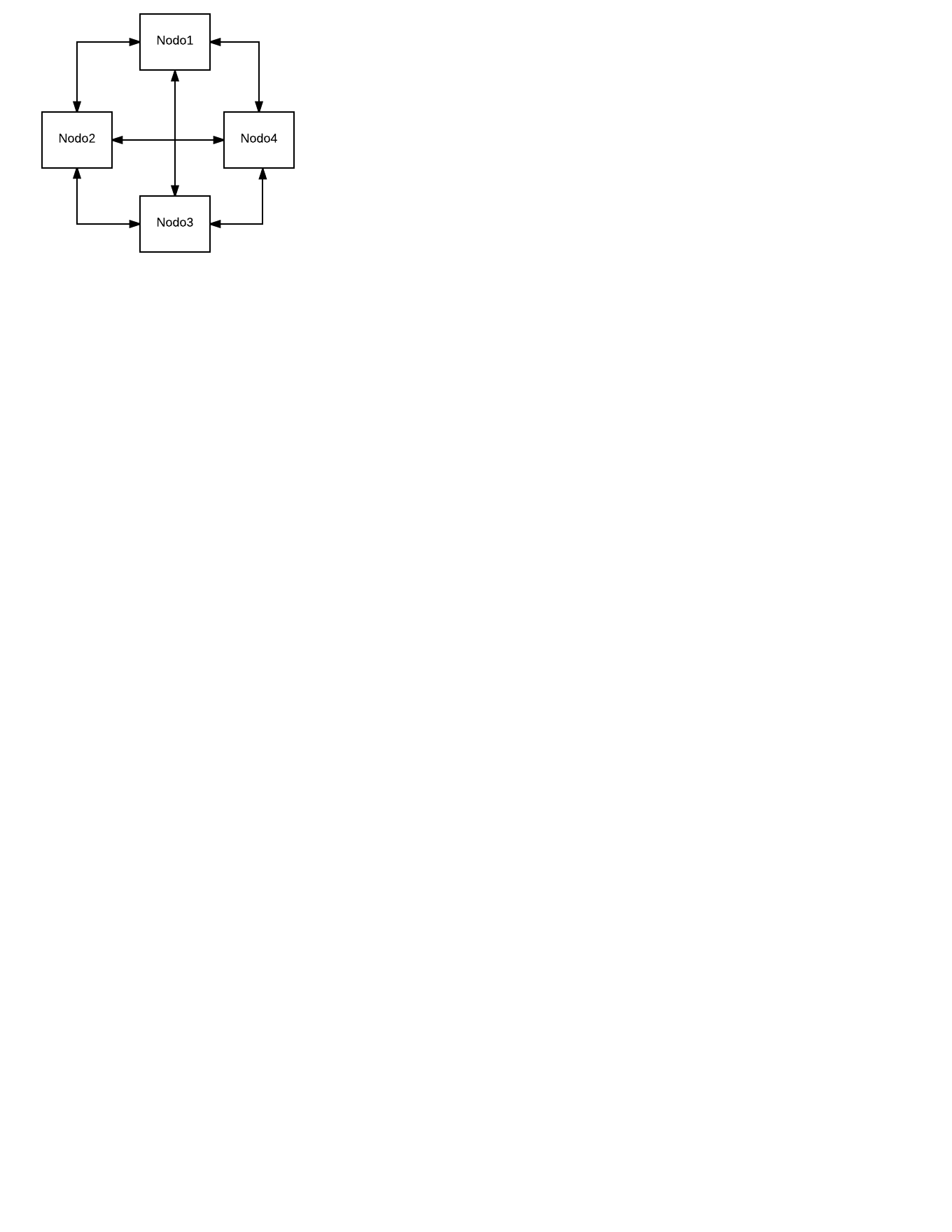
Resolución guia AP3 – Diseño de Arquitectura

