

## Mãos à obra: Primeira fila

Como apresentado no video, vamos criar e usar uma fila (*destination*). Isto é enviar uma mensagem e testar o consumidor.

1) Através da interface de administração **crie uma fila** com o nome `fila.financeiro` :



### Queues

| Name            | Number Of Pending Messages | Number Of Consumers | Messages Enqueued | Messages Dequeued | Views  | Operations              |
|-----------------|----------------------------|---------------------|-------------------|-------------------|--|-------------------------|
| fila.financeiro | 0                          | 0                   | 0                 | 0                 | Browse Active Consumers<br>Active Producers<br>atom<br>rss | Send To Purge<br>Delete |

2) Depois **envie uma mensagem** para a fila através da interface de administração (botão *send*).

### Send a JMS Message

| Message Header             |  |  |   |
|----------------------------|--|--|---|
| Destination                | <input type="text" value="fila.financeiro"/> | Queue or Topic                           | <input type="text" value="Queue"/>              |
| Correlation ID             | <input type="text"/>                         | Persistent Delivery                      | <input type="checkbox"/>                        |
| Reply To                   | <input type="text"/>                         | Priority                                 | <input type="text"/>                            |
| Type                       | <input type="text"/>                         | Time to live                             | <input type="text"/>                            |
| Message Group              | <input type="text"/>                         | Message Group Sequence Number            | <input type="text"/>                            |
| delay(ms)                  | <input type="text"/>                         | Time(ms) to wait before scheduling again | <input type="text"/>                            |
| Number of repeats          | <input type="text"/>                         | Use a CRON string for scheduling         | <input type="text"/>                            |
| Number of messages to send | <input type="text" value="1"/>               | Header to store the counter              | <input type="text" value="JMSXMessageCounter"/> |
|                            |  | <input type="button" value="Send"/>      | <input type="button" value="Redefinir"/>        |

  

| Message body  |
|---|
| <input type="text" value="Enter some text here for the message body..."/> |

3) Agora vamos testar o consumidor:

- Para isso baixe o JAR disponível [aqui \(https://s3.amazonaws.com/caelum-online-public/jms/aula-jms.jar\)](https://s3.amazonaws.com/caelum-online-public/jms/aula-jms.jar) e coloque em uma pasta de fácil acesso
- Para esta pasta copie também o JAR principal do ActiveMQ **activemq-all-5.12.0.jar** que está na raiz de instalação do ActiveMQ
- Abra um novo terminal e entre na pasta onde está o *aula-jms.jar* e *activemq-all-5.12.0.jar*
- Execute no terminal:

No Windows:

```
java -cp activemq-all-5.12.0.jar;aula-jms.jar br.com.caelum.TesteMensageria consome
```

No MacOS e Linux:

```
java -cp activemq-all-5.12.0.jar:aula-jms.jar br.com.caelum.TesteMensageria consome
```

Você deve ver o texto da mensagem no terminal. Se tiver com dúvidas pode rever essa parte no vídeo.