

Práctica 3

Nikita Polyanskiy

Y4441167L

Para la puesta en marcha:

1. Se inicia ActiveMQ (MTIS\apache-activemq-5.15.8\bin\win64\activemq.bat),
2. Se abre el proyecto de la Consola Central en Visual Studio 2022, y se ejecuta.
3. Se abre el código de Oficina1.py y Oficina2.py en Visual Studio Code, cada uno en una ventana separada, y se ejecutan.
4. Vemos la interacción y el estado de las oficinas y la consola desde los terminales:

```
C:\Users\niktr\source\repos\ConsolaCentral1\bin\Debug\net6.0\ConsolaCentral1.exe

Starting up Central Console...
Starting listener - topic:LectTemp2
Starting listener - topic:LectIllum1
Starting listener - topic:LectTemp1
Starting listener - topic:LectIllum2
Illum Office-1: 638 lumens
Temp Office-1: 47°C
Start illum activator for Office 1
Start temp activator for Office 1
Illum Office-2: 549 lumens
Start illum activator for Office 2
Temp Office-2: 1°C
Start temp activator for Office 2
Temp Office-1: 45°C
Illum Office-1: 636 lumens
Illum Office-2: 547 lumens
Temp Office-2: 3°C
Temp Office-1: 43°C
Illum Office-1: 634 lumens
Illum Office-2: 545 lumens
Temp Office-2: 5°C
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\niktr> & C:/Python310/python.exe "c:/User
y"
Starting up Office-2...
Starting listener - topic: ActIllum2
Starting listener - topic: ActTemp2
Current light level: 549 lumens
Current temperature: 1°C
Starting publisher - topic:LectIllum2
Starting publisher - topic:LectTemp2
Received a message "ilum:450"
Decreasing light level to 450 lumens
Received a message "temp:22"
Increasing temperature to 22°C
Current light level: 548 lumens
Current temperature: 2°C
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\niktr\Desktop\MTIS P3\Codigo> & C:/Pyth
P3/Codigo/Oficina1.py"
Starting up Office-1...
Starting listener - topic: ActIllum1
Starting listener - topic: ActTemp1
Current light level: 638 lumens
Current temperature: 47°C
Starting publisher - topic:LectIllum1
Starting publisher - topic:LectTemp1
Received a message "ilum:450"
Received a message "temp:22"
Decreasing light level to 450 lumens
Decreasing temperature to 22°C
Current temperature: 46°C
```

Implementacion:

- Se ha implementado la parte de Consola Central utilizando C# y la librería Apache.NMS.
- Se ha implementado la parte de las 2 oficinas utilizando Python y la librería Stomp.
- Para que la consola o las oficinas puedan recibir mensajes de varios topics a la vez, y enviar otros continuamente, se ha utilizado multithreading en ambos casos.

```
219 static void Main(string[] args)
220 {
221     Console.WriteLine("Starting up Central Console...");
222
223     Thread t1 = new Thread(() => MyListener(topicLi1));
224     Thread t2 = new Thread(() => MyListener(topicLi2));
225     Thread t3 = new Thread(() => MyListener(topicLt1));
226     Thread t4 = new Thread(() => MyListener(topicLt2));
227
228     t1.Start();
229     t2.Start();
230     t3.Start();
231     t4.Start();
232 }
233 }
```

```
159 def main():
160     print("Starting up Office-2...")
161
162     p1 = Thread(target=MyListener.start, kwargs={"topic":topicL1})
163     p2 = Thread(target=MyListener.start, kwargs={"topic":topicL2})
164     p3 = Thread(target=IlumControl.printIlum)
165     p4 = Thread(target=TempControl.printTemp)
166     p5 = Thread(target=MyPublisher.start, kwargs={"topic":topicP1})
167     p6 = Thread(target=MyPublisher.start, kwargs={"topic":topicP2})
168
169     p1.start()
170     p2.start()
171     p3.start()
172     p4.start()
173     p5.start()
174     p6.start()
175
176     p1.join()
177     p2.join()
178     p3.join()
179     p4.join()
180     p5.join()
181     p6.join()
```

- Para la parte de las oficinas (python) se ha utilizado un suscriptor asíncrono para recibir mensajes:

```

25 class MyListener(stomp.ConnectionListener):
26     def on_error(self, frame):
27         print('Received an error "%s"' % frame.body)
28
29     def on_message(self, frame):
30         print('Received a message "%s"' % frame.body)
31         msg = frame.body.split(":")
32         type = msg[0]
33         target = int(msg[1])
34         if(type=="temp"):
35             TempControl.activate(target)
36         elif(type=="ilum"):
37             IlumControl.activate(target)
38         else:
39             print("Error: wrong type (temp/ilum) received")
40
41     def start(topic):
42         print("Starting listener - topic: "+topic)
43
44         conn = stomp.Connection()
45         conn.set_listener('', MyListener())
46         conn.connect('admin', 'password', wait=True)
47
48         conn.subscribe(destination='/topic/'+topic, id=1, ack='auto')

```

- Para la parte de la consola central (C#) se ha utilizado un suscriptor síncrono:

```

33 static void MyListener(string top)
34 {
35     IConnectionFactory factory = new ConnectionFactory(brokerUri);
36     IConnection connection = factory.CreateConnection();
37     ISession session = connection.CreateSession();
38     ITopic topic = session.GetTopic(top);
39     IMessageConsumer consumer = session.CreateConsumer(topic);
40     connection.Start();
41
42     Console.WriteLine("Starting listener - topic: " + top);
43     while (true)
44     {
45         IMessage message = consumer.Receive();
46     }

```

- Se ha creado un total de 8 topics diferentes, de los cuales todos se envían/reciben desde la consola central:

```

9
10 private const string topicLi1 = "LectIlum1";
11 private const string topicLt1 = "LectTemp1";
12 private const string topicPi1 = "ActIlum1";
13 private const string topicPt1 = "ActTemp1";
14
15 private const string topicLi2 = "LectIlum2";
16 private const string topicLt2 = "LectTemp2";
17 private const string topicPi2 = "ActIlum2";
18 private const string topicPt2 = "ActTemp2";
19

```

- En el caso de las oficinas, cada una tiene 2 topics a los que está suscrita, y 2 topics a los que envía mensajes (cada oficina tiene 4 diferentes):

```

12 topicL1 = "ActIlum2"
13 topicL2 = "ActTemp2"
14
15 topicP1 = "LectIlum2"
16 topicP2 = "LectTemp2"

```

Funcionamiento de la aplicación:

Cuando se ejecutan las oficinas, se genera un valor aleatorio para la temperatura e iluminación.

Cada 5 segundos, las oficinas envían sus lecturas de temperatura e iluminación actual a la consola central, luego la consola verifica si esos niveles están en el rango adecuado (19-25°C y 400-500 lum), si no lo están, envían a la oficina correspondiente un mensaje con la temperatura/iluminación a la que se debe llegar (22°C y 450 lum).

```
25     private const int tempMin = 19;  
26     private const int tempMax = 25;  
27     private const int ilumMin = 400;  
28     private const int ilumMax = 500;  
29  
30     private const int targetTemp = 22;  
31     private const int targetIlum = 450;
```

En paralelo, cada 2 segundos las oficinas aumentan/disminuyen sus niveles de temperatura/iluminación (si el activador correspondiente está activado).

Cuando las oficinas llegan a la temperatura/iluminación adecuada, la consola central lo detecta, y envía un mensaje para apagar el activador correspondiente. La temperatura/iluminación se mantendrá en ese valor.