



## DISEÑO DE SISTEMAS

# Trabajo Práctico Anual Guía Práctica N° 4

**Grupo:** 04

**Integrantes:**

- Martín Javier Gauna
- Gustavo Matías Di Peppe
- Gonzalo Giliberti
- Juan Martin Conde
- Matias Vivone

**Fecha de entrega:** 11/11/2018

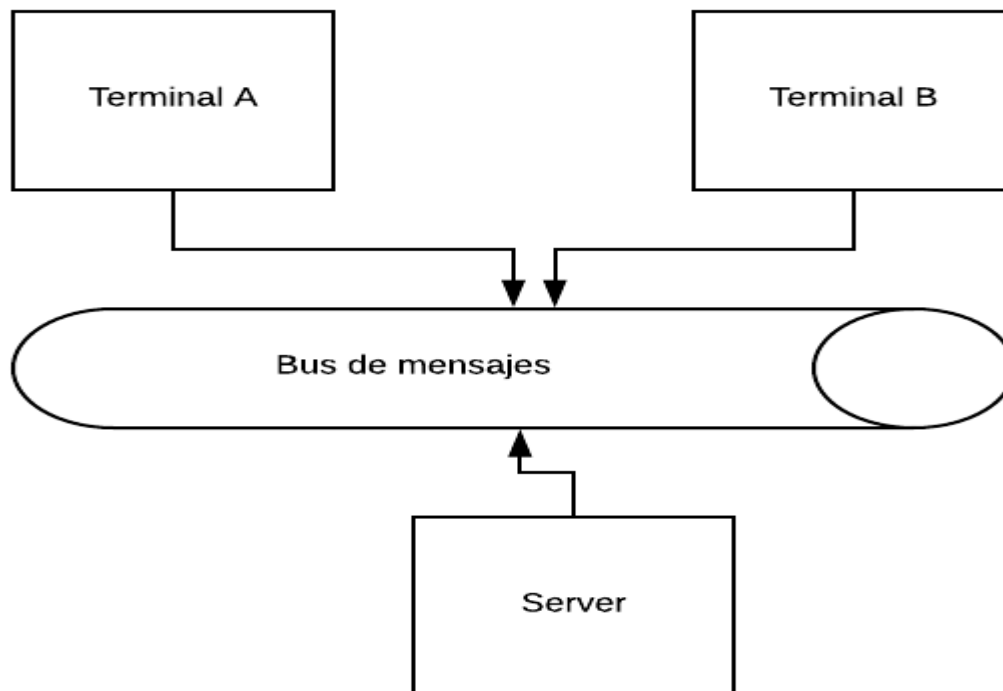
**Profesor:** Martin Agüero

**Ayudante a cargo:** Alejandro Leoz

### Caso 1: NFC

Un proveedor de servicios de cobranza necesita integrar las terminales lectoras NFC ubicadas en las tiendas de sus clientes, con el servidor central. Cada vez que se genera una transacción de compra, la terminal debe transmitir la operación asegurando su atomicidad. Se desea que en ningún momento la terminal quede bloqueada a la espera de una confirmación de recepción por parte del servidor de modo que, además, tolere alta frecuencia de operaciones. Asimismo, la solución debe ofrecer la suficiente flexibilidad para integrar diferentes fabricantes de terminales y soportar múltiples formatos de representación de los datos

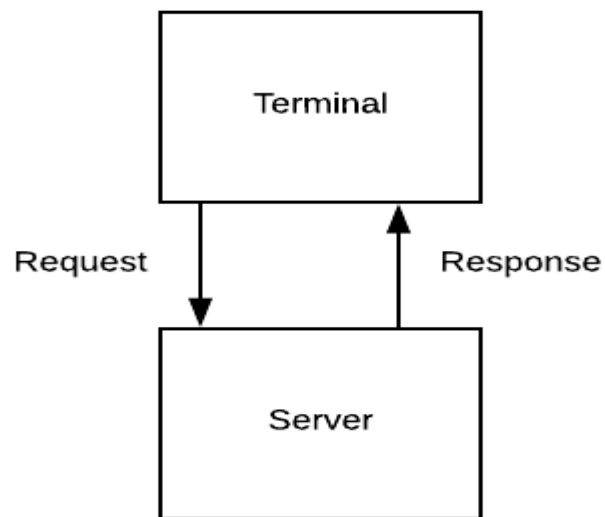
Para este caso hemos decidido utilizar una arquitectura **Cliente-Servidor** el cual se comunica a través de un **bus de mensajes** el cual permitirá que la comunicación sea asíncrona y esto permitirá que la terminal no se quede bloqueada a la espera de una respuesta. La mensajería no está atada a un solo tipo de mensaje, es decir, **el servidor sabrá interpretar un mensaje en formato json, xml, etc.**



## Caso 2: Trafico Aéreo

En la consola de control de tráfico aéreo donde el operador desarrolla su actividad analítica, se visualizan los datos recuperados de un servidor. Dada la criticidad del sistema, es tolerable que ocurran bloqueos en la terminal cuando debe sincronizar la información local con la remota. Al tratarse de una solución integral, donde todo el software es desarrollado por el mismo fabricante, es decir, no es necesario ofrecer representación estandarizada de los datos que se intercambien.

