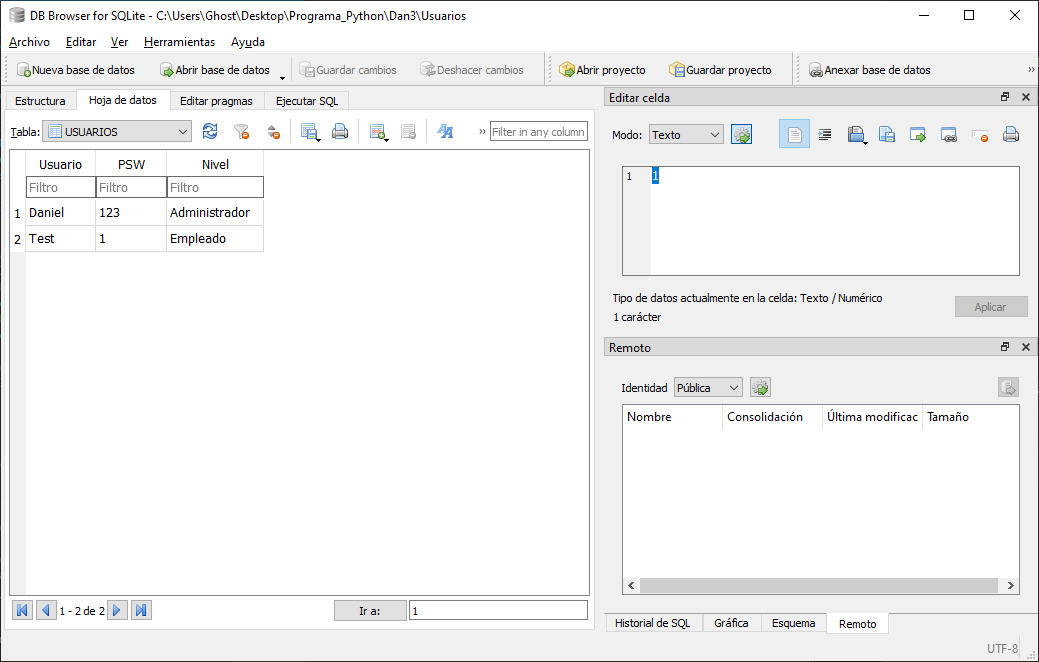
Para hacer la exportación nos ayudamos del “Simbolo del Sistema”, posicionándonos en la carpeta donde se encuentra el archivo llamado f\_Login.ui y se usa el siguiente comando:

pyuic5 f\_Login.ui -o f\_Login.py

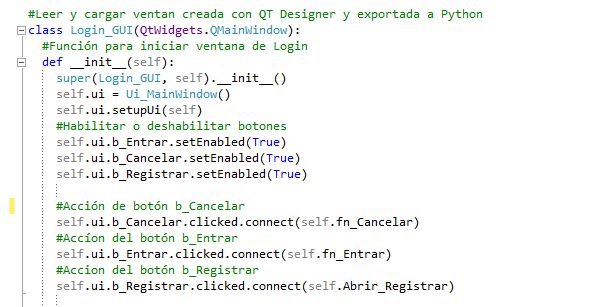
Lo siguiente que se necesita es tener nuestra base de datos creada con SQLite, para almacenar los usuarios que van a tener acceso al programa.

Se puede realizar mediante código de SQL o usando el programa DB Browser. Para nuestro caso es más practico hacerlo desde código para poder prevenir que si la base de datos no existe, se cree un archivo nuevo que tendrá las columnas necesarias, pero estará vacío.

