

Actividad 4 – SEMINARIO ZeroMq



Nuestro objetivo es usar en nuestra arquitectura una solución basada en el middleware ZeroMQ que nos permita aprovecharnos de su solución orientada a las comunicaciones asíncronas.

Cada carrito realiza las mismas operaciones pero las peticiones llegan por medio de mensajes usando el middleware ZeroMQ. **Propón una arquitectura que use los sockets ZeroMQ**, eligiendo aquella combinación **que nos permita**:

- Tener **un componente carrito que reciba peticiones** y pueda asignarse a un usuario (un endpoint), pero dicha asignación se realice a través de un intermediario.
- Dicho intermediario (**broker**) deberá manejar una lista de carritos vacíos, **que asignará** a la primera petición del cliente (puede anotarse su identificación), y hasta que el cliente cierre la conexión. A partir de esa primera operación (que será añadir un el primer producto) todas las operaciones de ese endpoint se realizarán sobre **ese carrito**. Cuando se cierre la conexión del cliente el broker vaciará el carrito y lo registrará como carritos disponibles.
- **La comprobación de stocks** del producto añadido no se realiza directamente por los carritos, si no que se **genera una petición a otro componente** que irá realizando dicha comprobación e irá respondiendo a cada petición del carrito. Un carrito puede generar más de una petición en un momento dado, antes de que la anterior petición de comprobación de stocks obtenga una respuesta.

¿Podría usarse el broker implementado con ZeroMQ y los carritos con REST y conectarse de alguna forma? ¿Proporciona esta solución con ZeroMQ alguna ventaja sobre una API REST?

Pensar cómo encajaría en nuestra arquitectura un mecanismo (usando Zeromq) que permita determinar si el carrito continua funcionando con normalidad. En caso contrario usar el mecanismo del [circuit breaker](#) (proxy de estado fallido) para evitar que se produzcan peticiones repetidas a elementos que probablemente puedan estar caídos, evitando así un error en cascada que colapse el sistema. ¿Que componente realizaría este cometido?