

01



Getting started
Conceitos



AOBA!

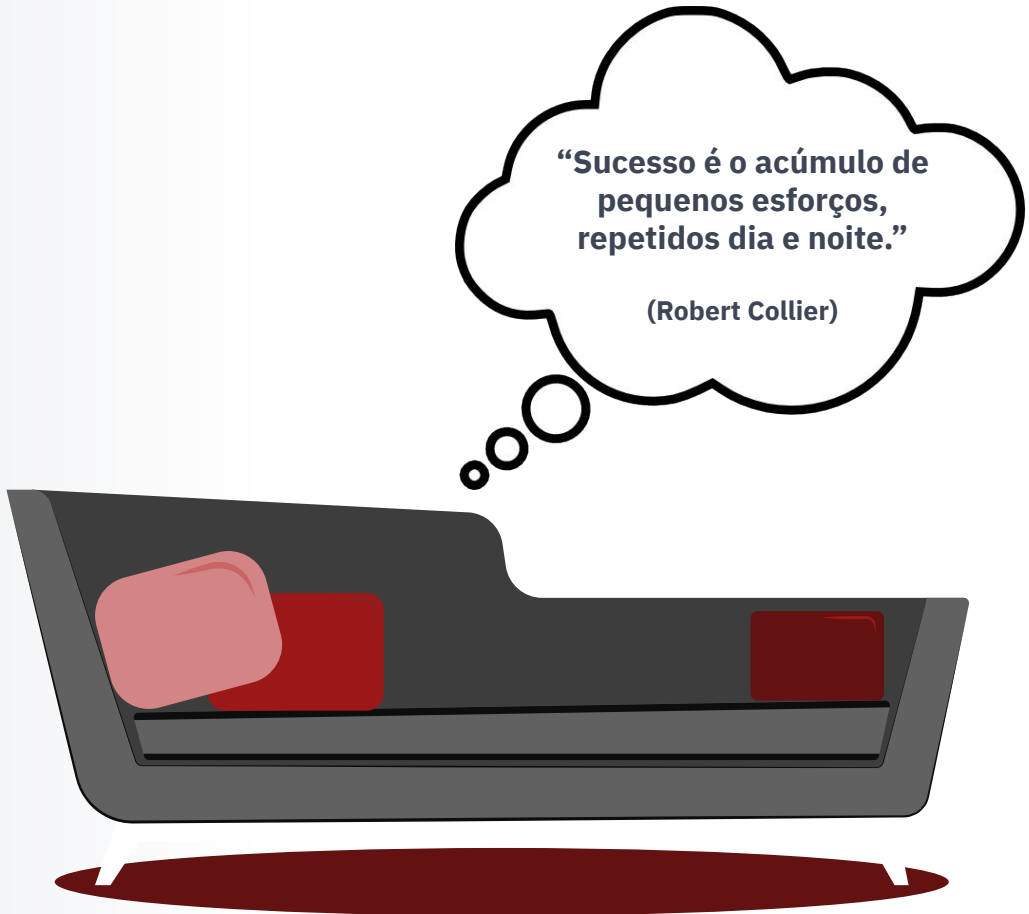
Ricardo de Luna Galdino
Software Engineer



LINKEDIN:
[linkedin.com/in/ricardo-galdino](https://www.linkedin.com/in/ricardo-galdino)

GITHUB:
github.com/ricardogaldino

WHATSAPP GROUP:
[engsoft.org](https://www.whatsapp.com/channel/0029va833333333333333333)



**“Sucesso é o acúmulo de
pequenos esforços,
repetidos dia e noite.”**

(Robert Collier)

Cultura ágil de aprendizado



Microlearning:

- É uma metodologia de ensino que subdivide um assunto em doses menores de conteúdo, com atividades rápidas, auxiliando na compreensão e retenção deste conteúdo.

