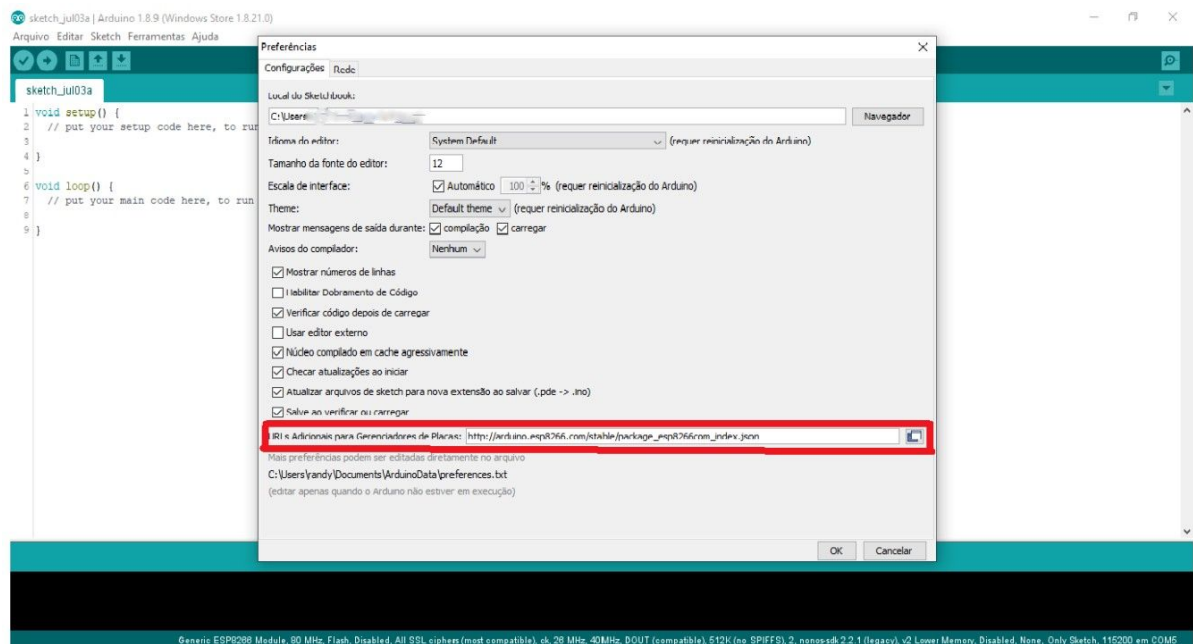


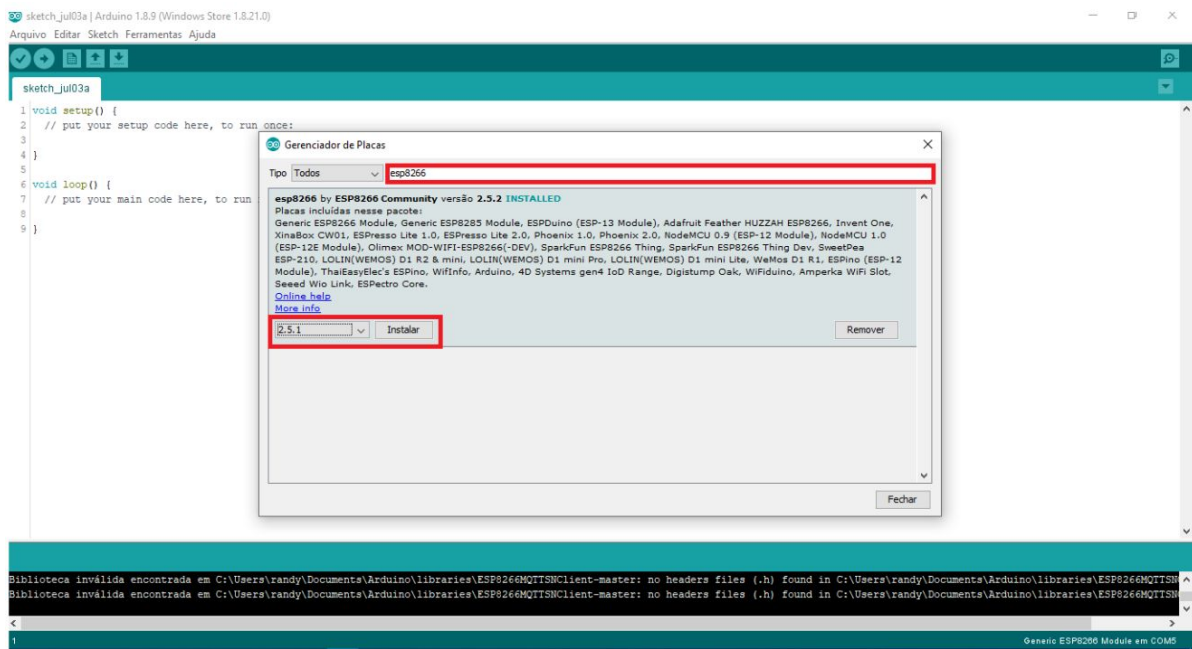
Primeiro, foi baixado a IDE do arduino a partir do link <https://www.arduino.cc/en/Main/Software>. Após a instalação, adicione o json do esp8266 em Arquivo > Preferências > Configurações > URLs adicionais para gerenciadores de placas > Adicionar a seguinte URL > [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) > OK. Conforme a Figura 5.

