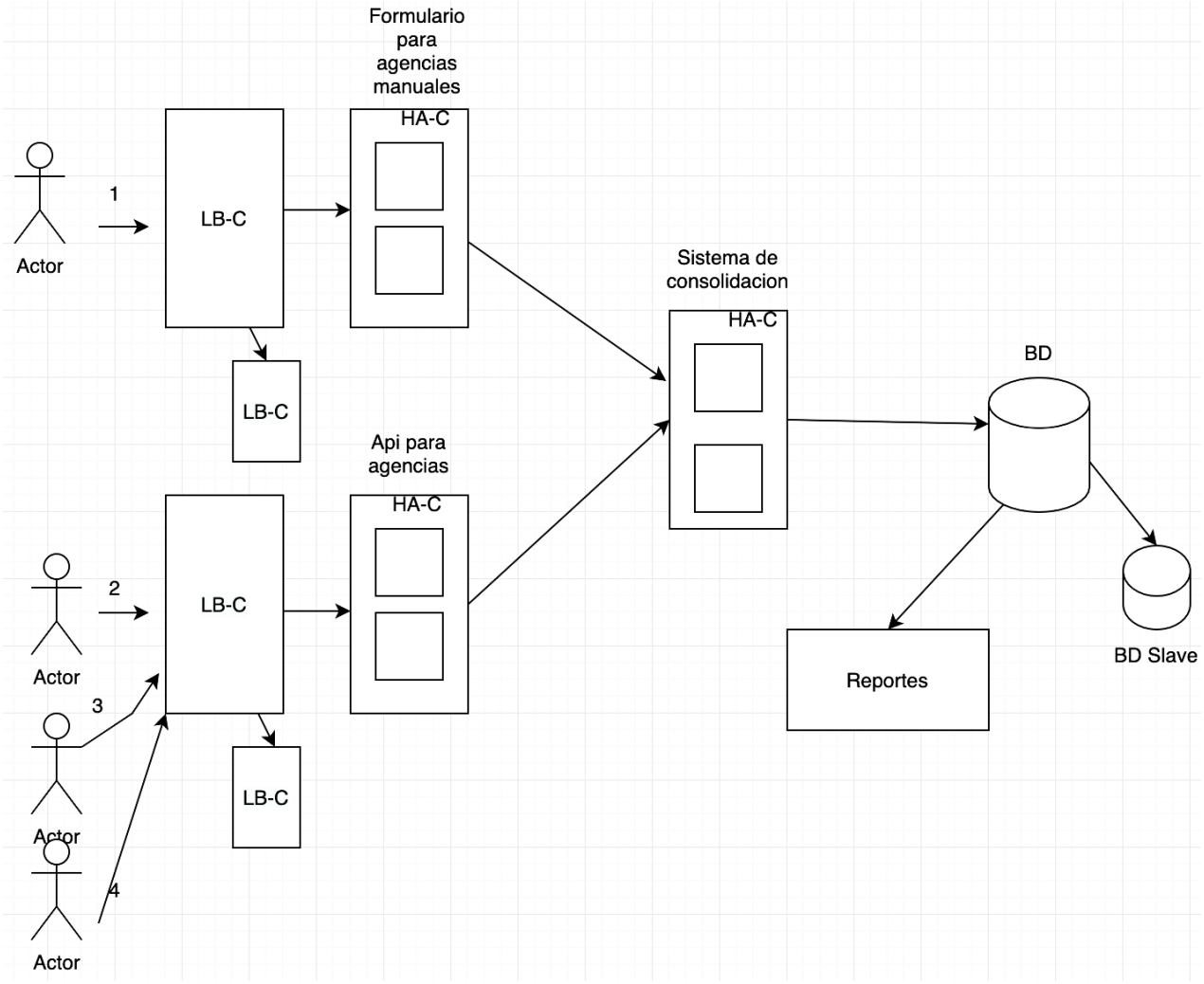
ARQUITECTURA

EJERCICIO 1 – Agencia de Seguridad

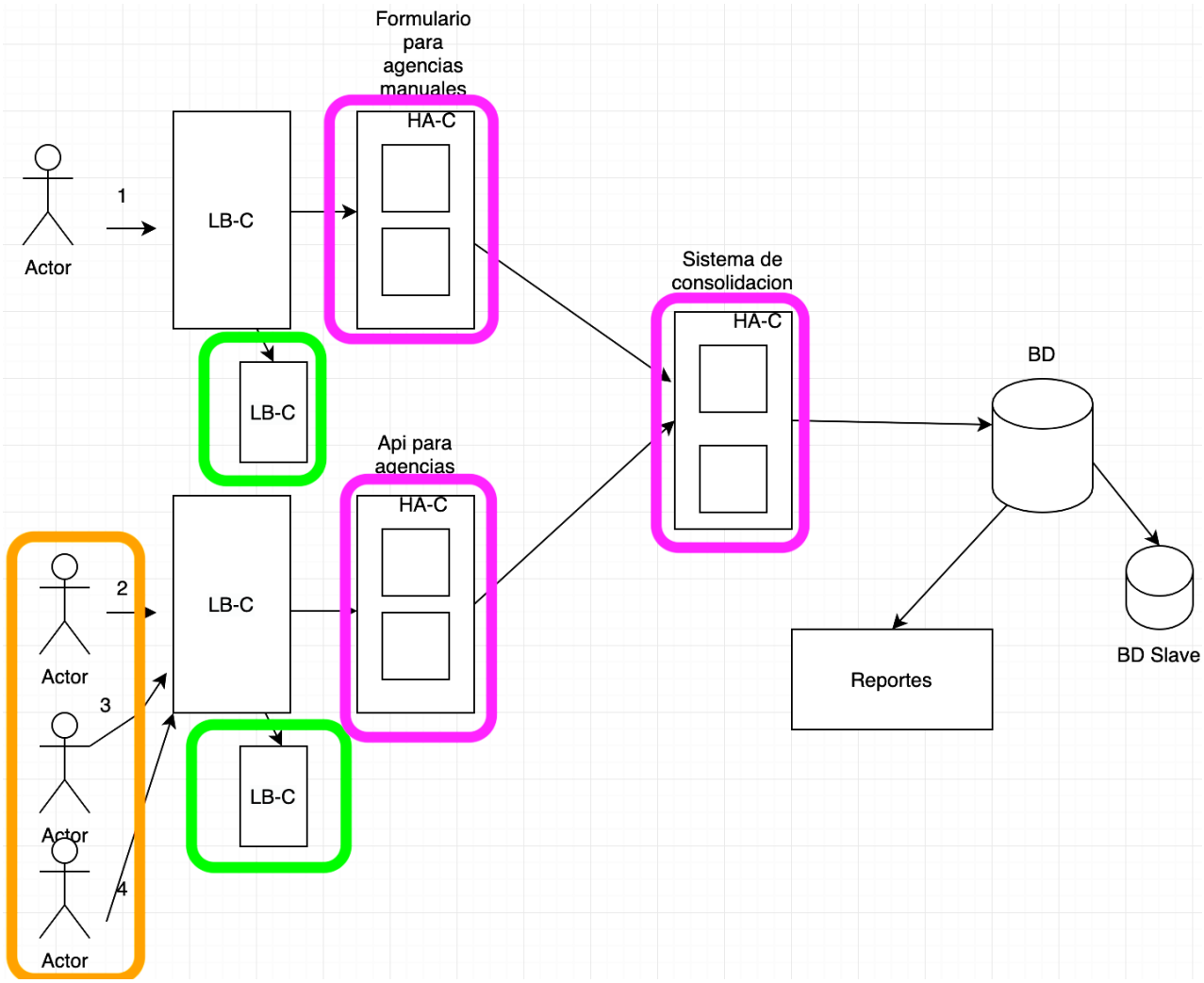
PROPUESTA #1



Lo que hicimos fue abarcar los 4 puntos mencionados de carga de datos.

1) Las agencias ingresan los datos mediante un formulario online, el cual tiene como estructura un **LB-C** por si hay varias peticiones a la vez (tiene su resguardo por si se cae) que se conectan con un **HA-C** con varios nodos de aplicación para dar disponibilidad. Los nodos de aplicación sería un *front* que luego conectan con el **Sistema de Consolidación** para el proceso de los datos y luego impactar en la **base de datos**.
2) Se expone una **API** para que las agencias puedan enviar información de forma automática y esta información luego sea procesada por el **Sistema de Consolidación** e impacte en la **base de datos**.
3) Envío de información por correo electrónico, subida del archivo exportado.
4) Subida de información de los dos formatos relevados.