Pílulas do Conhecimento:

- São pequenos conteúdos apresentados ao profissional para que consiga assimilar de forma mais focada e objetiva, melhorando a eficiência e potencializando os resultados obtidos.
- Microlearning é composto por diversas Pílulas do Conhecimento.

Glossário

Vamos rever alguns conceitos e termos...

Mensageria ou MOM (Message Oriented Middleware)

- ▶ É uma **abordagem arquitetural** para sistemas distribuídos que traz uma infraestrutura de software **“cliente/servidor”** na qual existe uma **camada intermediária (Servidor Middleware)** na comunicação **entre aplicações** (clientes), substituindo a comunicação direta (comunicação síncrona) por um sistema de troca de mensagens (**comunicação assíncrona**).

AMQP

(Advanced Message Queuing Protocol)

- ▶ É um **protocolo** de comunicação utilizado na arquitetura “MOM”, para filas de mensagens, o qual garante a interoperabilidade entre as aplicações clientes e o servidor.
- ▶ Define um **padrão para criar, enviar, receber e ler mensagens**.

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Message Broker

- ▶ É o **“Servidor Middleware de Mensagens”** em uma arquitetura “MOM”.
- ▶ Ele implementa protocolos de comunicação como o “AMQP”.
- ▶ É responsável por **garantir** que a **mensagem** enviada de uma aplicação (produtor) seja **recebida, enfileirada** e enviada corretamente para uma ou mais aplicações (consumidor) de acordo com critérios pré-definidos (**roteamento**).

Exemplos: RabbitMQ, Kafka, ..

RabbitMQ (AMQP)

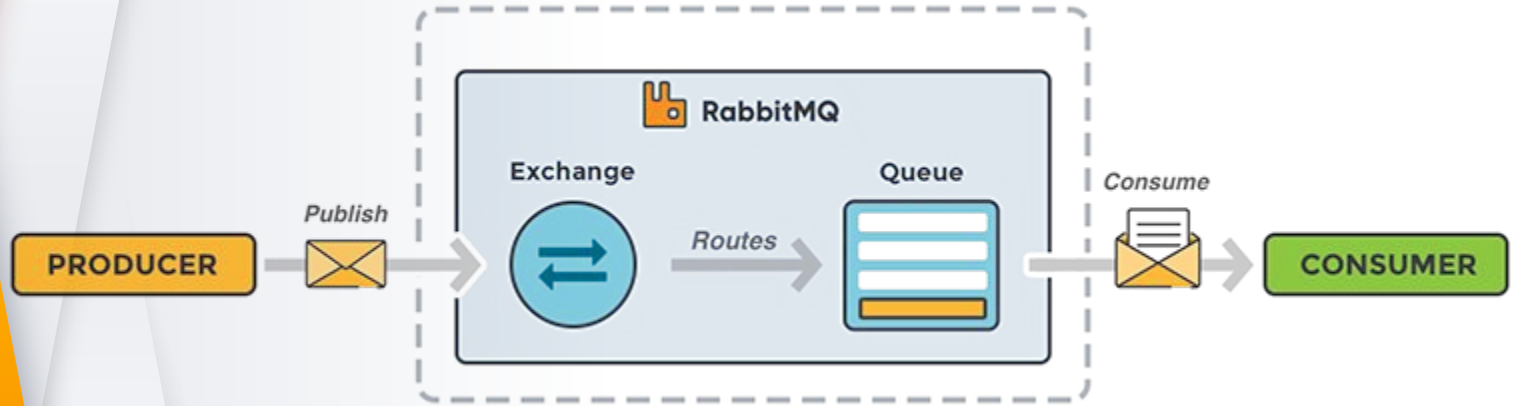
Message Broker

RabbitMQ

<https://www.rabbitmq.com/>

- ▶ É um **Message Broker** de código aberto que implementa o protocolo “**AMQP**”.
- ▶ Escrito na linguagem de programação “**Erlang**” pela Rabbit Technologies, foi lançado em 2007 e seu código fonte foi publicado sob a licença pública Mozilla.
- ▶ Ele segue a estrutura do AMQP, **composto** principalmente **pelas** seguintes **entidades**:
 - ▶ **Producers**
 - ▶ **Consumers**
 - ▶ **Exchanges**
 - ▶ **Queues**
 - ▶ **Bindings**

Fluxo de Mensagens no RabbitMQ



Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Producer (Publisher)

- ▶ É a aplicação que **envia uma mensagem** para o **exchange** do Message Broker.

