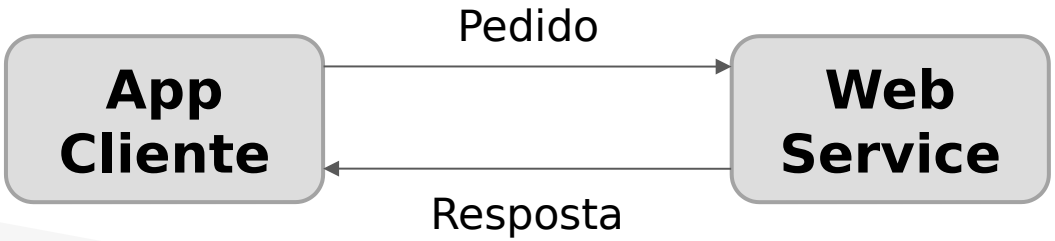
Serviços e Interoperabilidade de Sistemas

Messaging

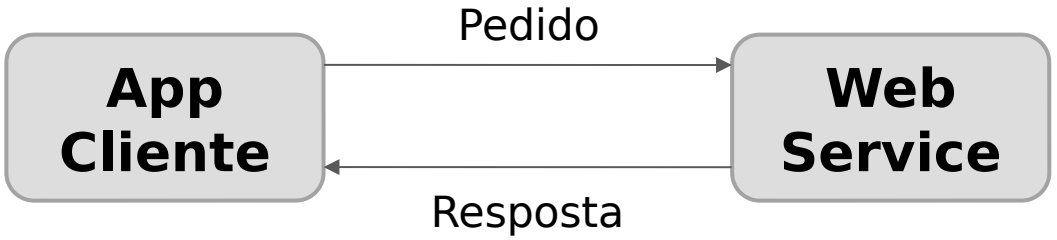
Até agora…

Modelo de interação Pedido/Resposta (Request/Response)

1. Se o servidor falhar…
2. Se servidor voltar a funcionar…
3. Servidor responde mas cliente “cracha”
4. Sem contenção. Se 1.000.000 de pedidos…
…



Até agora...

