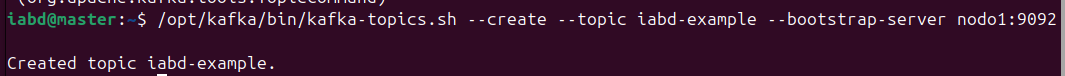
# EJERCICIOS

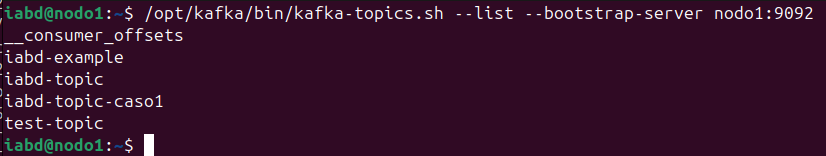
### EJERCICIO 1

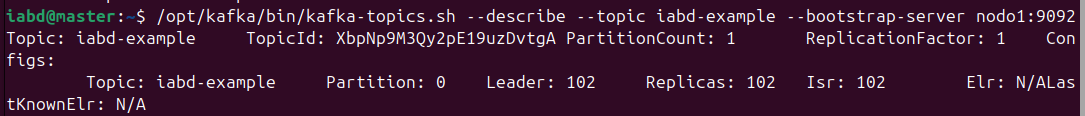
Crea un clúster Kafka con 3 nodos y 3 brokers con sistemas Ubuntu (para hacerlo sigue el documento 4\_2Instlación Kafka).

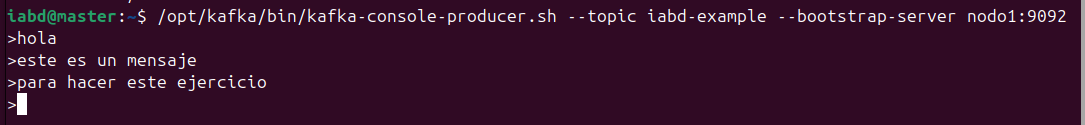
### EJERCICIO 2

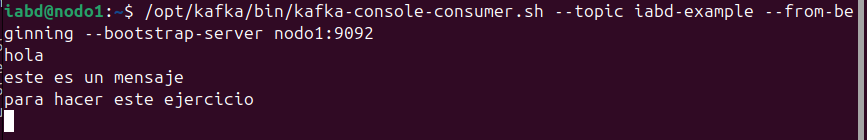
Realiza el caso de uso 0





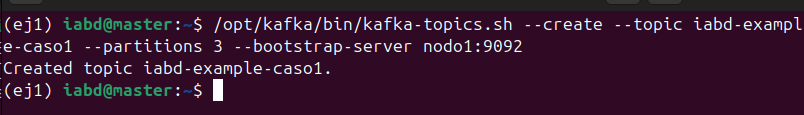


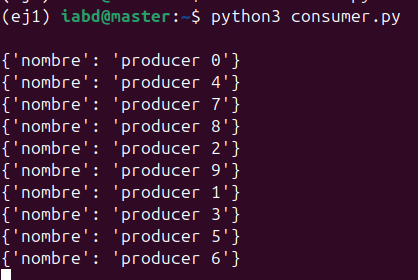




### EJERCICIO 3

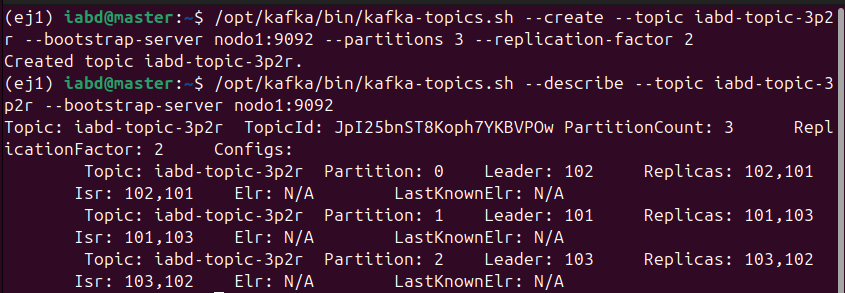
Realiza el caso de uso 1.

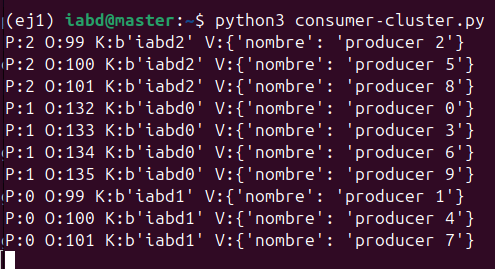




### EJERCICIO 4

Realiza el caso de uso 2





### EJERCICIO 5

Se pide:

1. Crea un topic llamado iabd-topic-ejercicio5 con 4 particiones y un factor de replicación 2.
2. A continuación, lanza dos consumidores que pertenezcan al grupo de consumidores consumer-group-iabd.
3. Lanza un productor y envía varios mensajes compuestos de clave:valor y comprueba cómo aparecen en los consumidores.
4. Detén ambos consumidores.
5. Envía un nuevo mensaje.
6. Obtén información sobre el estado del grupo de consumidores consumer-group-iabd y explica sus valores.