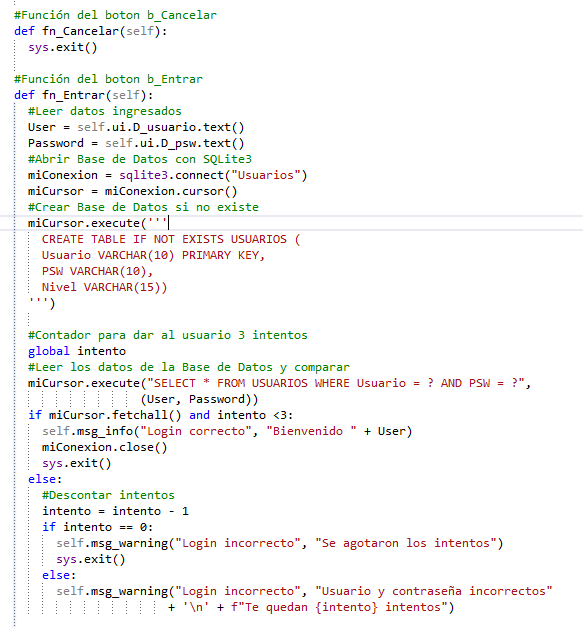




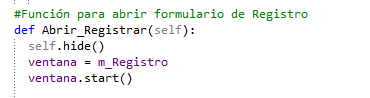
Se continua con el llamado de la ventana y se comienzan a llamar los botones para tener un estado por default al lanzar nuestra aplicación. Para las acciones al presionar los botones o interactuar con los mismos, se crean funciones para cada acción que queramos realizar. Cabe aclarar que todo el código que tiene que ver con la ventana se tiene que hacer dentro de una clase.



Aquí se detalla el código de las funciones que se activan al usar el botón “Cancelar”, el botón “Entrar”.

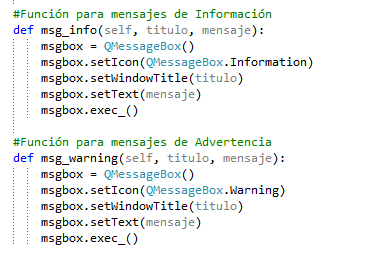


Para el caso del botón registrar se hace el llamado a la ventana creada para el registro y modificación de usuarios, que se creó igualmente con Qt designer que va a tener su propio módulo de Python llamado m\_Registro, por lo que se debe importar a nuestro módulo de m\_Login para poder llamar a esta ventana.

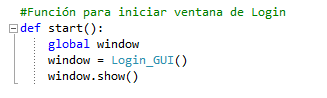


nuestra interfaz gráfica se llama f\_Registro.

Adicionalmente se crean unas funciones para poder mostrar al usuario mensajes de advertencia.

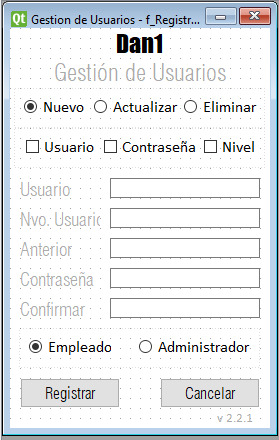


Y finalmente se crea una función para poder llamar a esta ventana de Login desde otros módulos, ya sea el modulo principal o donde se requiera. Este código debe estar fuera de la clase.



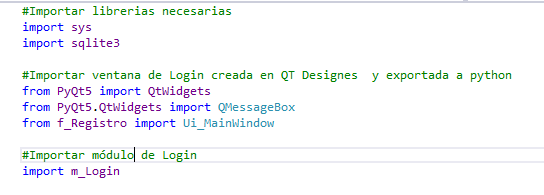
**Ventana Registro**

Se realiza el diseño usando nuevamente Qt Designer, tratando de contemplar todas las opciones necesarias para registrar, actualizar la información o eliminar un usuario.

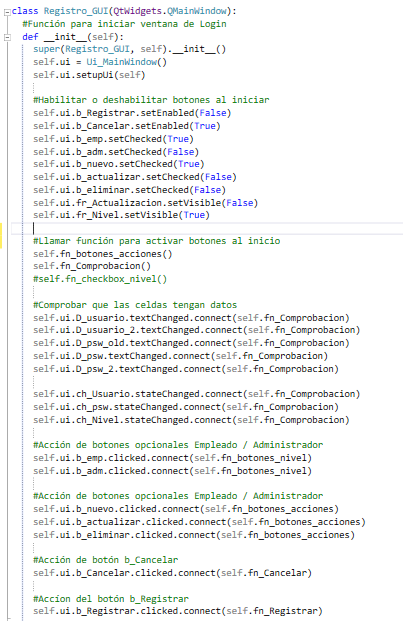


Este módulo fue más complejo debido a la cantidad de opciones para elegir que tendrá el usuario, pero sin embargo se logró hacer la programación de todas las condiciones.

