

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Campus de Frutal

MQTT USANDO BD MS ACCESS

Professor: Sérgio Portari

Curso: Sistemas Distribuídos

Alunos: Rafaela Queiroz

Wesley Soares

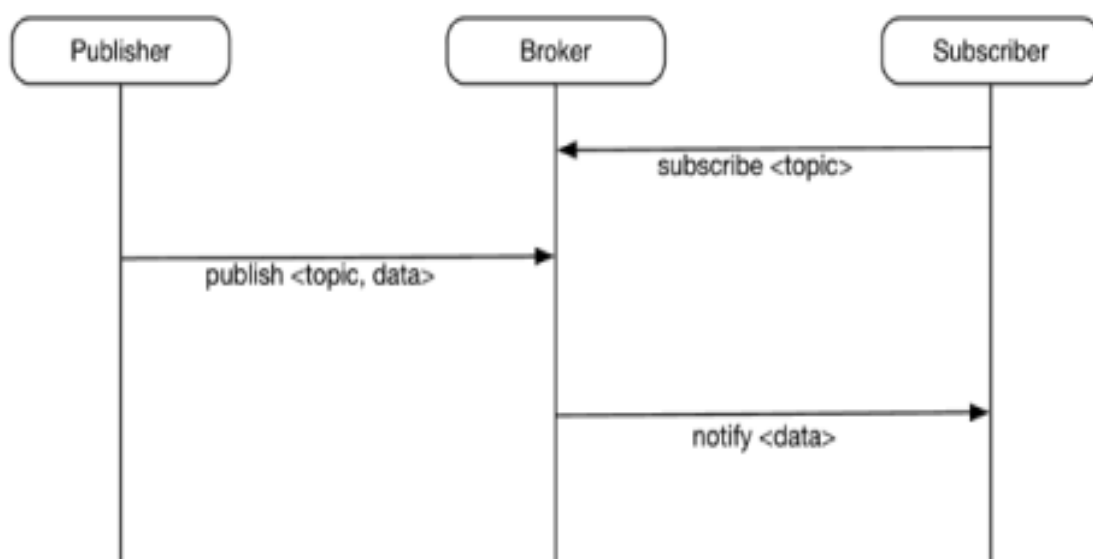
DOCUMENTAÇÃO SOBRE O MQTT USANDO BD MS ACCESS

O protocolo *Message Queue Telemetry Transport* (MQTT) está presente no dia a dia da Internet das Coisas (IoT) e o seu principal uso é fazer as máquinas conversarem, também conhecido como Machine-to-Machine (M2M). O protocolo MQTT foi desenvolvido pela IBM e Eurotech e é projetado para enviar dados através de redes intermitentes ou com baixa banda de dados, para isto o protocolo é desenvolvido em cima de vários conceitos que garantem uma alta taxa de entrega das mensagens.

O protocolo MQTT é baseado no TCP/IP e ambos, cliente e broker, necessitam da pilha TCP/IP para o seu funcionamento.

Pub/Sub

O MQTT utiliza o paradigma [publish/subscribe \(pub/sub\)](#) para a troca de mensagens. O paradigma pub/sub implementa um middleware chamado de broker. O broker é responsável por receber, enfileirar e disparar as mensagens recebidas dos publishers para os subscribers. O publisher é responsável por se conectar ao broker e publicar mensagens. Já o subscriber é responsável por se conectar ao broker e receber as mensagens que ele tiver interesse. Na imagem abaixo possível observar a arquitetura do paradigma pub/sub.



Passo 1 – Download dos Softwares

Passo 2 – Instalações dos Softwares

Passo 3 – Execução MQTT no Access

Passo 1 – Download dos Softwares

Acessar a página oficial Mosquitto - <http://mosquitto.org>

1. Download da versão Windows [mosquitto-1.4.15a-install-win32.exe](http://mosquitto.org/files/bin/mosquitto-1.4.15a-install-win32.exe)

NOTA: a versão disponível é apenas para o processador do computador 32 bits.

