

Formulário - Publisher & Receiver

```
/*
 * Pub/Sub Publisher
 */

// imports

public class App {

    public static void main(String[] args) throws Exception {
        TopicName topicName = TopicName.of(project, topic);
        Publisher publisher = Publisher.newBuilder(topicName).build();

        PubsubMessage pubsubMessage = PubsubMessage.newBuilder()
            .setData(ByteString.copyFromUtf8(data))
            .putAttributes("key1", "value1")
            .build();

        publisher.publish(pubsubMessage);

        publisher.shutdown();
    }
}
```

```
/*
 * Pub/Sub Subscriber
 */

// imports

public class App {

    public static void main(String[] args) throws Exception {
        /* Variável de ambiente com chave
        * GOOGLE_APPLICATION_CREDENTIALS=<pathname do ficheiro json com
        chave>
        */

        GoogleCredentials credentials =
        GoogleCredentials.getApplicationDefault();

        FirestoreOptions options = FirestoreOptions
            .newBuilder()
            .setDatabaseId("db-name")
```

```

        .setCredentials(credentials)
        .build();
    Firestore db = options.getService();

    ProjectSubscriptionName subscriptionName =
    ProjectSubscriptionName.of(project, subscription);

    ExecutorProvider executorProvider = InstantiatingExecutorProvider
        .newBuilder()
        .setExecutorThreadCount(1)
        .build();

    Subscriber subscriber = Subscriber.newBuilder(
        subscriptionName,
        new MessageReceiver() {
            @Override
            public void receiveMessage(PubsubMessage message,
AckReplyConsumer consumer) {
                try {
                    // ...
                    consumer.ack();
                } catch (Exception e) {
                    System.err.println("An error occurred: " +
e.getMessage());
                    consumer.nack();
                }
            }
        }
    )
    .setExecutorProvider(executorProvider)
    .build();

    subscriber.startAsync().awaitRunning();

    while (true) {
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            System.err.println("An error occurred: " + e.getMessage());
        }
    }
}

```