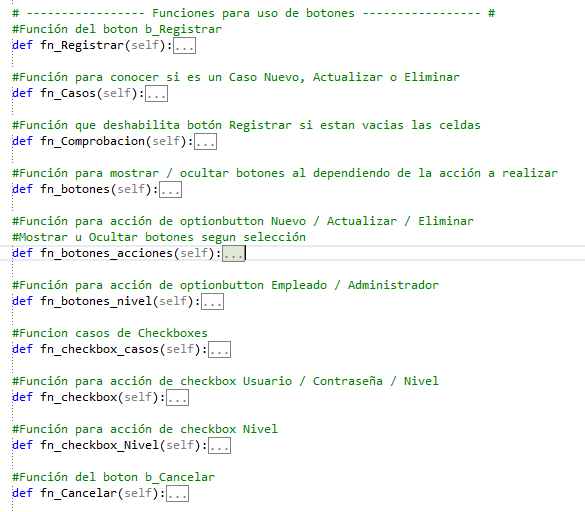
Primeramente, se hace el llamado de las librerías y módulos que se van a utilizar, así como de nuestra interfaz grafica previamente convertida a un archivo f\_Registro.py.



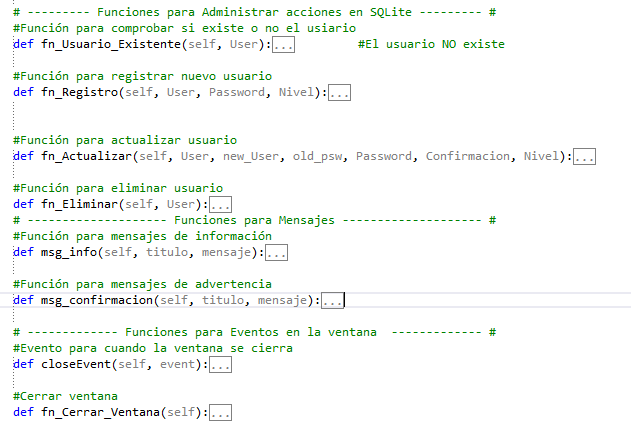
Posteriormente se inicializa nuestra ventana para cargar nuestros botones como queremos que estén al inicio de la aplicación.



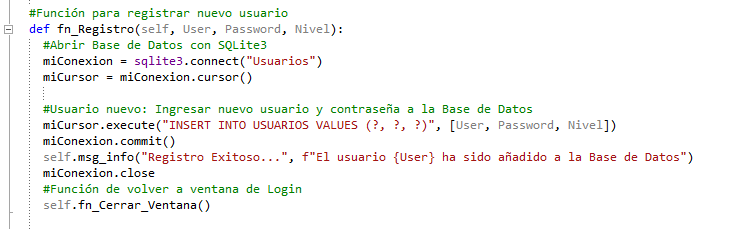
Lo siguiente seria crear las funciones para activar las acciones dependiendo de los botones a la hora de que el usuario interactúe con ellos (se muestran de manera contraída por la extensa cantidad de código).



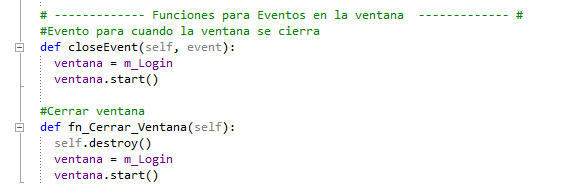
Se debe crear el código para interactuar con la base de datos, ya se apara crear, editar o eliminar, según sea el requerimiento del usuario al momento de invocar esta ventana (se muestra en forma contraída).



Se pone el ejemplo de la función que ingresa un nuevo registro de usuario en la base de datos.



Para esta ventana se requiere el uso de eventos, para mostrar al usuario la ventana de Login una vez que se cierra esta ventana, por lo que se crean las funciones necesarias para realizar estas acciones al presentarse el evento deseado.



También en esta ventana se requiere que se programe una función para poder llamar a esta ventana desde cualquier otra, por lo que se crea la función correspondiente.