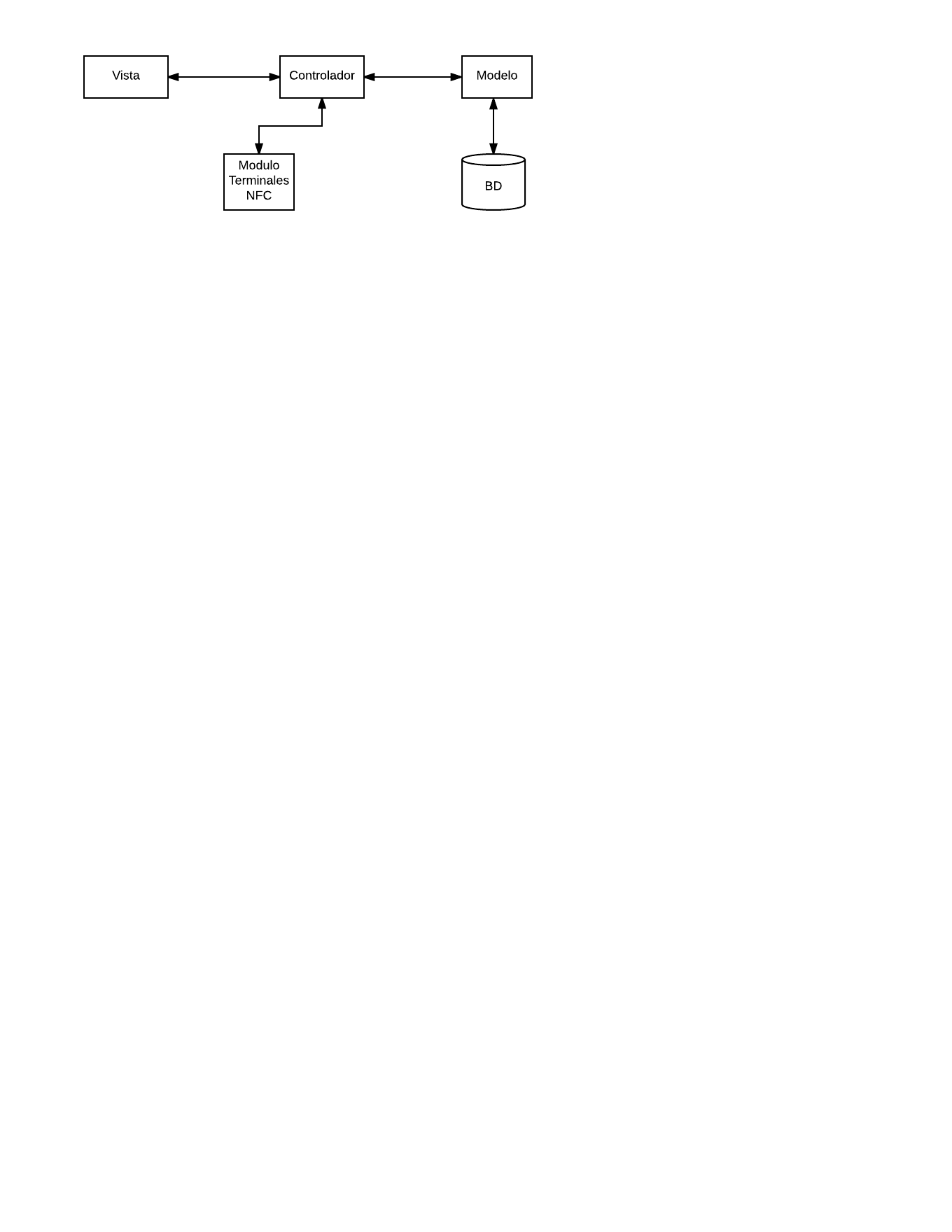
Ejercicio 1

Se pensó en una arquitectura Peer-to-Peer, aprovechando que es descentralizada, usa de la mejor manera el ancho de banda, se evita el único punto de falla, cada nodo puede ser tanto cliente como servidor así que se maximiza el procesamiento y permite una escalabilidad horizontal al poder seguir agregando mas nodos, para esta arquitectura se planteo usar el patrón de integración Call & Return ya que es un modelo donde un componente puede usar un servicio provisto por otro, como desventaja tenemos la dependencia y el bloqueo hasta que la petición es atendida