Consumer (Subscriber)

- ▶ É a aplicação que **consome a mensagem** que está presente em uma **fila** do Message Broker.

Message (Mensagem)

- ▶ São os dados a serem transferidos, geralmente no formato “**JSON**” ou “XML”.
- ▶ Todas as mensagens são **publicadas primeiro em uma exchange para depois serem encaminhadas a uma fila**.

Queue (Fila)

- ▶ É a fila que **recebe** as mensagens geradas por um produtor e **encaminhadas por uma exchange** e que serão consumidas por um ou mais consumers.
- ▶ Ela mantém as mensagens em memória ou em disco até que sejam consumidas.
- ▶ Implementa o método FIFO (First In, First Out).

Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Dead-Letter Queues (DLQ)

- ▶ Às vezes, as **mensagens não podem ser processadas** devido a vários problemas possíveis, como condições errôneas no aplicativo produtor ou consumidor ou uma alteração de estado inesperada que causa um problema com o código do aplicativo. Nesses casos as mensagens **devem ser redirecionadas para uma fila de mensagens mortas** onde deve ser analisado o problema.

Routing Key ou Binding Key

- ▶ As chaves de roteamento são os **identificadores** dos **vínculos entre os exchanges e as filas**.
- ▶ É um **atributo da mensagem** adicionado ao seu cabeçalho pelo produtor.
- ▶ Routing Key (Message) e Binding Key (Exchange) são o mesmo identificador só que em entidades diferentes.

Exchange (Router)

- ▶ É uma espécie de **roteador** que recebe as mensagens de um produtor e **encaminha para filas** específicas **seguindo critérios de roteamento** (vínculos).
- ▶ Os vínculos de roteamento podem variar de acordo com o **tipo da exchange**, sendo eles:
 - ▶ Default
 - ▶ Direct
 - ▶ Fanout
 - ▶ Topic
 - ▶ Headers

**Obs: veremos mais à frente a diferença entre eles!*

Bindings (vínculo)

- ▶ São regras que as exchanges utilizam para rotear mensagens até as filas de destino.
- ▶ Estabelecem uma **relação entre filas e um exchange**.

Entidades do RabbitMQ (AMQP)

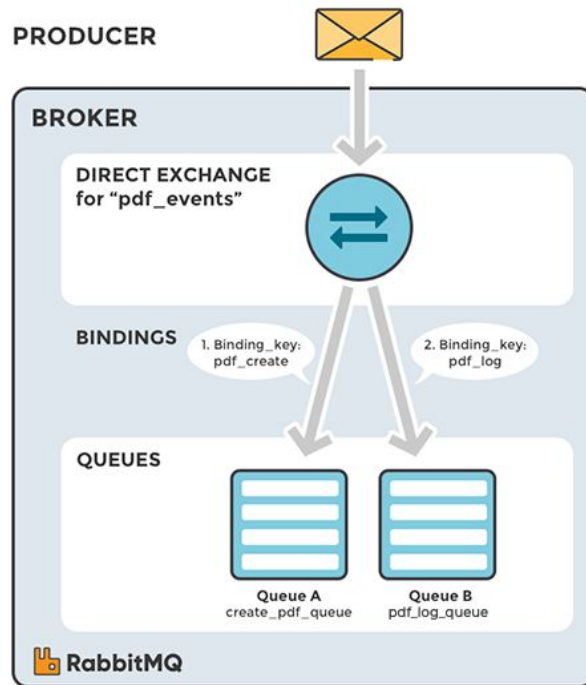
<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Direct Exchange

- ▶ A Direct Exchange entrega mensagens para filas com base na chave de roteamento de mensagens.

