Evaluación Docker y Comunicación

En el siguiente documento se muestra de manera breve el funcionamiento de lo que es el uso de la arquitectura de (Topología de anillo) el porque se decidió escoger este tipo de topología es por el hecho de que debemos realizar de que tres nodos a, b y c se comuniquen entres si das las circunstancias y requerimientos del pedido del enunciado optamos por esta tipo de arquitectura ya que si bien es algo sencilla de ver puede llegar a complicarse si vemos los de manera indistinta por diferente manera de variables a continuación la imagen del requerimiento completado del ciclo final:

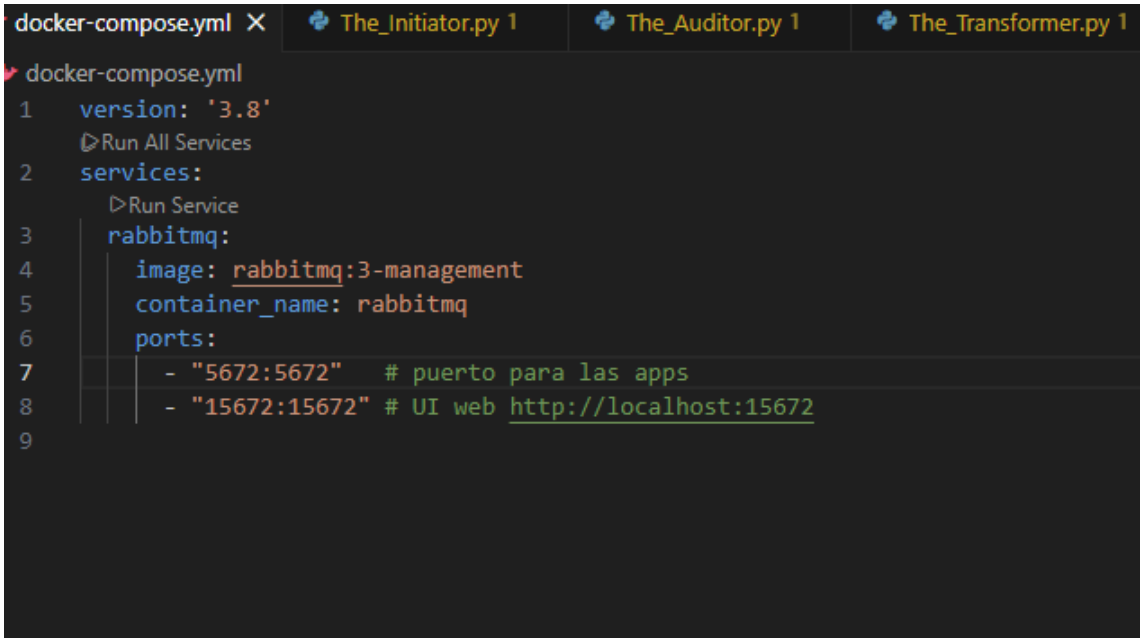
```
PS C:\Users\Estudiante\Desktop\NewPractice\MecanismosDeComunicacion> python The_Initiator.py
[MON-Producer] Enviando valores numericos al NODO A... Ctrl+C para salir
[ENVIADO] {"sensor_id": "124", "power_level": 78, "audit_trail": ""}
[ENVIADO] {"sensor_id": "191", "power_level": 214, "audit_trail": ""}

  ciclo Completado
```

Por otro lado tenemos los nodos que fueron funcionales apero casi parecidos entre si por que lo único que poda lagar a cambiar como tal son las funciones respectivas de cada nodo

```python
import pika
import json
import requests
import The_Initiator

RABBITMQ_HOST = "localhost"
QUEUE_NAME = "sensores_mon"
WS_BRIDGE_URL = "http://localhost:9000/alert"  # endpoint del WS Server

def clasificar_alerta(evento):
    audit_trailxx = evento.get("audit_trail1")
    power_levelxx = evento.get("power_level1")
    sensor_idxx = evento.get("sensor_id1")



    power_levelxx = power_levelxx -5 , "Se resto 5 "
    audit_trailxx = "B_processed"

    alerta_procesada = {
        "sensor_idxx": sensor_idxx,
        "power_levelxx": power_levelxx,
        "audit_trailxx": audit_trailxx
    }
    return alerta_procesada

def enviar_a_ws(alerta):
    try:
        resp = requests.post(WS_BRIDGE_URL, json=alerta, timeout=2)
        if resp.status_code != 200:
```

El en este proyecto tuvimos que realizar 3 commits porque tuvimos errores de sincronización.

Por otro lado también tenemos lo que el archivo de Docker.yaml



Ninguna novedad solo cargamos lo que es el rabbitmq