Curso: Arquitetura Hexagonal/Ports and Adapters NA PRÁTICA
Instrutor: Danilo Arantes

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

import com.arantes.hexagonal.adapters.in.consumer.message.CustomerMessage;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.annotation.EnableKafka;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.springframework.kafka.support.serializer.JsonDeserializer;

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

import static org.apache.kafka.clients.consumer.ConsumerConfig.AUTO_OFFSET_RESET_CONFIG;
import static org.apache.kafka.clients.consumer.ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG;
import static org.apache.kafka.clients.consumer.ConsumerConfig.GROUP_ID_CONFIG;
import static org.apache.kafka.clients.consumer.ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG;
import static org.apache.kafka.clients.consumer.ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG;

@EnableKafka
@Configuration
public class KafkaConsumerConfig {

    @Bean
    public ConsumerFactory<String, CustomerMessage> consumerFactory() {
        Map<String, Object> props = new HashMap<>();
        props.put(BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(GROUP_ID_CONFIG, "arantes");
        props.put(KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        props.put(VALUE_DESERIALIZER_CLASS_CONFIG, JsonDeserializer.class);
        props.put(AUTO_OFFSET_RESET_CONFIG, "earliest");
        return new DefaultKafkaConsumerFactory<>(props, new StringDeserializer(), new
JsonDeserializer<>(CustomerMessage.class));
    }

    @Bean
    public ConcurrentKafkaListenerContainerFactory<String, CustomerMessage>
kafkaListenerContainerFactory() {
        ConcurrentKafkaListenerContainerFactory<String, CustomerMessage> factory = new
ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory());
        return factory;
    }

}
```