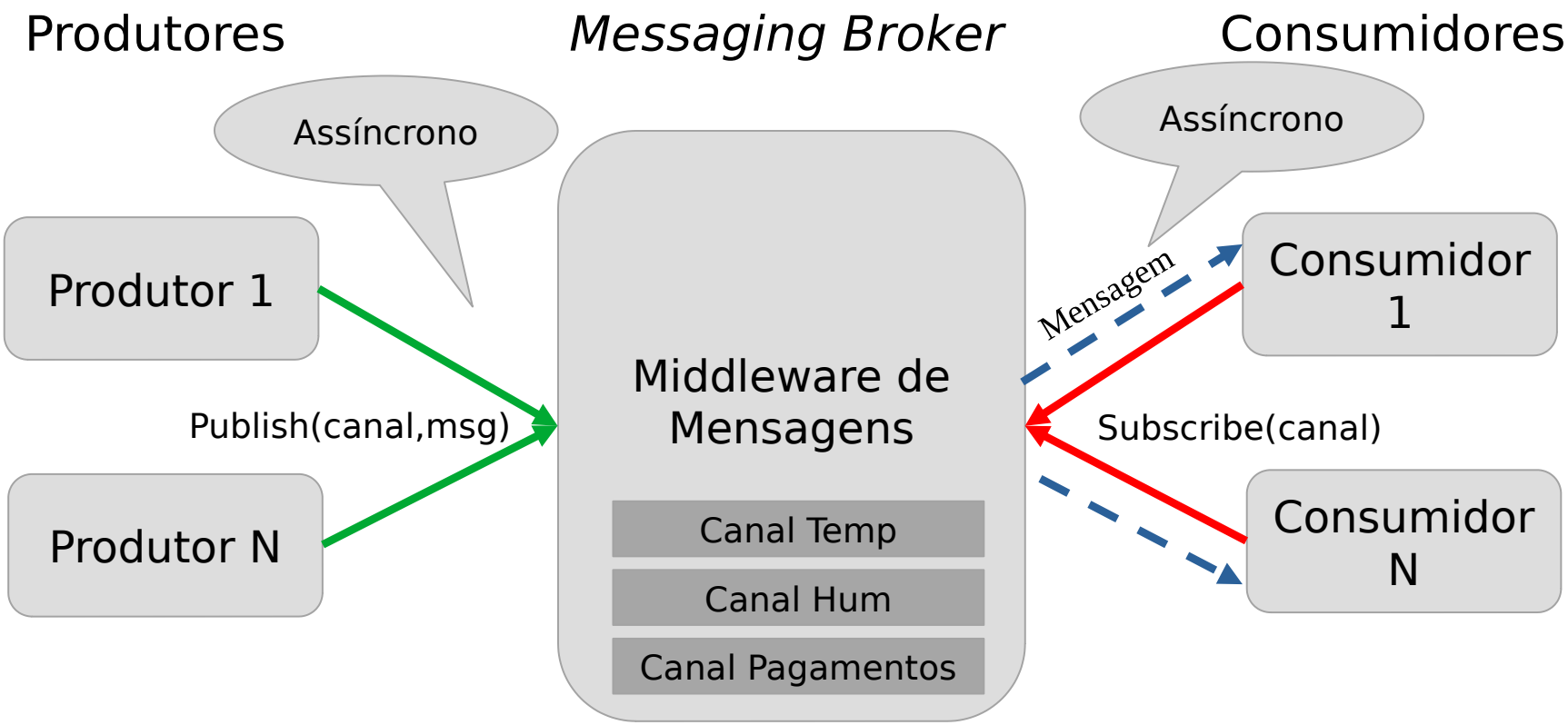


Modelo de interação Pedido/Resposta (Request/Response)

1. Se o servidor falhar...
2. Se servidor voltar a funcionar...
3. Servidor responde mas cliente “cracha”
4. Sem contenção. Se 1.000.000 de pedidos...
...

Alternativas?

Messaging (middleware orientado à mensagem)



Modelo Publish/Subscribe

Messaging (middleware orientado à mensagem)

Rabbit MQ, Mosquitto, ActiveMQ, IBM MQ Series, Tuxedo, …

1. Se ‘servidor’ falha, middleware cliente guarda as msgs
2. Se ‘servidor’ volta a funcionar, recebe msgs pendentes…
3. ‘Servidor’ responde mas cliente cracha. A resposta é guardada e enviada assim que possível…
4. Se 1.000.000 de mensagens, existe contenção ‘natural’

Messaging (middleware orientado à mensagem)

