

# DOCUMENTACIÓN PYTHON

El programa está dividido en 5 archivos de código y uno de configuración.

El módulo **communication** se encarga de la comunicación entre Python y el servidor Java. Intenta conectarse al servidor con un timeout de 10 segundos, si no lo consigue, el programa se iniciara igualmente, pero avisando de que no está conectado al servidor. Podrá conectarse cuando el servidor esté funcionando. Avisa al servidor cuando tiene que mandar un email, y manda un String con los datos necesarios para hacer la query en la DDBB. Utiliza la librería socket.

El módulo **stream** se encarga de controlar la PiCamera y de hacer el streaming. Principalmente, utiliza las librerías picamera, socketserver, threading y server.

En este módulo están las variables que configuran el stream y la grabación. El streaming siempre está funcionando en el thread correspondiente, y controla la cámara a través del *splitter\_port=1* (puerto 1). Las grabaciones se controlan con los métodos *record()* y *stopRecording()*. El método *record()* crea el nombre del archivo, empieza a grabar, y da el aviso al servidor (con el método del archivo **communication**) para que mande el email. Y finalmente, el método *stopRecording()* para la grabación, y avisa al servidor para que la guarde en la DDBB.

El módulo **main** es donde están los threads principales. Está compuesto por 6 tipos de thread, gui.refresh, gui.checkConnection, movimiento, temporizador, main y stream. El thread movimiento se encarga de controlar el sensor de infrarrojo. Los threads temporizador se encargan de controlar la duración de las grabaciones, para registrarlas en la DDBB, y para controlar el tamaño de los archivos con un límite ampliable de 30 segundos. Se crea uno con cada grabación, y se limpian automáticamente al llegar a 10 threads usados. El thread stream se encarga de controlar el streaming y las grabaciones. El thread gui.refresh se encarga de actualizar los datos en la parte gráfica del programa. El thread gui.checkConnection se encarga de comprobar periódicamente que

la conexión con el servidor es correcta. El thread main se encarga de conectar el thread stream, el thread movimiento, y el thread temporizador.

El módulo **gui** se encarga de la parte gráfica del programa. Muestra visualmente y en tiempo real, el estado de la conexión con el servidor, el estado del streaming, y el estado de la grabación, así como la duración de la misma. También se encarga de aplicar la configuración del usuario, y de la persistencia de datos, mediante la librería [pickle](#).

El módulo **exec** contiene el flujo principal de ejecución. Llama al método [main.init\(\)](#) que a su vez se encarga de iniciar todos los threads y variables del módulo main. También contiene el thread principal, [stream.window.mainloop\(\)](#), que inicia la parte gráfica.

El archivo **config.pickle** guarda la configuración del usuario.