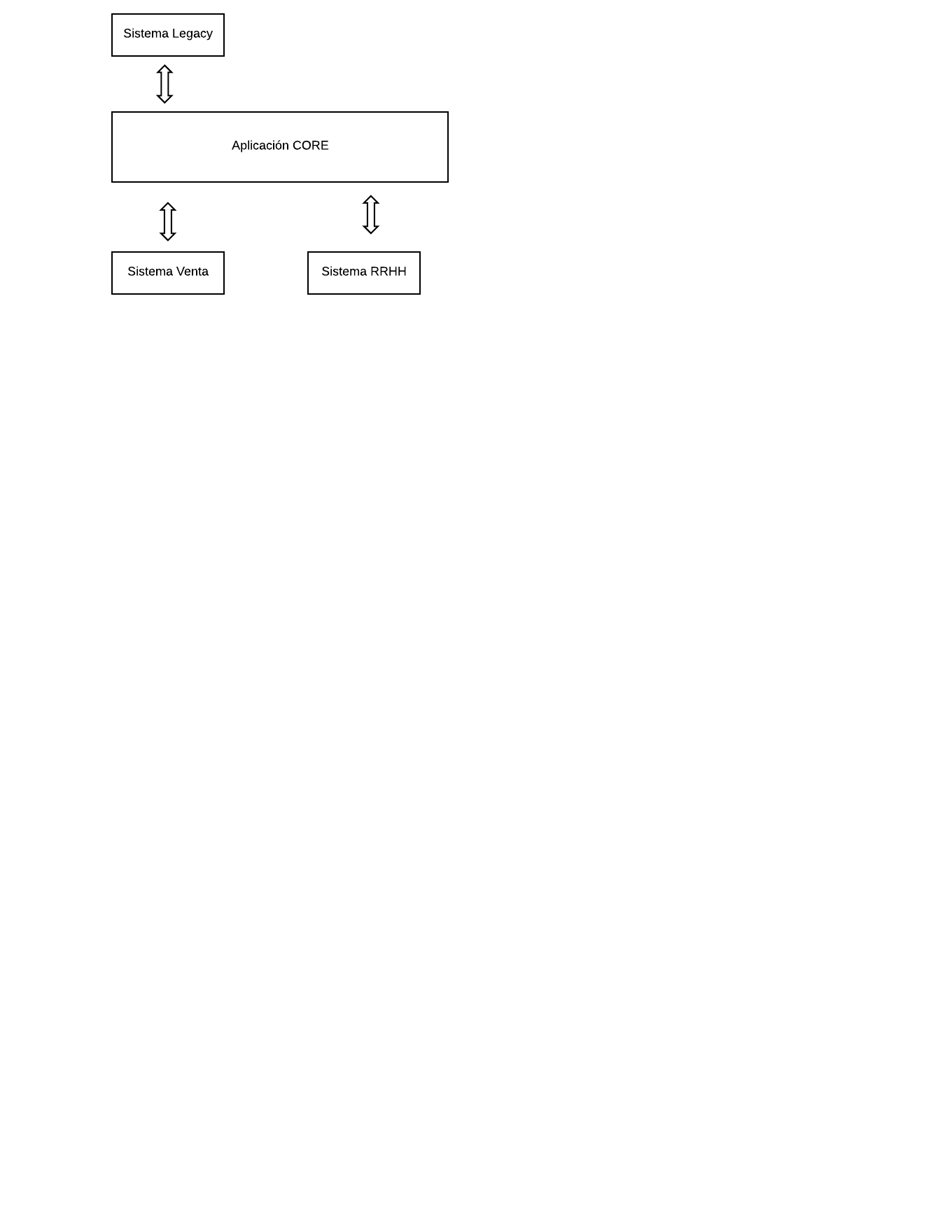
Ejercicio 2

Se pensó en la arquitectura MVC para trabajar los niveles por separador, en el nivel de lógica de negocio se agregara el modulo para realizar el cobro con terminales lectoras NFC y poder a futuro seguir integrando mas módulos, al momento de generarse una nueva transacción esta llegara a la capa de BD donde el motor se encargara de ejecutarla asegurando su atomicidad. El patrón de integración que se usaría sería el de eventos, dado que usa mensajes asincrónicos que evita el bloqueo al momento de generar la petición por parte del origen