Figura 5 - IDE do Arduino



Agora, é necessário instalar a biblioteca da placa seguindo para a aba Ferramentas > Placa > Gerenciamento de Placas > Pesquisar “esp8266” > Selecionar a versão mais recente > Instalar > Fechar. Conforme a Figura 6.

Figura 6 - Instalando a biblioteca da placa



Em seguida, deverá ser alterado a placa na qual a IDE deve compilar o código. No menu Ferramentas > Placas > Selecionar “NodeMCU v1.0” > Fechar.

O próximo passo é instalar a biblioteca do cliente MQTT-SN para sistemas embarcados encontrada no link <https://github.com/S3ler/arduino-mqtt-sn-client>. Após baixar a biblioteca, a mesma terá que ser extraída para o diretório de bibliotecas do Arduino. A partir de agora, em Arquivo > Exemplos > Arduino mqtt-sn-client > esp8266 terá exemplos de implementações.

A biblioteca tem apenas dois erros, que são necessários para poder publicar mensagens. Todos os créditos por arrumar esses erros estão no *link* da *issue* (<https://github.com/S3ler/arduino-mqtt-sn-client/issues>) no *github*. Apenas fazendo as modificações constatadas na *issue*, e a biblioteca está pronta para uso.

Será utilizado o *Broker* Mosquitto, disponibilizado no seguinte *link*: <https://mosquitto.org/download/>. A maneira mais fácil de instalar no Ubuntu 18.04 LTS é abrir um terminal de comando e digitar “sudo apt-get install mosquitto” e com isso o programa já estará pronto para uso.

Dentro da pasta /etc/mosquitto/ será possível encontrar o arquivo “mosquitto.conf”, que está todas as configurações do *broker*. As configurações utilizadas nesse artigo estão na Figura 7. Para executar o programa tem que iniciar o programa com o comando: “mosquitto -c /etc/mosquitto/mosquitto.conf”.

Figura 7 - Configuração do *Broker* Mosquitto

```
pid_file /var/run/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

#include_dir /etc/mosquitto/conf.d

log_dest syslog
log_dest stdout
log_dest topic
log_type error
log_type warning
log_type notice
log_type information
connection_messages true
log_timestamp true
allow_anonymous true
#password_file /etc/mosquitto/pwfile
listener 1883
protocol mqtt
```

O último passo de preparação é a instalação de um *Gateway*. Será utilizado o Paho Gateway, que está disponível no *link*: <https://github.com/eclipse/paho.mqtt-sn.embedded-c>.

Após baixar e extrair os arquivos, será necessário entrar na pasta “MQTTSNGateway” e executar os seguintes comandos: “make”, “make install” e “make clean”.

Agora, no diretório onde o usuário instalou o seu *gateway*, terá um arquivo chamado “gateway.conf” que consta as configurações do *gateway* e precisa ser modificado. As modificações feitas para esse artigo estão ilustradas na Figura 8.

Figura 8 - gateway.conf

```

BrokerName=127.0.0.1
BrokerPortNo=1883
BrokerSecurePortNo=8883

#
# When AggregatingGateway=YES or ClientAuthentication=YES,
# All clients must be specified by the ClientList File
#

ClientAuthentication=NO
AggregatingGateway=NO
QoS-1=NO
Forwarder=NO

#ClientsList=/path/to/your_clients.conf

PredefinedTopic=NO
#PredefinedTopicList=/path/to/your_predefinedTopic.conf

#RootCAfile=/etc/ssl/certs/ca-certificates.crt
#RootCApath=/etc/ssl/certs/
#CertsFile=/path/to/certKey.pem
#PrivateKey=/path/to/privateKey.pem

GatewayID=1
GatewayName=PahoGateway-01
KeepAlive=900
#LoginID=your_ID
#Password=your_Password

```

Para iniciar o *gateway*, é necessário apenas executar o seguinte comando: “./MQTTSTNGateway” no diretório onde o mesmo foi instalado. Vale ressaltar que o *Gateway* e o *Broker* estão no mesmo computador operando com o Ubuntu 18.04 LTS, enquanto o cliente em um ESP8266.

## 4.2 Implementação do CoAP

Para a implementação do protocolo CoAP, também foi utilizado a IDE do Arduino, com a biblioteca de desenvolvimento para o ESP8266. Para a instalação dos mesmos, é só seguir os passos mostrados na seção anterior.

Os códigos estão disponíveis no link <https://github.com/automote/ESP-CoAP>. A instalação dessa biblioteca é igual como mostrado na seção anterior.

Com tudo instalado, é só fazer *upload* para dois ESPs (um utilizando o código do cliente, e outro do servidor) e verificar a execução.