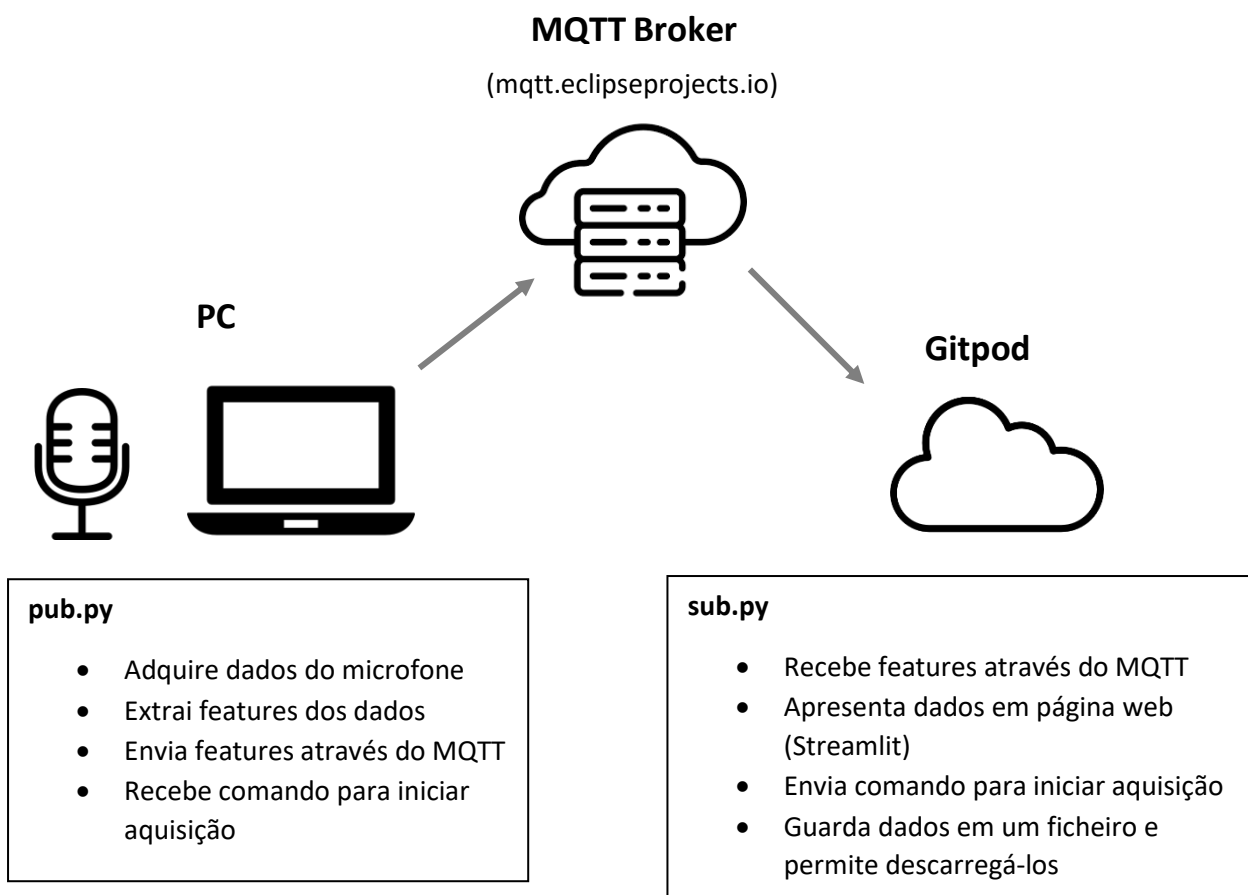


# Cloud Logger de Instrumentação

## AAIB - Projeto alternativo

Professores: Hugo Gamboa e Pedro Vieira

Aluno: Gustavo Schliemann – 56332



### pub.py

Ficheiro executado localmente que envia os dados a serem mostrados no Streamlit. Para a aquisição, foi utilizado o *sounddevice* com uma frequência de aquisição de 22050 Hz e duração de 5 segundos. As features extraídas foram os coeficientes de Fourier do sinal da gravação, com a utilização da função *librosa.stft()* e, a partir dela, a potência em relação ao valor máximo do sinal, em dB, com *librosa.power\_to\_db()*. Esta última foi convertida em *json* e enviada para o MQTT broker.

Para que fosse possível a aplicação na cloud controlar a aquisição, foi criada uma *thread* que executa a subscrição a um tópico enviado pela aplicação. Desta forma, uma vez executado o programa, estaria a subscrever em paralelo à execução do restante código continuamente. Uma vez recebida a mensagem “Start”, o programa inicia uma nova aquisição.

## sub.py

Ficheiro executado no Gitpod que recebe os dados do MQTT broker e os apresenta em uma página web local através do Streamlit. Para a apresentação dos dados foi utilizada a função *librosa.display.specshow()* que constrói um espectrograma a partir dos dados enviados. A forma para aceder aos dados foi através da conversão do tipo *string* para *json*, depois de *json* para *np.ndarray*. Para o armazenamento dos dados recebidos, foi utilizada uma variável de sessão *st.session\_state* do tipo *np.ndarray*. Desta forma foi possível aceder aos dados enviados pelo MQTT broker após a execução do Streamlit.

Durante a execução, foram criados blocos com *placeholder.container()* para que a página web ficasse com a mesma disposição após a execução do Streamlit e direcionar onde os dados seriam exibidos.

Para controlar o início da aquisição, foi criado um botão que ao ser acionado, envia a mensagem “Start” para o MQTT broker, que por sua vez é recebida no PC, já em execução. Ainda ao acionar o botão, cria-se uma *thread* que executa em paralelo a subscrição ao tópico referente ao envio dos dados.

Por fim, os dados contidos na variável de sessão são acedidos e primeiramente convertidos em um *panda dataframe*, para depois serem convertidos em csv. A possibilidade de descarregar os dados é feita através de um botão *st.download\_button*.