PROPUESTA #1 – RESPUESTA Y CORRECCIÓN



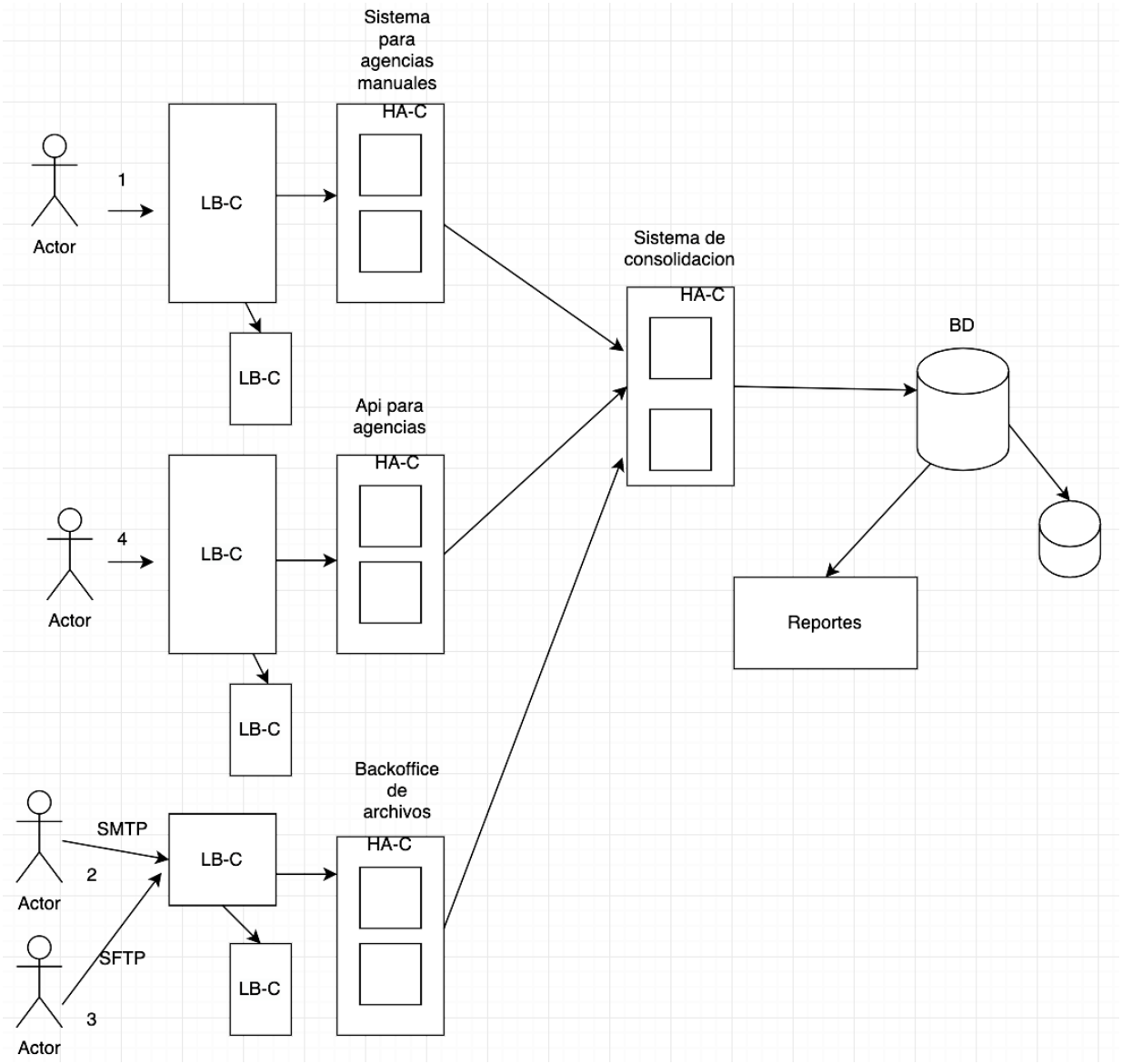
I. No queda claro qué representa cada **LB** pequeño (marcado en verde sobre la imagen).