Para este caso también proponemos una estructura **Cliente-Servidor** pero en este caso la comunicación tiene que ser sincrónica debido a la critico del problema que se busca resolver, se utiliza una comunicación del tipo **Call-Return**



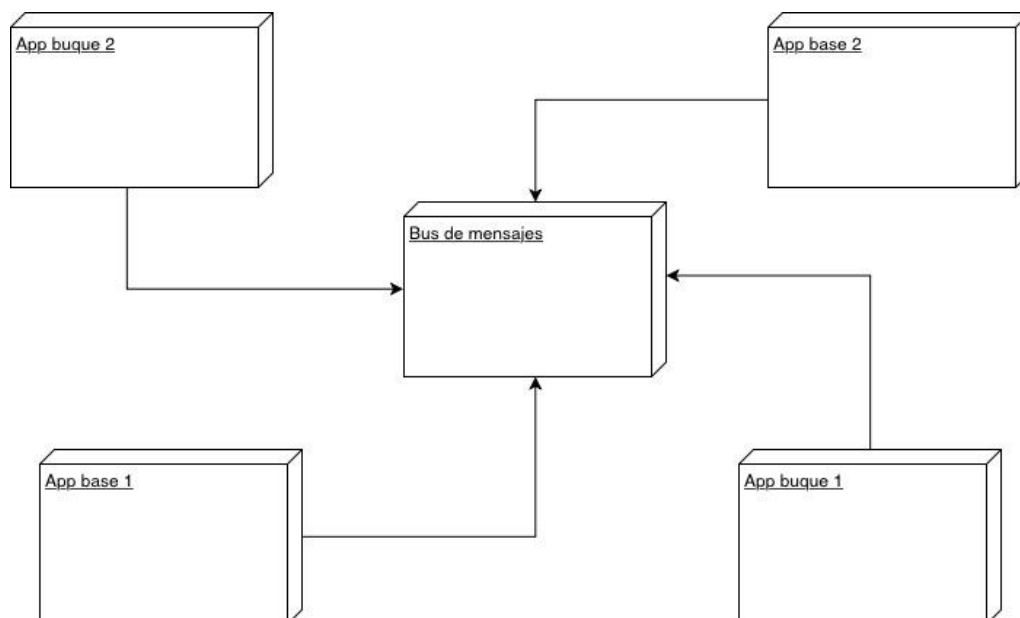
### Caso 3: Marina mercante

Una compañía naviera precisa actualizar el modo como sus buques notifican a la base acerca de las condiciones meteorológicas de la región oceánica que están atravesando. A su vez, el sistema deberá suministrar a cada buque reportes regionales de su interés. Como la calidad del canal de comunicación satelital podría verse afectado por lluvias intensas, es crítico que, tanto en las mediciones [buque → base] como los informes [base → buque], esté garantizada su integridad. También debe soportar gran cantidad de conexiones concurrentes al servidor.

Para resolver este caso hemos elegido un patrón de **BUS de Mensajes** ya que desacopla los sistemas que se están integrando (buques y bases) librándose de la mala calidad de comunicación satelital que podría haber.

Esto se logra a través de un “intermediario”, el BUS de mensajes que gestiona la relación entre los componentes.

Por otro lado, es un patrón de *mensajería asincrónica* lo cual nos brinda un mecanismo de transporte desacoplado, confiable y eficiente, tal como nos pide el enunciado de este caso. Como desventaja, la programación de este modelo es compleja, lo cual hace que la curva de aprendizaje sea mayor.



#### Caso 4: Fintech

Un emprendimiento que surgió como una pequeña iniciativa “de garage” está incrementando su cartera de clientes. El núcleo de su actividad consiste en suministrar información a sistemas de inversores acerca del desempeño histórico de diferentes compañías que cotizan en la bolsa. Como en algunos casos estos servicios de información son personalizados, es restricción obligatoria (mandatory) que las fuentes estén aisladas entre sí (al menos desde un enfoque lógico). En algunos casos, estos informes pueden ser una agregación de servicios propios o de terceros con diferentes formatos de representación. Es deseable que la solución permita alta flexibilidad en función de cambios en los requerimientos de los clientes y a su vez soporte escalabilidad.

Más info: <https://www.pwc.com/us/en/financial-services/publications/viewpoints/assets/pwc-fsi-what-is-fintech.pdf>

Para este caso hemos elegido el patrón de “Memoria compartida distribuida” ya que emplea una cola de mensajes que encapsula la información a compartir, acorde al pedido del caso de que las fuentes estén aisladas entre sí. Por otro lado, este patrón nos proporciona escalabilidad, es decir que podemos extender el sistema sin mucho problema, tal como el caso lo requiere.

Como desventaja la topología de red muy importante y se requerirán altos recursos de administración de la red. Sin embargo, por ser un emprendimiento fintech, se espera que cuenten con el *know-how* y la inversión necesaria para esto.

