

---

# Projecto 1

---

Licenciatura em Engenharia Informática - Computação Distribuída

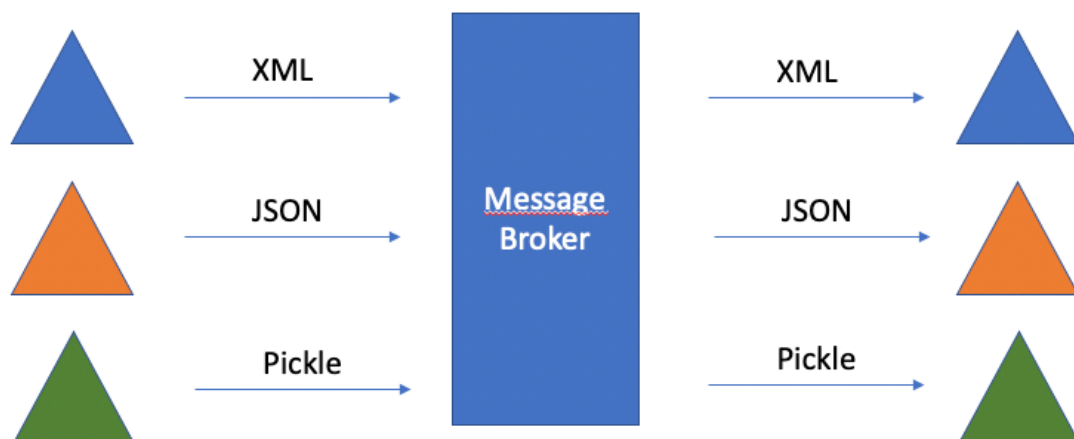
## Message Broker

### Docentes:

- Diogo Gomes ([dgomes@ua.pt](mailto:dgomes@ua.pt))
- Nuno Lau ([nunolau@ua.pt](mailto:nunolau@ua.pt))

### Producers

### Consumers



---

### Conceitos a abordar:

- Message Broker
- Middleware's
- Marshalling

---

### Introdução

Um Message Broker é um middleware tipicamente utilizado em sistemas distribuídos que necessitam de fazer EAI (Enterprise Architecture Integration). Muitas empresas (Enterprises) são fornecidas por fabricantes de software distintos que precisam de ser integrados para completar os processos de negócio da empresa. A integração é tanto mais difícil quando os softwares comunicam através de protocolos distintos e usam serializações de dados distintas.

Os Message Broker são usados nestas situações para traduzir a comunicação entre estes softwares.

No nosso projecto temos um conjunto de produtores (Producers) de dados que precisam de ser consumidos por consumidores que suportam mecanismos de serialização distintos. A tarefa deste

projecto consiste no desenvolvimento de um Message Broker rudimentar que resolva esta situação. Para esta tarefa é fornecido este guião e um código seminal disponível em <https://classroom.github.com/g/ts-DVzu0>

## Objectivos

---

O objectivo deste trabalho é o desenvolvimento de um Message Broker capaz de interligar produtores (Producers) e consumidores (Consumers) através de um protocolo **PubSub** comum e de três mecanismos de serialização distintos (XML, JSON e Pickle).

Para além do Message Broker é necessário o desenvolvimento de um middleware que abstraia os produtores e consumidores de todo o processo de comunicação.

O protocolo PubSub deve ser definido por cada grupo e documentado num relatório. Este protocolo deve ser implementado sobre o protocolo TCP com o Message Broker a fazer uso do porto 8000 para as mensagens de sinalização. Este protocolo deve, no mínimo, incluir:

- Mensagem de **Subscrição** de um tópico
- Mensagem de **Publicação** num tópico

Possíveis outras mensagens:

- Mensagem de **Listagem** de tópicos
- Mensagem de **Cancelamento de Subscrição** de um tópico

Tanto os produtores como os consumidores deverão ser capazes de sinalizar ao Message Broker o mecanismo de serialização que suportam, sendo competência do Message Broker garantir a troca de mensagens de todos para todos independentemente do mecanismo de serialização suportado.

Uma característica dos Message Brokers é a sua capacidade de reter mensagens (persistir) que podem ser entregues aos consumidores assim que estes se ligam. Um consumidor que se ligue ao Message Broker e subscreva um tópico onde previamente foram publicadas mensagens, deve receber a última mensagem publicada.

Junto com este enunciado são fornecidas duas aplicações python (`producer.py` e `consumer.py`) que fazem uso de uma classe que implementa um esqueleto de middleware (`middleware.py`). Não deve fazer qualquer alteração em `producer.py` nem `consumer.py`, mas deve alterar o código de `middleware.py` de acordo com as suas necessidades.

A resolução correcta do exercício pressupõe (por nível de dificuldade crescente):

- 1 Producer – Message Broker – 1 Consumer (1 único mecanismo de serialização)
- 1 Producer – Message Broker – 1 Consumer (2 mecanismos de serialização)
- 1 Producer – Message Broker – N Consumers (N mecanismos de serialização)
- N Producers – Message Broker – N Consumers (N mecanismos de serialização)

Os trabalhos que almejem notas superiores a 16 valores devem implementar tópicos hierárquicos (em árvore). A subscrição de um tópico raiz deve implicar a subscrição implícita de todos os tópicos filhos dessa raiz. Exemplo: a subscrição do tópico *weather* deve implicar a subscrição de *temperature*, *humidity* e *pressure*. A estrutura hierárquica dos tópicos deve usar a seguinte nomenclature: */weather/temperature*, */weather/humidity*, */weather/pressure*. E deve suportar futuros tópicos não previstos e com profundidade crescente.

## Avaliação

---

A avaliação deste trabalho será feita através da submissão de código na plataforma GitHub classroom e de um relatório em formato PDF com não mais de 2 páginas colocado no mesmo repositório junto com o código.

Está em avaliação o protocolo definido e documentado, assim como as *features* implementadas de acordo com os objectivos definidos na secção anterior.