Criterios de separación

Desintegradores

Granularidad:

- Alcance y funcionalidad de un servicio: El objetivo es que cada servicio tenga una funcionalidad del sistema, y de esta manera manejar también una fácil **extensibilidad** con respecto a una entidad del negocio dentro del alcance del servicio.
- Escalabilidad: Tener la posibilidad de escalar una funcionalidad en particular si existe una alta demanda momentánea.
- Seguridad: Poder aislar y atacar las necesidades de seguridad que pueda necesitar el servicio, sin impactar en el resto del sistema (backups, encriptación, etc)

Transacciones:

- Orquestación basada en solicitudes: Se optó por este patrón para garantizar la consistencia eventual en el sistema distribuido. Por lo tanto a partir de un orquestador que manejara la creación, compra y entrega de una orden.

Integradores

Granularidad:

- Transacciones de la base de datos: mantener ACID cuando se reduce el stock y se genera una orden, ya que puede ser crítico a la hora de mantener una consistencia del stock y no rechazar la orden al comprador (ya que afecta negativamente la experiencia del usuario al tener que comprar de nuevo).

Transacciones:

- Consolidación de los servicios: como se mencionó y explicó en el punto de granularidad, se opta por consolidar el servicio de productos y órdenes en uno, ya que se va a requerir manejar una transacción ACID en las colecciones de productos y órdenes.

Responsabilidad de cada componente

order-orchestrator:

- Encargado de orquestar el flujo de compra de un producto, el pago por dicha compra y la entrega o finalización de dicha compra (Incluye además notificación en dichos flujos).

products-orders-service:

- Ownership (CRUD) de los productos y órdenes.
- Brindar la búsqueda y filtrado de productos

users-service:

- Ownership (CRUD) de los usuarios (compradores y vendedores).

notifications-service:

- Realizar las notificaciones al comprador o vendedor.

payments-service

- Encargado de procesar el pago de la orden.

Observaciones

Mejoras: Se agregó RequestId (system-request-id) en logs enviado por header, para debugging entre servicios. Quedó agregar en servicios de payments-service y notification-service.

Cambios:

- Se saca obtener todos los productos del seller en users-service para evitar entregar toda la lista sin paginar de productos. En cambio, se agrega en el endpoint de filtrado de productos que reciba el **sellerId** para filtrar.
- Se modifica que el seller confirme la orden, sino que el customer al pagar se confirme automáticamente.

MongoDBs:

- Se utiliza una misma db, pero cada servicio y repositorio utiliza su propia **collection** correspondiente (*dejando de lado la transacción ACID de compra y stock del producto*). Se podría crear una db para cada repositorio y servicio, pero no lo vemos necesario aún, y por simplicidad lo vemos útil de esta forma.