Acessar a página oficial Eclipse Paho - <https://www.eclipse.org/paho/>

1. Download da versão Windows <https://projects.eclipse.org/projects/iot.paho>

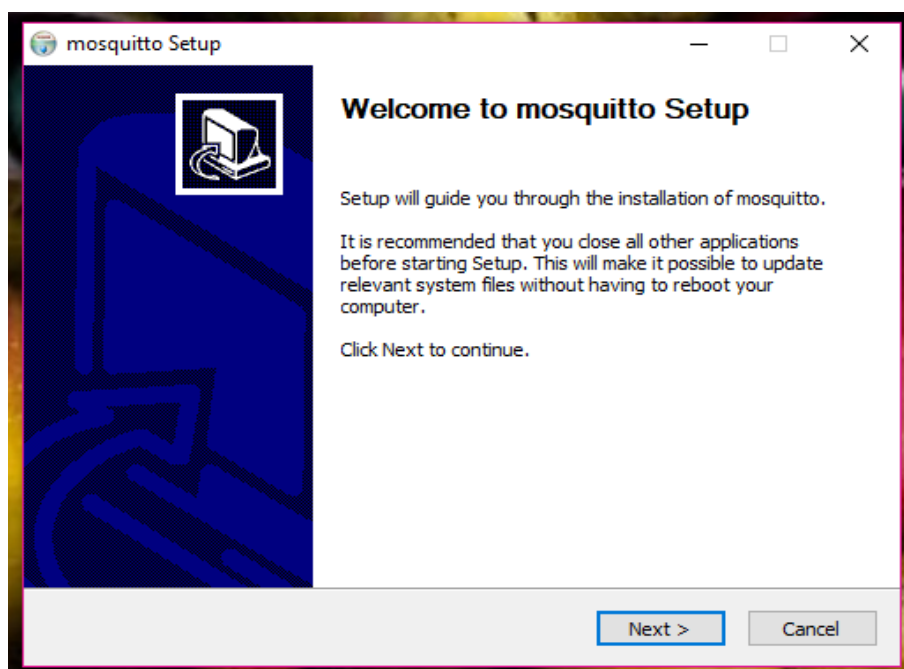
NOTA: escolher o processador do computador 32 ou 64 bits.

Passo 2 – Instalações dos Softwares

As instruções que se seguem têm em conta que se pretende instalar os softwares em computadores com sistema operacional Windows.

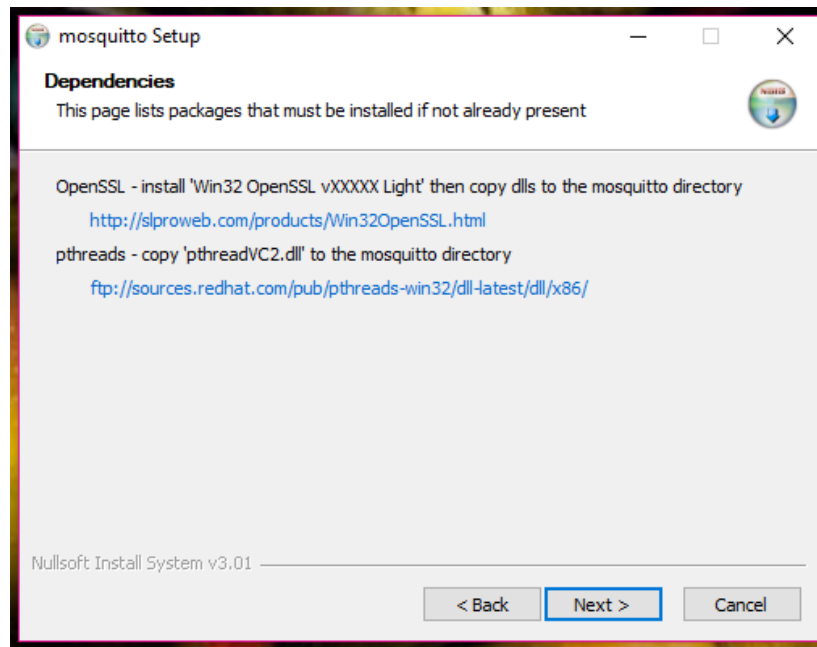
Para instalar o Mosquitto no seu computador deve seguir os seguintes passos:

1. Executar o instalador [mosquitto-1.4.15a-install-win32.exe](http://mosquitto.org/files/bin/mosquitto-1.4.15a-install-win32.exe) em modo Administrador.