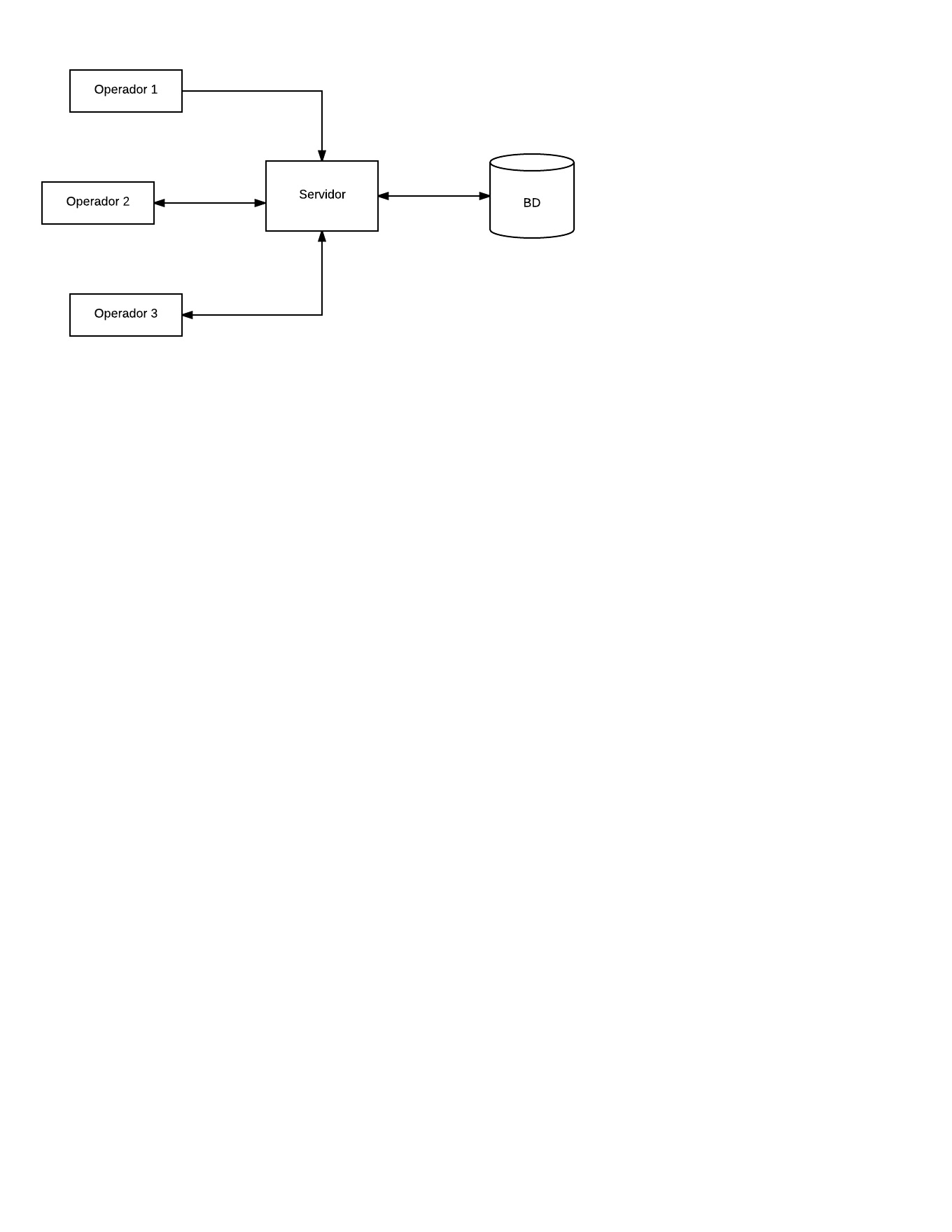
Ejercicio 3

Se pensó en una arquitectura SOA dado que pone foco en la interoperabilidad permitiendo acoplar diferentes sistemas y poder manejar la lógica propia para cada sistema. Se planteo el patrón Call & Return para que se consuman los servicios entre componentes consumidor y proveedor usando SOAP o REST para provechar la comunicación HTTP, la desventaja seria el bloqueo que ocurre hasta que la petición es atendida