Default Exchange

- ▶ A Default Exchange **é uma Direct Exchange** sem nome, pré-declarada pelo Broker quando se cria uma fila.
- ▶ É **criada e vinculada automaticamente a uma fila**.
- ▶ Sua **chave de roteamento** é igual ao **nome da fila**.

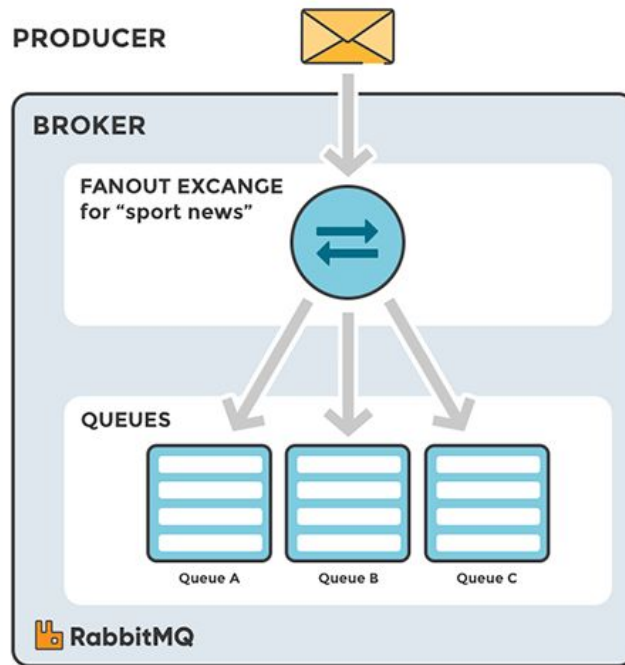


Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Fanout Exchange

- ▶ A Fanout Exchange **copia e roteia mensagens para todas as filas que estão vinculadas a ela independentemente das chaves de roteamento ou correspondência de padrões**, como nas trocas diretas e de tópicos.
- ▶ As chaves fornecidas serão simplesmente ignoradas

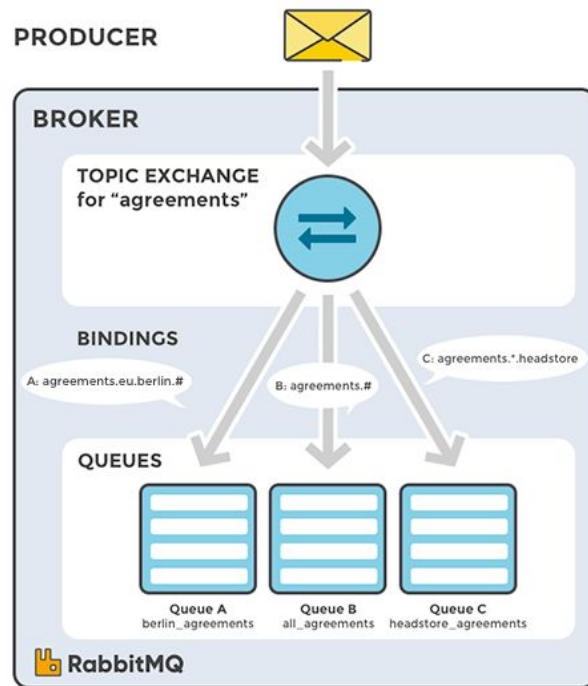


Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Topic Exchange

- ▶ As Topic Exchange **roteiam mensagens para filas com base em correspondências curinga entre a chave de roteamento e o padrão de roteamento**, que é especificado pela ligação da fila.
- ▶ A chave de roteamento deve ser uma lista de palavras, delimitadas por um **ponto (.)**
- ▶ Os padrões de roteamento podem conter:
 - ▶ O **asterisco (“*”)** para corresponder a uma palavra em uma posição específica da chave de roteamento.
 - ▶ O símbolo **“#”** indica uma correspondência de zero ou mais palavras.