2. Inicializar a instalação das dependências do Mosquitto, onde contém duas referências OpenSSL e Pthreads DLL.

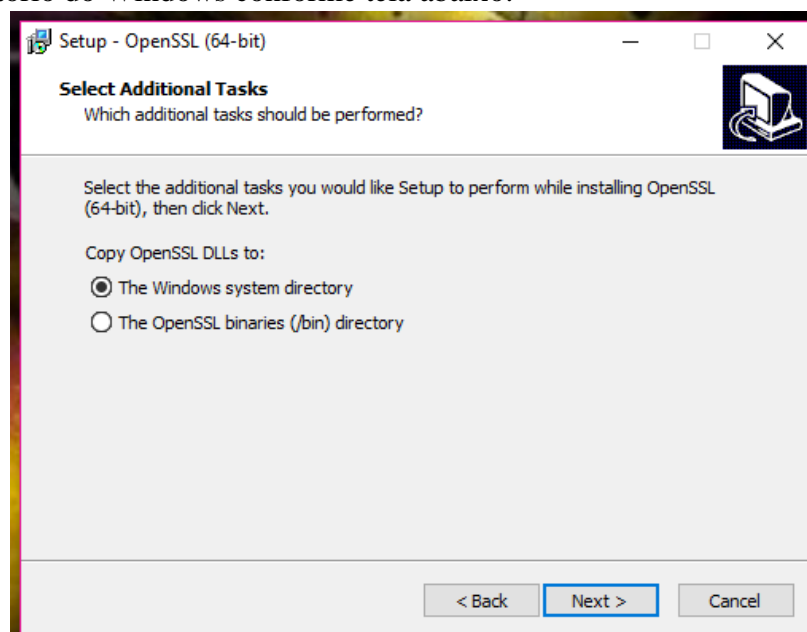
NOTA: o download dos arquivos e a instalação do OpenSSL devem ser feitos preferencialmente antes da instalação ou antes de executar o Mosquitto para que não haja problemas de execução.



3. As próximas telas seguem o padrão de instalação rápida do Windows, "Next... Install", não havendo necessidade em detalhá-las porém é necessário lembrar a pasta onde o Mosquitto foi instalado para efetuar a colocação das bibliotecas acima mencionadas.

NOTA: substituir o arquivo pthreadVC2.dll caso tenha instalado anteriormente.

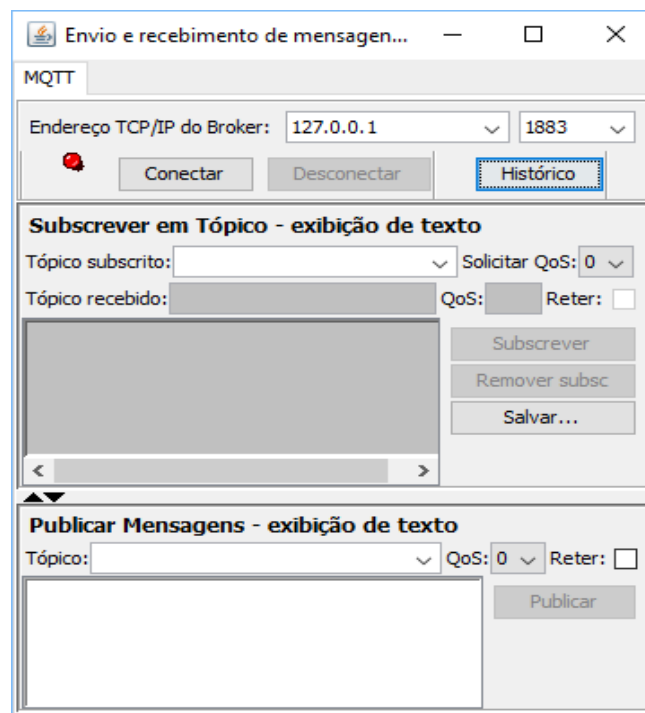
4. Caso a máquina ainda não possua o OpenSSL é só executar seu instalador lembrando de aceitar o "License Agreement" e marcar para copiar as dlls para o diretório do Windows conforme tela abaixo:



