**Dicionário:**

**Publish/subscribe** é um padrão de comunicação assíncrona que permite que os **publishers** publiquem mensagens para um tópico, e os **subscribers** se inscrevam para receber mensagens desse tópico.

**Publishers** são componentes que criam e enviam mensagens.

**Subscribers** são componentes que recebem mensagens.

**Tópicos** são categorias de mensagens.

*Exemplo:*

*Imagine que temos um sistema de monitoramento de temperatura que consiste em um sensor, um message broker e um servidor. O sensor é responsável por coletar dados de temperatura, o message broker é responsável por distribuir as mensagens para os assinantes interessados, e o servidor é responsável por armazenar e processar os dados.*

*O sensor é um publisher. Ele publica mensagens com dados de temperatura.*

*O servidor é um subscriber. Ele se inscreve para receber mensagens do tópico "temperatura".*

*Quando o sensor publica uma mensagem, o message broker distribui a mensagem para o servidor. O servidor então recebe a mensagem e armazena os dados.*

**A atividade prática da disciplina consiste no desenvolvimento de uma versão simplificada de um Message Broker que deve implementar o padrão de mensagens publish/subscribe e que será utilizado na construção e simulação de um ambiente de sensoriamento remoto.**

**O que é Message Broker,explique com exemplo?**

Imagine que você tem um sistema de monitoramento de temperatura que consiste em um sensor, um transmissor e um servidor. O sensor é responsável por coletar dados de temperatura, o transmissor é responsável por enviar esses dados para o servidor, e o servidor é responsável por armazenar e processar os dados.

No modelo tradicional de comunicação, o sensor e o servidor precisam saber um do outro. O sensor precisa saber o endereço IP do servidor para poder enviar os dados, e o servidor precisa saber o endereço IP do sensor para poder receber os dados. Isso cria uma dependência entre os dois componentes, o que pode dificultar o desenvolvimento e a manutenção do sistema.

O padrão publish/subscribe resolve esse problema ao introduzir um intermediário entre o sensor e o servidor, chamado de message broker. O message broker é responsável por receber as mensagens do sensor e distribuí-las para os assinantes interessados. Os assinantes são componentes que estão interessados em receber as mensagens publicadas pelo message broker.

No nosso exemplo, o sensor pode publicar mensagens com dados de temperatura. O servidor pode se inscrever para receber essas mensagens, para que possa armazená-las e processá-las.

Com o padrão publish/subscribe, o sensor não precisa saber o endereço IP do servidor. Ele só precisa saber o endereço IP do message broker. O servidor também não precisa saber o endereço IP do sensor. Ele só precisa se inscrever para receber as mensagens publicadas pelo message broker.

Isso torna o sistema mais flexível e fácil de manter. O sensor pode ser alterado ou substituído sem que seja necessário alterar o servidor. O servidor pode ser movido para outra máquina sem que seja necessário alterar o sensor.

Aqui estão alguns exemplos de como o padrão publish/subscribe pode ser usado:

* Um sistema de monitoramento de temperatura pode usar o padrão publish/subscribe para enviar dados para um servidor central.
* Um sistema de notificação pode usar o padrão publish/subscribe para enviar alertas para usuários ou dispositivos.
* Um sistema de comércio eletrônico pode usar o padrão publish/subscribe para enviar informações de pedidos para um sistema de logística.

No caso da atividade prática da disciplina, o message broker será usado para comunicar sensores remotos com um servidor central. Os sensores irão publicar mensagens com dados de sensoriamento, e o servidor irá se inscrever para receber essas mensagens.

Aqui está uma ilustração do modelo de comunicação usando o padrão publish/subscribe:

[Sensor] -> [Message Broker] -> [Servidor]



Fonte: IA Bard