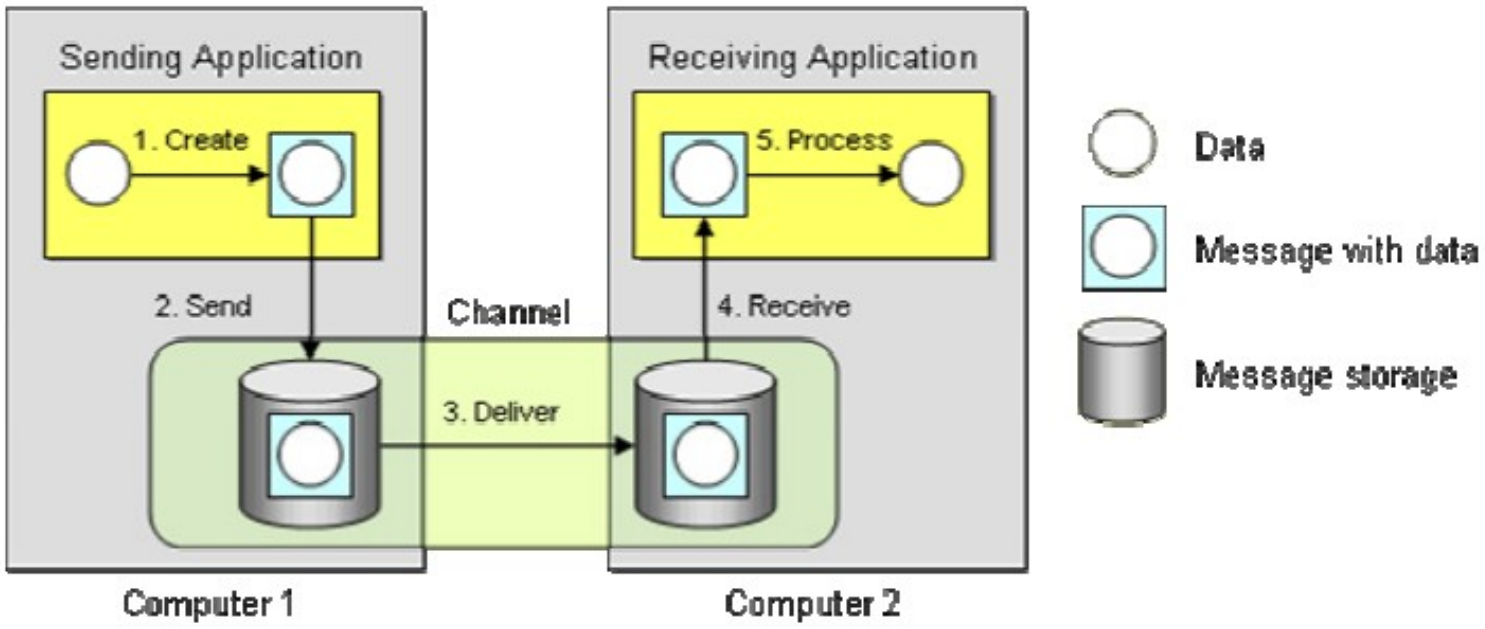
Características dos sistemas de mensagens

- Objetivo: mover mensagens do **produtor** para o **consumidor** de forma **confiável**
- Inclui base de dados – para persistência de mensagens
- Usa canais ou tópicos para definir “temas de conversação”
  - Compras
  - Encomendas
  - Temperatura
  - Tráfego

Messaging (middleware orientado à mensagem)

Características dos sistemas de mensagens

5 operações principais:



Messaging (middleware orientado à mensagem)

Conceitos/abstrações importantes usados:

Envia e esquece (*Send and Forget*) – a mensagem é enviada e o produtor passa à próxima tarefa…

Guarda e encaminha (*Store and forward*) – antes de cada envio, a mensagem é sempre guardada (na origem, broker e destino)

As aplicações delegam 100% no broker a responsabilidade do envio correto das mensagens

Messaging (middleware orientado à mensagem)

Comparação com a abordagem "Pedido/Resposta"

- Desacoplamento máximo (espaço e tempo) ?
- Comunicação assíncrona (*send & forget*)
  - Desempenho máximo (sem espera ativa)
- Evita degradação ou "craches" em situação de "carga"
- Operação "desconetado" tratada com normalidade [1]

  (telemóvel). A sincronização é feita auto/ no *reconnect*…

Messaging (middleware orientado à mensagem)

A integração por *messaging* resolve todos os problemas?

Messaging (middleware orientado à mensagem)

A integração por *messaging* resolve todos os problemas?

- Não! Depende do cenário!
  - Tempo real => Pedido/Resposta é mais apropriado!
  - Várias aplicações => *Publish/Subscribe* é mais apropriado

Erro ou não, preciso de uma resposta!

