Manual on How to Run UCI

1. We need to install a couple of libraries that are required for our program. These libraries are as follows:

```
sudo apt-get install build-essential  libactivemq-cpp-dev libapr1-dev libaprutil1-dev uuid-dev libssl-dev libgps-dev libdbus-1-dev libsystemd-dev libcap2-dev
```

2. Before starting the program, we need to configure the ActiveMQ Apache server. This server does not run on the machine where our program will be executed; it will be hosted on a local server, to which all clients needing to run UCI will connect.

-Install java :

```
sudo apt install openjdk-11-jdk
```

-Download activemq apache :

```
wget https://archive.apache.org/dist/activemq/5.18.5/apache-activemq-5.18.5-bin.tar.gz
```

-To decompress el activemq apache:

```
tar -xvzf apache-activemq-5.18.5-bin.tar.gz
```

-move the servidor activemq broker at opt

```
sudo mv apache-activemq-5.18.5 /opt/activemq
```

-Change the configuration file.

Enter /opt/activemq/conf and change the activemq.xml file that is at the root where you found this document, it has the same name.

-Lift the server

```
/opt/activemq/bin/activemq start
```

3. In the previous step we changed a file called activemq.xml, this is because it contains a directive where users are defined with their roles and passwords if they enter they will have to look for the following properties:

```
<plugins>
   <!-- <jaasAuthenticationPlugin configuration="activemq-domain" /> ->
   <simpleAuthenticationPlugin>
      <users>
         <authenticationUser username="santiago" password="sol0"  />
         <authenticationUser username="jorge"  password="itz1"/>
      </users>
   </simpleAuthenticationPlugin>
</plugins>
```

To add a user you need to put a line within users with the authenticationUser property and define their password and password, we have this by default, however we have to look for a lesser way to manage users, I am on the lookout to perform this task.

4.Allow the port to be enabled, and view the public ip of our local server

```
sudo ufw allow 61616/tcp
```

```
wget -qO- ifconfig.me
```

5.If they added a user, they must take the username and password and put in the configuration of the program that is in:

Uci/Conf.xml

There you will find the following properties:

```
<Conf>
   <Properties type="brokerURI">tcp://localhost:61616</Properties>
   <Properties type="topicName">TEST.TOPIC</Properties>
   <Properties type="clientID">1</Properties>
   <Properties type="subscriptionName">MyDurableSubscription</Properties>
   <Properties type="username">santiago</Properties>
   <Properties type="password">sol0</Properties>
   <Properties type ="filenameMessage">Messages.xml</Properties>
   <Properties type="key-openssl">a3f9b6c8d2e4f1a7c6e8b9d0a1f2c3e4d5f6a7b8c9d0>
   <Properties type="iv-openssl">4f3b2a1d9c8e7f6d5b4a3c2f1e0d9a8b</Properties>
</Conf>
```

They must put in username and password the data they created in the previous file activemq.xml, they must also change the clientID each client must have a different identifier, for the moment it must be put manual however a method can be developed to make it automatic or in another more friendly way, at the same time they must put the public ip that they obtained in the brokerURI property, they must only change the localhost for the public ip

6. Now they must shut down the server and raise it again, this so that new users can read:

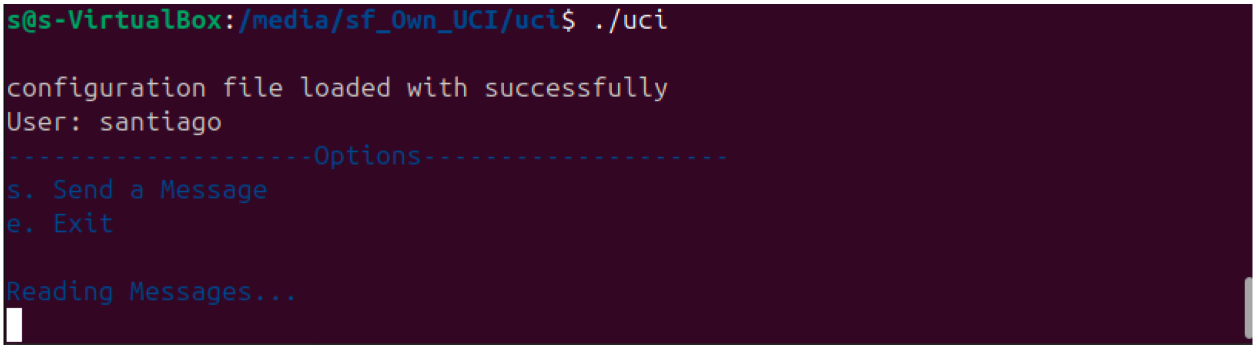
```
/opt/activemq/bin/activemq stop
```

```
/opt/activemq/bin/activemq start
```

7.Finally, we can already run the program, we enter uci/ and execute:

```
./uci
```

If everything goes well, our execution should look like this:



```
s@s-VirtualBox:/media/sf_Own_UCI/uci$ ./uci

configuration file loaded with successfully
User: santiago
--------------------Options--------------------
s. Send a Message
e. Exit

Reading Messages...
```

In the first instance our program is in read mode, that is, it is listening to any message that is sent on the ip and port in which we are connected, these messages are sent encrypted and when receiving them decrypted, when receiving the messages we receive the following parameters:

Received message: hola como están?

with uuid:ba44f943-506a-4091-aa62-e96ba3b8032c

date Time: 2024-08-26 11:35:21

Latitude and Longitude: 0.000000,0.000000

If we want to send a message we press the letter : s on our keyboard and then we enter the message we want to send and we give enter, the program will again enter in read mode, if we want to exit the program we press the letter : e.

All messages sent and received are stored at the root of where our program is in a file called: Messages.xml

There is the UUID, date time, latitude and longitude, and of course our message.