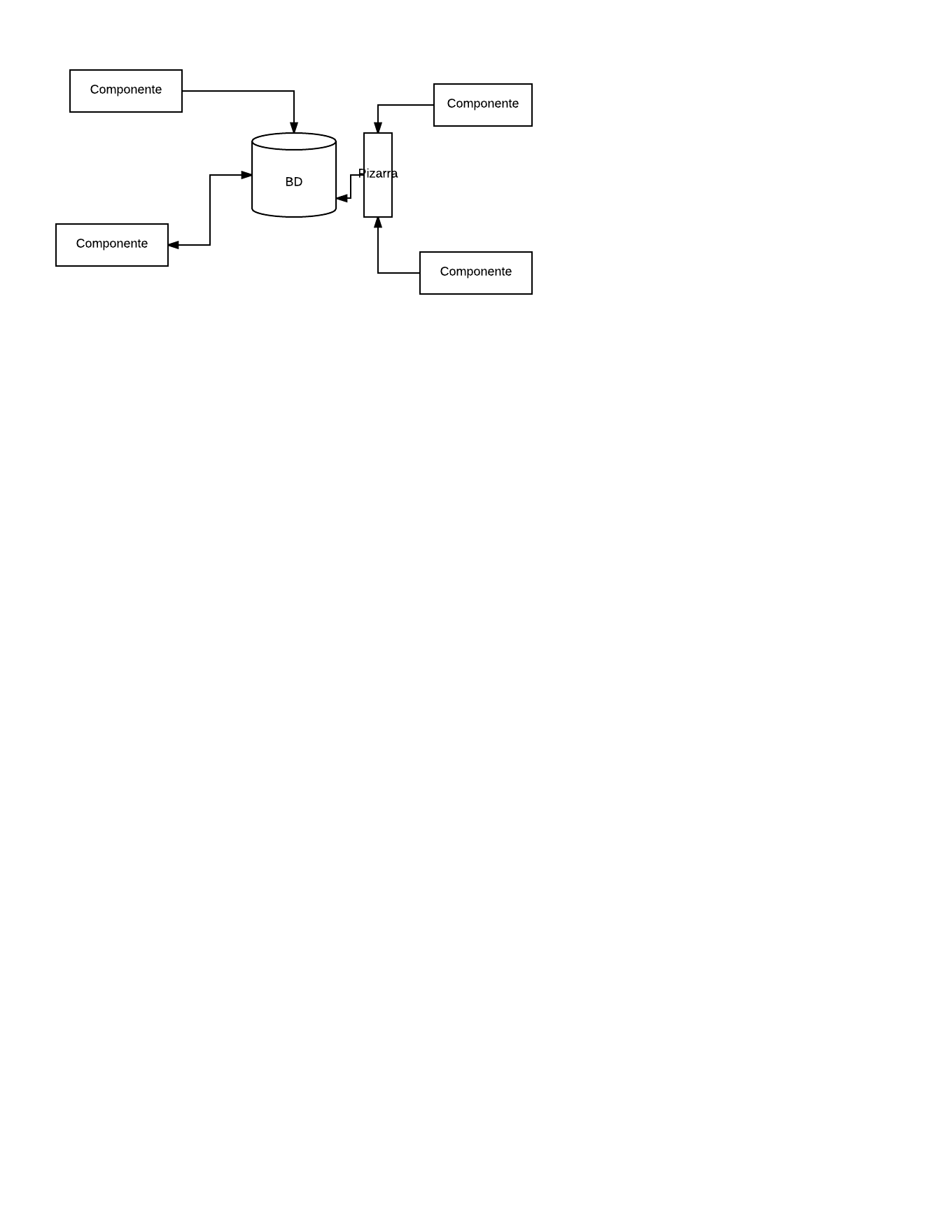
Ejercicio 4

Se pensó en la arquitectura cliente – servidor, dado que es aceptable los bloqueos al momento de la comunicación sincrónica, por ello se podría usar como patrón de integración el de memoria compartida, dado que el mismo maneja un espacio único de memoria donde los bloqueos son manejados con semáforos, un aspecto critico sería el único punto de falla que se genera tanto por la arquitectura cliente servidor como por el patrón de integración aplicado dado que la información está centralizada.



Ejercicio 5

Se pensó en la arquitectura centrada en datos dado la facilidad que ofrece de integración, además se aprovecharía la fácil administración que tiene y se manejaría un patrón de integridad como el de memoria compartida para delegar la responsabilidad del manejo de datos al motor de BD, claro que esto conlleva a tener algunos aspectos críticos como el único punto de falla pero quizá se podría resolver manejando otra BD como respaldo, también para evitar las transacciones excesivas si podría agregar algún componente pizarra donde se necesite tener como entrada la salida de otro componente



Ejercicio 6

Se pensó en la arquitectura de microservicios, donde cada servicio expuesto puede ser atómico o un grupo de servicios más pequeños, con esta arquitectura se puede escalar fácilmente y se da un cierto grado de independencia entre servicios, separando también la BD para tener fuentes aisladas. Como patrón de integración se pude usar Call & Return para el consumo de estos servicios por HTTP sea SOAP o REST pero con la desventaja del bloqueo por parte del consumidor hasta que el proveedor atienda la petición.