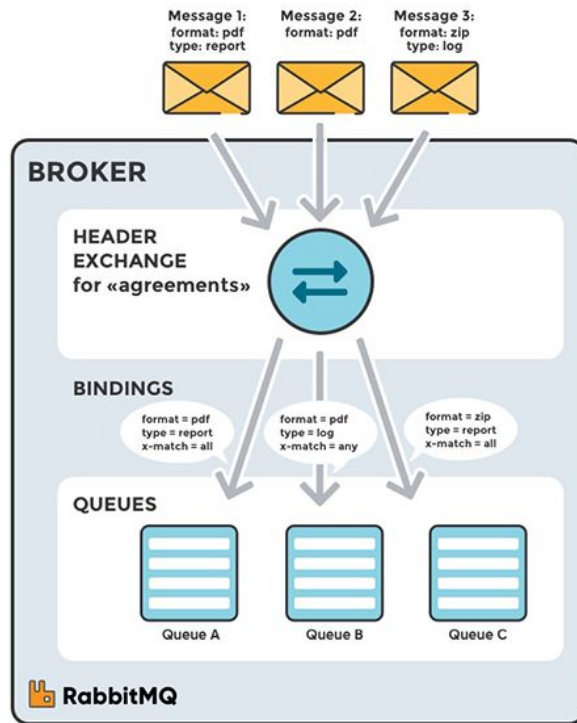


Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Headers Exchange

- ▶ A Header Exchange faz o **roteamento baseado em atributos no cabeçalho da mensagem** e não em uma chave de roteamento.
- ▶ Uma mensagem é considerada correspondente se o valor do cabeçalho for igual ao valor especificado na vinculação.



Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Message Acknowledgements (ack)

- ▶ É a **confirmação** que a aplicação envia ao Broker de que a **mensagem foi consumida com sucesso** e **pode ser removida da fila**.
- ▶ Essa confirmação pode ser automática assim que recebe a mensagem ou através de comandos de acordo com a implementação do desenvolvedor.

Negative Acknowledgements (nack)

- ▶ É a **negação** que a aplicação envia ao Broker informando que **não é possível processar a mensagem** devido algum problema na mesma.

Exemplo:

- ▶ Esperava uma mensagem no formato “JSON” e recebeu em XML.
- ▶ Recebeu um “JSON” com atributos diferentes do acordado (contrato) .
- ▶ A **mensagem pode ser colocada em uma fila “DLQ” ou excluída.**

Entidades do RabbitMQ (AMQP)

<https://www.rabbitmq.com/tutorials/amqp-concepts.html>

Connection (TCP)

- ▶ É a conexão entre a aplicação cliente e o broker RabbitMQ.

Channel (Canal)

- ▶ É uma **subconexão** ou “conexão virtual” **dentro da conexão** TCP. Manter muitas conexões TCP abertas ao mesmo tempo é indesejável, pois consome recursos do sistema e dificulta a configuração de firewalls.
- ▶ Toda **publicação e o consumo de mensagens** **passam por um canal**.

Virtual Hosts

- ▶ É possível **segregar o servidor** RabbitMQ em **“ambientes isolados”** chamados de Virtual Hosts .
- ▶ Em cada Virtual Host podem ser criadas queues, exchanges e outras **entidades isoladas**, com diferentes níveis de acesso.

TO BE CONTINUED...

Getting started **RabbitMQ + Docker**

<https://github.com/ricardogaldino/microlearning-rabbitmq/blob/main/docs/microlearning-rabbitmq-002-docker.pdf>