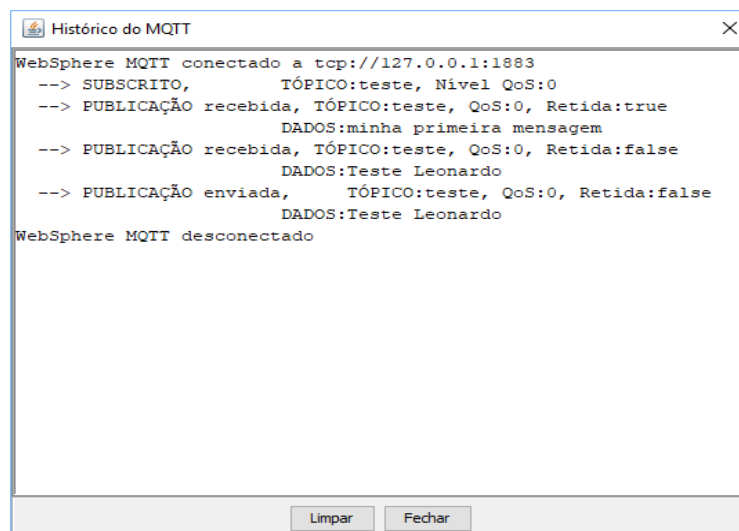
- Para acompanhar a movimentação dos dados dentro das tabelas é importante possuir o MSAccess instalado na máquina mas caso não possua as alterações acontecerão normalmente pois dependem do protocolo jdbc que reconheça a extensão utilizada e não diretamente do aplicativo.

Passo 3 – Estrutura do programa em Java

Foi utilizada uma janela híbrida que possui 3 janelas internalizadas sendo na parte superior a conexão ao servidor, na parte central a janela de subscrição e recebimento de mensagens e na parte inferior a janela de publicação e envio.



Além desta janela é possível exibir um histórico de todas as movimentações feitas no servidor MQTT ao clicar no botão “Histórico”.



A movimentação de mensagens junto ao servidor MQTT é feita por intermédio da ferramenta Eclipse Paho e as conexões por meio da transação ConnOpts interna ao arquivo de mesmo nome e que usa como parâmetros a janela principal MQTTFrame e algumas propriedades que são: IP do servidor ("IPAddressList"), Porta de comunicação ("IPPortList"), Identificação do cliente ("ClientId"), Tipo de persistência ("Persistence"), Diretório do arquivo de persistência ("PersistenceDir") e nome do arquivo de propriedades ("MQTT Utility properties").

```
propFile = new FileOutputStream( PROP_FILE );

// First populate the properties object
// IPADDRESS
props.setProperty("IPAddressList", view.constructPropertyValue("IPAddressList" ) );

// IPPORT
props.setProperty("IPPortList", view.constructPropertyValue("IPPortList" ) );

// CLIENTID
props.setProperty("ClientId", view.constructPropertyValue("ClientId" ) );

// PERSISTENCE ENABLED
props.setProperty("Persistence", view.constructPropertyValue("Persistence" ) );

// PERSISTENCE DIRECTORY
props.setProperty("PersistenceDir", view.constructPropertyValue("PersistenceDir" ) );

// Now write to disk
props.store( propFile, "MQTT Utility properties" );
```

