

Diseño de la aplicación:

El trabajo final esta diseñado en base a dos aplicaciones, independientes -en cuanto a su ejecución- el uno de la otra. La aplicación servidor, el cual se ejecuta mediante **server.py** está pensada para que este siempre a la escucha. Como si la misma se encontraría realmente alojada en un servidor; en una maquina distinta a la que se ejecuta el cliente.

El cliente, es la idea de la agenda que se hizo en el nivel inicial, pero hecha desde cero. La interfaz gráfica, fue modificada para que funcione desde PyQt5. Para base de datos, se implemento el ORM de peewee. Asimismo, como mencione, es el cliente. Es decir, el programa envía todos los registros cargados en pantalla, para que el CRUD, se realice en el servidor.

Aplicación Servidor:

La aplicación servidor consta de tres clases.

server.py

Como mencione, anteriormente, el módulo **server.py** es aquel con el que se inicia la aplicación servidor. También, esta pensado para que su ejecución sea independiente a la del cliente y este siempre a la escucha. Por esa razón, no se le agrego una opción de encendido / apagado del servidor a la aplicación del cliente.

El servidor es creado mediante la librería sockets y utiliza el protocolo TCP/IP.

Al recibir instrucciones de parte del cliente, el servidor va ir ejecutando diferentes funciones, como puede ser, el alta de un registro en la base de datos, como la eliminación y modificación del mismo. También puede enviar al cliente todos los registros existentes de base de datos. Finalmente, también puede cambiar el estado de la suscripción para el patrón **observador**.

Es necesario aclarar que, tanto la manipulación de base de datos, como el estado de la suscripción, es a un nivel superficial. Las funciones que manejan directamente la información se encuentran en los módulos *bbdd.py* y *observer.py*.

Por último, destaco que, por las características de la aplicación, no implemente el modelo MVC, ya que creo que no es lo óptimo para la misma.

bbdd.py

El modulo *bbdd.py* contiene todas las funciones para manejar la base de datos. La creación de la misma, se va a realizar mediante el **ORM peewee**. En base a él, se realizarán las funciones características de un CRUD con la información que se reciba desde el servidor.

observer.py

El módulo **observer** verificará si la suscripción esta activa mediante la lectura del archivo "**subscription.txt**", el cual tendrá un valor **True** por defecto. Si el valor es verdadero, enviara al cliente la confirmación de la acción realizada para que la guarde en un log de eventos.

Agenda:

La aplicación agenda consta de cinco módulos.

controlador.py

Es el modulo principal, el que inicia la aplicación (MVC).

vista.py:

Es la vista del patrón MVC. Realizado en PyQt5. Es la interfaz grafica que permite al usuario visualizar los datos que existen en la base de datos del servidor.

modelo.py

El modelo (MVC) se encarga de recolectar la información necesaria para el CRUD en la base de datos. Para eso, también consta con funciones de validación de datos para verificar que la información en pantalla cumple con los requisitos necesarios.

Toda la información recolectada es enviada al modulo cliente.py para ser manejada por el servidor.

cliente.py

Este modulo se encarga de enviar la información al servidor para que en el mismo se ejecute la acción requerida por el cliente. Las acciones pueden ser desde el alta/baja/modificación de registros. La lectura de la base de datos, para después ser visualizada en el treeview de la vista. Como la modificación del estado de la suscripción.

subscriber.py

El modulo subscriber.py guarda un log de eventos en una base de datos (ORM peewee).