



# **publish subscribe**

Bianca da Rocha Bartolomei  
Carolina Vasques Moreira  
Luís Otávio Malta Conceição

# Índice

1. [Introdução](#) - Comunicação indireta
2. Sistema [publicar/assinar](#) - O que é
3. Modelos de [inscrição](#)
4. [Questões](#) de implementação
5. Arquitetura do [sistema](#)

A large satellite dish antenna is shown in the lower-left corner, its complex metal framework and mesh surface visible. The dish is angled upwards towards the right. The background is a clear, solid blue sky. Overlaid on the right side of the image is a semi-transparent blue rectangular box containing the text "comunicação indireta" in white, bold, sans-serif font.

# comunicação indireta

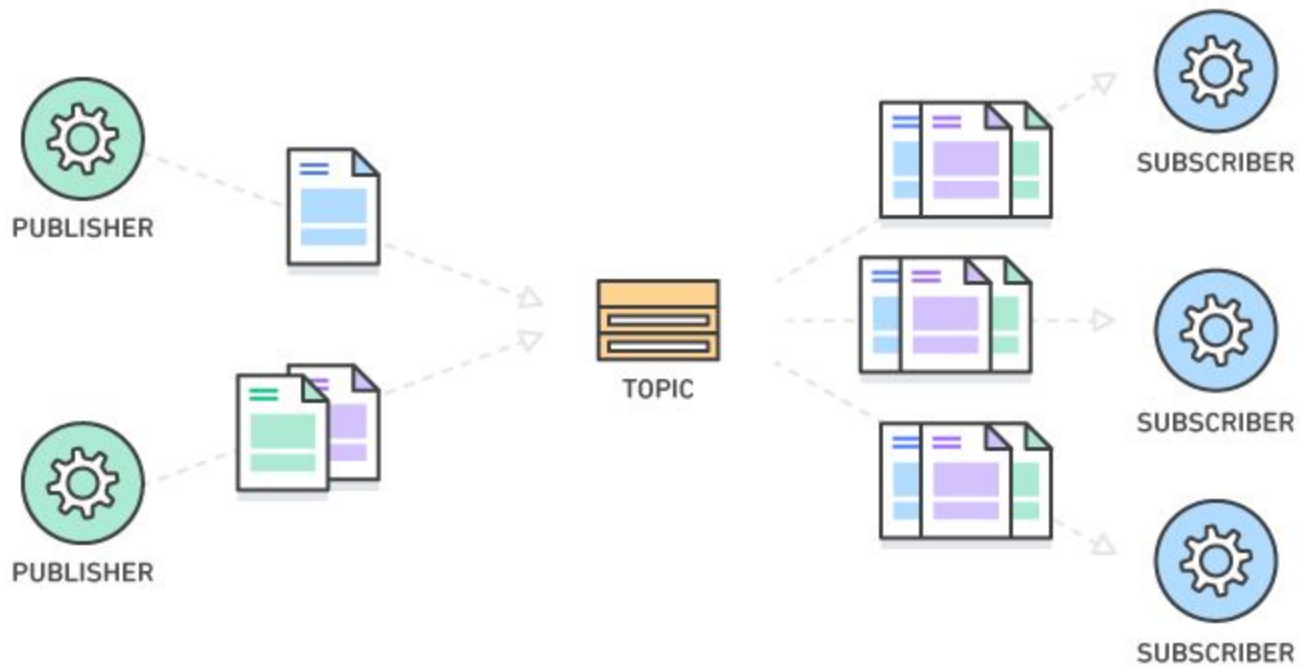
# comunicação indireta

Existe uma entidade intermediária (**broker**) que é responsável pela comunicação entre o remetente e o destinatário; isto permite um alto grau de **desacoplamento** entre eles.

- Desacoplamento espacial
- Desacoplamento temporal



o que é um sistema  
publicar-assinar?





A person is sitting at a desk, working on a laptop. Their hands are on the keyboard. A pair of glasses is on the desk next to the laptop. A smartphone is in the foreground. The entire image has a blue overlay.

# modelos de inscrição

# modelos de inscrição

Formas de mapear uma determinada mensagem de um **publisher** ao **subscriber** interessado:



CANAL



TÓPICO



CONTEÚDO



TIPO



CONTEXTO



CONCEITO



# questões de implementação

```
32 self.fingerprints = set()
33 self.logdups = True
34 self.debug = debug
35 self.logger = logging.getLogger(__name__)
36
37 if path:
38     self.file = open(os.path.join(path, 'requests.log'),
39                     'a')
40     self.file.seek(0)
41     self.fingerprints.update(requests_log)
42
43 @classmethod
44 def from_session(cls, session):
45     return cls(jobs_dir, session)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
```

# questões de implementação

- Implementações centralizadas
  - Gargalo e ponto único de falha
- Implementações distribuídas
  - Rede de brokers ou P2P
  - Roteamento de mensagens:
    - Inundação
    - Filtragem
    - Rendez-vous
    - Anúncios

# arquitetura do sistema









# arquitetura da implementação

Implementação é baseada em **tópico** e **centralizada**. Seus nós são publicadores e assinantes ao mesmo tempo. Uso do **banco de dados**:

- **ps.messages**(id intpkey, corpo varchar, idpub intfkey, idsub intfkey, idtopicfkey, createdat timestamp)
- **ps.nodes**(id intpkey, ip varchar, porta varchar)
- **ps.topics**(id intpkey, name varchar)
- **ps.topicsnodes**(idtopic intpkey fkey, idsub intpkey fkey)
- **trigger** para remoção de mensagens antigas

# funcionalidades do sistema

-  Publicar mensagem;
-  Criar tópicos;
-  Listar tópicos disponíveis;
-  Listar tópicos inscritos pelo nó;
-  Assinar um tópico;
-  Cancelar uma assinatura.





[https://github.com/luismalta/publisher\\_subscriber](https://github.com/luismalta/publisher_subscriber)