II. De la forma que se plantea, detrás del **LB** hay UN ***cluster* de alta disponibilidad** (marcado en fucsia sobre la imagen), con lo cual no hay balanceo de cargas (todas las peticiones van al mismo lugar).

III. ¿Dónde se visualiza el tratamiento propio del caso 2 (mail con archivo adjunto que termina en un servidor de mail)? Los casos 2, 3 y 4 parecen tener el mismo tratamiento (marcado en naranja sobre la imagen).

IV. ¿Dónde se visualiza el tratamiento propio del caso 3 (generación automática del reporte que se deja en una carpeta de un sistema de archivos)? Los casos 2, 3 y 4 parecen tener el mismo tratamiento (marcado en naranja sobre la imagen).

PROPUESTA #2



- Respecto del **[I]**, agregamos los **balanceadores de carga** (marcados en verde) por si se llegaran a caer los **balanceadores de carga** principales. La idea es evitar un punto único de falla. ¿Nos podría indicar como se puede representar mediante un gráfico?
- Respecto del **[II]**, tenemos una duda: ¿habría que agregar varios **HA-C** "apilados" (como quien dice) de manera que no haya solamente un único **HA-C**? Lo que quisimos representar es que dentro del **HA-C** hay varios nodos de aplicación (en este caso solo dibujamos 2 pero podrían ser más para tener mayor redundancia/disponibilidad.
- Respecto del **[III]**. Caso 2: Lo que queremos representar es que el actor 2 envía el mail con el adjunto y luego un agente nuestro carga la información en nuestro ***backoffice*** de agencia. Luego este envía la información al **Sistema de Consolidación**.
- Respecto del **[IV]**. Caso 3: Lo que queremos representar es que el actor 3 deja el reporte en una carpeta de un **sistema de archivos** y luego un agente nuestro carga la información en nuestro ***backoffice*** de agencia. Luego este envía la información al **Sistema de Consolidación**.

PROPUESTA #2 – RESPUESTA Y CORRECCIÓN

- *Respecto del **[I]**…*

  Pongan los **balanceadores de carga** en alta disponibilidad (vimos un ejemplo en clase) representándolos como lo hicieron en otras partes del mismo diagrama.

- *Respecto del **[II]**…*

  Lo primero que habría que preguntarse es si se espera tener picos de recepción de información de las agencias que justifiquen la presencia de un **balanceador**. Si este fuera el caso, pueden poner el **balanceador** en alta disponibilidad y detrás de ellos dos o más **servidores** que procesen. Pero insisto con la pregunta: ¿no alcanza con un **servidor de procesamiento** eventualmente protegido por un **HA-C**?