### EJERCICIO 6

Crea un topic denominado iabd-python-topic (con 4 particiones) y utilizando python crea:

1. un productor que envíe datos de una persona cada 10 segundos al topic. Para ello, crea el script personas.py y utiliza la librería Faker para crear los datos ficticios de personas:
2. un consumidor que reciba las personas y, mediante pymongo, las inserte en mongodb en una colección llamada kafka\_personas.

Comprueba que funciona correctamente.

### EJERCICIO 7

Realiza el caso de uso 3

### EJERCICIO 8

Realiza el caso de uso 3, pero separando el consumidorSilverProductorGold en tres consumidores diferentes, uno que consuma y guarde en S3 (consumidorSilverS3.py), otro que consuma y guarde en MongoDB (consumidorSilverMongoDB.py) y el tercero que consuma, agrupe mensajes y produzca el mensaje al topic gold (consumidorSilverGroupProducerGold.py).

### EJERCICIO 9

Antes de hacer el ejercicio tienes que leer el documento BGD\_2\_7\_Nifi.docx (Big Data Aplicado UD2) y hacer los ejercicios BGD\_2\_8\_Ejercicios.docx.

Haz con Nifi del Caso 3, la parte de crear el productor Bronze y el productor Silver, mediante un grupo de procesadores que llamaremos NifiKafka que contendrá un flujo similar a:

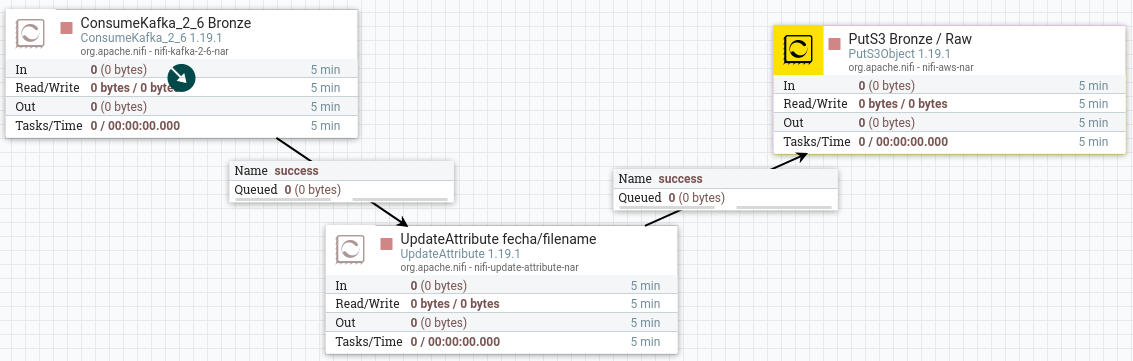
Diagrama

Descripción generada automáticamente con confianza media

Comprueba que funciona correctamente y después de acabar crea un grupo y descarga el flujo en formato json.

### EJERCICIO 10

Siguiendo con el ejercicio anterior ahora con Nifi hay que recuperar los mensajes del topic iabd-aemet-silver y colocarlos en S3, mediante un grupo de procesadores que llamaremos kafka\_consumer\_bronze que contendrá un flujo similar a:



Comprueba que funciona correctamente y después de acabar crea un grupo y descarga el flujo en formato json (a la hora de guardar el S3 nos va a dar un error de credenciales porque PutS3Object no soporta las sesiones temporales que da AWS Academy).

### EJERCICIO 11

Crea un conector (Kafka Connect) para que consuma los mensajes del topic iabd-aemet-bronze y los inserte en MongoDB de forma automática.

Para ello, puedes utilizar el [MongoDB Connector (Source and Sink)](https://www.confluent.io/hub/mongodb/kafka-connect-mongodb) y consultar su [documentación](https://www.mongodb.com/docs/kafka-connector/current/) y un ejemplo de [fichero de configuración](https://github.com/mongodb/mongo-kafka/blob/master/config/MongoSinkConnector.properties).

Ten en cuenta que para lanzar los dos conectores a la vez deberás lanzar Kafka Connect y pasarle todos los conectores como parámetros:

### EJERCICIO 12 (OPCIONAL)

Crear un clúster de Kafka con 4 brokers usando docker.