A movimentação no banco de dados MSAccess é feita por intermédio da ferramenta UnCanAccess e as conexões por meio do arquivo ConexaoAccessJava8 e o arquivo externo do banco MQTT.accdb acessado através de um protocolo do tipo jdbc e utilizado nos arquivos PubPanel e SubPanel com inclusões feitas através do método PreparedStatement e consultas através de instrução “Select” SQL:

```
ConexaoAccessJava8 conecta = new ConexaoAccessJava8();
conecta.conexao();
PreparedStatement pst = conecta.conn.prepareStatement("insert into Clie1 (Topico,QOS,Retido,
pst.setString(1, topicName);
pst.setString(2, (String) qosList.getSelectedItemAt().toString());
pst.setBoolean(3, retained.isSelected());
pst.setString(4, pubData.getText());
pst.executeUpdate();
System.out.println("Salvo com sucesso");
```

```

ConexaoAccessJava8 conecta = new ConexaoAccessJava8();
conecta.conexao();
PreparedStatement pst = conecta.conn.prepareStatement("insert into Clie2 (Topico,QoS,Retido,1
pst.setString(1, receivedTopic.getText());
pst.setString(2, receivedQoS.getText());
pst.setBoolean(3, receivedRetain.isSelected());
pst.setString(4, receivedData.getText());
pst.setString(5, receivedTopic.getToolTipText());
pst.executeUpdate();
pst.close();
System.out.println("Salvo com sucesso");

ConexaoAccessJava8 conecta = new ConexaoAccessJava8();
conecta.conexao();
conecta.executaSQL("select * from Clie1");
conecta.res.last();

//Transferindo os dados de Clie1 para a tela
this.receivedTopic.setText(String.valueOf(conecta.res.getString("Topico")));
this.receivedQoS.setText(String.valueOf(conecta.res.getString("QoS")));
this.receivedRetain.setSelected(conecta.res.getBoolean("Retido"));
this.receivedData.setText(String.valueOf(conecta.res.getString("Mensagem")));
this.receivedTopic.setToolTipText(String.valueOf(conecta.res.getString("Seq")));

```

Funcionamento:

Ao executar o programa é aberta a janela principal.

Para iniciar a comunicação é necessário clicar no botão “Conectar”. Caso o servidor MQTT já esteja instalado o botão será desabilitado, será habilitado o botão “Desconectar” e o LED mudará sua cor de vermelho para verde. Obs.: Caso haja alguma instabilidade na conexão com o servidor o LED poderá ficar na cor amarelo e piscando.

Para iniciar o envio de mensagens é necessário criar um tópico através de uma subscrição utilizando o campo “Tópico subscrito” e o botão “Subscrever”. Ao fazer a subscrição o tópico é automaticamente repassado para o campo “Tópico” na seção “Publicar Mensagens”.

Obs.:

- Só é possível publicar mensagens em tópicos com subscrição feita;
- Caso haja alguma mensagem retida no servidor a mesma será exibida preenchendo os campos “Tópico recebido” e “Mensagem”

Para enviar uma mensagem usa a parte inferior da janela, nela escolhe-se o tópico, o nível de qualidade do envio (QoS), se a mensagem ficará retida no servidor, escreve-se a mensagem e clica-se no botão “Publicar”. Ao clicar neste botão a mensagem é enviada ao servidor MQTT e os dados Tópico, QoS, Reter e Mensagem são salvos na tabela Clie1 no banco de dados.

Após a publicação o programa fará a recuperação da mesma no servidor MQTT e exibirá nos campos próprios na seção “Subscrever”.

Ao clicar no botão “Salvar...” o programa recupera os dados salvos na tabela Clie1 e exibe nos campos Tópico recebido, QoS, Reter e Mensagem da seção “Subscrever”, feito isto ele recupera novamente os dados dos campos e salva novamente na tabela Clie2 finalizando assim o objetivo proposto do trabalho.

REFERENCIAS BIBLIOGRÁFICAS

- ✓ <https://github.com/andrew-nguyen/ucanaccess.git>
- ✓ <https://github.com/eclipse/paho.mqtt.java.git>
- ✓ <https://mvnrepository.com/artifact/org.eclipse.paho/org.eclipse.paho.client.mqttv3/1.2.0>
- ✓ <https://www.youtube.com/watch?v=yM1eljamBwM>