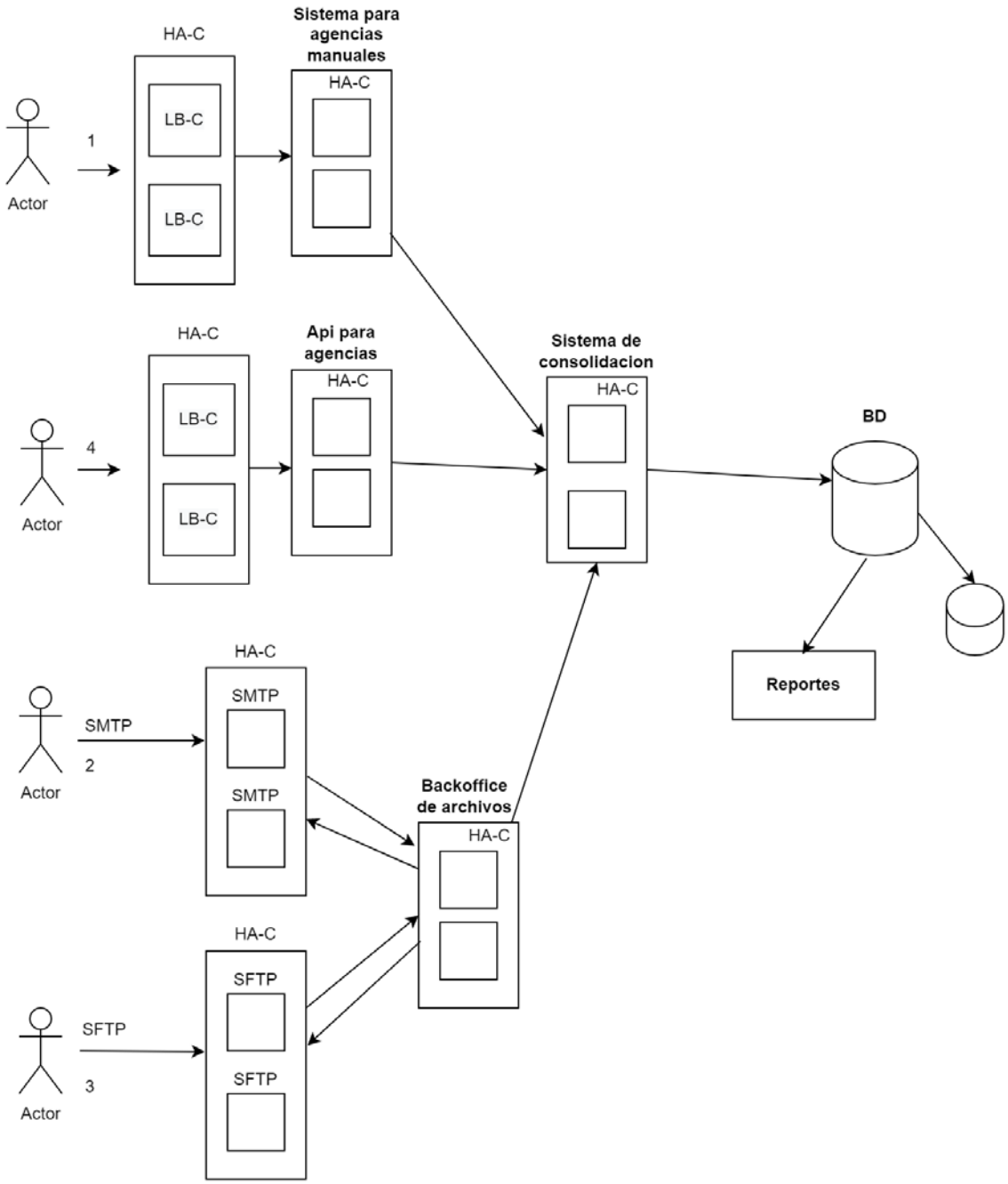
- *Respecto del **[III]**. Caso 2…*

  En el diagrama el **balanceador** parece recibir el mail. Un tratamiento posible es que el mail sea recibido en un **servidor de mail** y que el agente interactúe con él para obtener los archivos adjuntos que es necesario procesar.

- *Respecto del **[IV]**. Caso 3…*

  Entonces habrá un **file server** donde llegan los archivos vía SFTP (sin **balanceador**). Luego el agente deberá tomar de allí los archivos.

PROPUESTA #3



- Para los casos **[1]** y **[4]**, se ubicaron los ***load balancer*** con **HA-C**.
- Para los casos **[2]** y **[3]**, se eliminaron los ***load balancer*** (consideramos que no habrá demasiadas solicitudes que ameriten su presencia) y se agregaron los **servidores de procesamiento** (de **SMTP** y **SFTP**, respectivamente) con **HA-C**.

  Ambos ***clusters*** recibirán mensajes/archivos y estos serán consumidos desde el ***backoffice*** donde se mostrará la estructura original (habrá un proceso que se dispara desde ***backoffice*** para poder recuperar los datos).

  Luego, esta información será enviada en forma uniforme al **Sistema de Consolidación** para que, luego de ser debidamente tratada junto con la otra información proveniente de las otras fuentes (casos **[1]** y **[4]**), sea persistida en la **base de datos**.

PROPUESTA #3 – RESPUESTA Y CORRECCIÓN

- Respecto de los casos 1 y 4, el uso de un **balanceador de carga** requiere que existan dos o más nodos sobre los cuales se repartirá la carga de trabajo. Si detrás del **balanceador** solo hay un ***cluster* de alta disponibilidad** (con varios nodos atendiendo la misma petición) no hay carga que balancear: todo va a un solo lugar (**HA-C** actúa como una unidad).

- En referencia a la **base de datos**, habría que aclarar si lo que se muestra es un esquema de replicación u otra cosa.

- El módulo de **reportes** no es consumido por nadie y el intercambio con la **base de datos** (su motor) debería ser bidireccional.

- Ajustados los aspectos anteriores, habría que hacer algunas consideraciones sobre la infraestructura de red y seguridad.