Exemplo 1: interligar 3 aplicações

Pedido/Resposta: cada app tem que gerir 3 ligações
Publish/Subscribe: cada app gere 1 ligação

Exemplo 2: interligar 10 aplicações

Pedido/Resposta: **45** ligações
Publish/Subscribe: 10 ligações

N(N-1)/2, N corresponde ao número de aplicações envolvidas

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Usa o protocolo MQTT - MQ Telemetry Transport

Binário

Poucos campos (“leve”)

Indicado para Internet das Coisas (sensores)

TCP/IP, porto 1883 (reserved with IANA)

Port 8883 para MQTT por SSL

(Abstraído pela **API** para cada linguagem de programação)

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

QoS – Quality of Service (‘governa’ a entrega das mensagens)

“At most once“ – no máximo uma (mais “leve”)

“At least once“ – pelo menos uma

“Exactly once“ – exatamente uma (“mais pesado”)

\- Recursos usados +

A escolha depende do cenário aplicacional e dos recursos disponíveis…

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Instalação do Mosquitto Messaging Broker

- https://mosquitto.org/download (binary)
- Dependência das bibliotecas **OpenSSL** e **PThreads** (são dadas indicações durante a instalação do Mosquitto)
  - Zip com DLLs no Moodle

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Teste preliminar do Mosquitto Messaging Broker

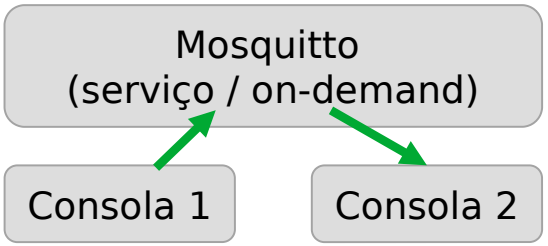
Subscrever canal temp na consola 2:

c:\>mosquitto_sub -v -t ‘temp‘ –h 127.0.0.1

Post para canal temp, na consola 1:

c:\>mosquitto_pub -t ‘temp' -m “27” –h 127.0.0.1

O “27” apareceu na consola 2?



Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

API para C#: M2Mqtt

Abrir Visual Studio e criar app

Tools | Package Manager Console/Nuget:

Install-Package **M2Mqtt** (executar este comando)

Para nos abstrair do protocolo MQTT

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

API para Java: **Eclipse Paho Java Client**

MVN Repository: **org.eclipse.paho.client.mqttv3.test-1.0.2.jar**

Ou

```
<dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.0</version>
</dependency>
```

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Projeto exemplo – Subscriber (NetBeans)

```
public class ClienteSubMQTT implements MqttCallback {
@Override
public void connectionLost(Throwable t)

@Override
public void deliveryComplete(IMqttDeliveryToken token)

@Override
public void messageArrived(String topic, MqttMessage message)
```

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Projeto exemplo – Subscriber (NetBeans)

```
MqttClient myClient;
String clientID = “id_sub”;
myClient = new MqttClient("tcp://127.0.0.1:1883", clientID,null);
myClient.setCallback(this);
myClient.connect();
```

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Projeto exemplo – Subscriber (NetBeans)

```
String myTopic = "news";       int subQoS = 0;
myClient.subscribe(myTopic, subQoS);


....

String[] arrTopics = new String[1]; arrTopics[0] = "news";
myClient.unsubscribe(arrTopics);
```

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Projeto exemplo – Publisher (NetBeans)

```
MqttClient myClient;
String clientID = “id_pub”;

myClient = new MqttClient("tcp://127.0.0.1:1883", clientID, null);

myClient.connect();
```

Messaging (middleware orientado à mensagem)

O caso prático do Mosquitto Messaging Broker

Projeto exemplo – Publisher (NetBeans)

```
String myTopic = "news";
MqttTopic topic = myClient.getTopic( myTopic );


String pubMsg = "{\"pubmsg\":\"1\"}";
int pubQoS = 0;
MqttMessage message = new MqttMessage(pubMsg.getBytes());
message.setQos(pubQoS);
message.setRetained(false);
topic.publish(message);
```