Curso: Descomplicando Clean Architecture NA PRÁTICA
Instrutor: Danilo Arantes

```java
package com.arantes.cleanarch.config;

import org.apache.kafka.common.serialization.StringSerializer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.core.DefaultKafkaProducerFactory;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.kafka.core.ProducerFactory;

import java.util.HashMap;
import java.util.Map;

import static org.apache.kafka.clients.consumer.ConsumerConfig.GROUP_ID_CONFIG;
import static org.apache.kafka.clients.producer.ProducerConfig.BOOTSTRAP_SERVERS_CONFIG;
import static
org.apache.kafka.clients.producer.ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG;
import static
org.apache.kafka.clients.producer.ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG;

@Configuration
public class KafkaProducerConfig {

    @Bean
    public ProducerFactory<String, String> producerFactory() {
        Map<String, Object> configProps = new HashMap<>();
        configProps.put(BOOTSTRAP_SERVERS_CONFIG,"localhost:9092");
        configProps.put(GROUP_ID_CONFIG, "arantes");
        configProps.put(KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        configProps.put(VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
        return new DefaultKafkaProducerFactory<>(configProps);
    }

    @Bean
    public KafkaTemplate<String, String> kafkaTemplate() {
        return new KafkaTemplate<>(producerFactory());
    }

}
```