Servicio de gestión de vistas

Eduardo Gimeno Soriano

Sergio Álvarez Peiro

4 de diciembre de 2019

Introducción

La práctica consiste en implementar un sistema con tolerancia a fallos mediante el uso de replicación primario-copia. Se va a implementar como un servidor gestor de vistas que atiende a los clientes que actúan como servidor primario, copias o en espera. El gestor de vistas no está replicado, por lo que podría fallar. Para controlar el comportamiento entre las copias los nodos se comunicarán mediante el envío de latidos al gestor de vistas.

Sistema de tolerancia a fallos primario-copia

El servidor gestor de vistas guarda un estado para poder gestionar y mandarles las vistas a todos los clientes. Una vista se compone por un identificador numérico, el nombre del nodo primario y la lista de nombres de los nodos copia. En caso de que no esté definido, los nombres tomarán valor :undefined.

El gestor tiene dos vistas guardadas simultáneamente: la vista válida que es la que se envía a los clientes cuando preguntan y es la que considera como la actual. La segunda vista es la previa a convertirse en la nueva válida, la vista tentativa. La vista tentativa se convierte en la nueva vista válida cuando el nodo que es considerado primario confirma la vista tentativa enviada con el latido anterior.

El resto del estado que guarda el servidor es la lista de latidos que se reciben de los clientes. Cada latido contiene el número que identifica la última vista conocida por el cliente y el nombre del nodo emisor. Por último una variable booleana consistencia, que será true siempre que no se caigan el nodo primario y copia o el servidor primario caiga sin confirmar la vista tentativa. Esta variable indica que se mantiene la consistencia del sistema.

Inicialización del sistema

En un principio el servidor no sabe nada de ningún nodo, ambas vistas tendrán todo en :undefined. Cuando le llega un mensaje se pondrá ese nodo como primario en la tentativa, hasta que se confirme como primario. Durante ese periodo llegan latidos también de otros nodos, que se pueden poner como nodos copia en la tentativa. El resto de nodos, se añadirán a la espera.

Ejecución del sistema normal

Esta situación es cuando no se producen fallos. En este caso los nodos recibirán siempre del gestor el mismo nodo primario. Las dos vistas gestionadas por el servidor serán la misma. Si se llega a recibir un latido de un nodo nuevo, se añadirá a espera.

Situación de caída de copia

Esta situación se produce si un nodo establecido como copia no envía el latido 4 veces. En ese caso el servidor promociona a uno de los nodos de espera a nodo copia. Si no hay nodos en espera, tendrá que considerarse la copia como :undefined. Cuando ocurre esto el nodo primario y la copia no serán consistentes por lo que no se confirmará la vista hasta que la transferencia de datos finalice y la consistencia vuelva.

Situación de caída de primario

Si se produce la caída de la misma forma que en la copia pero del nodo primario, el servidor promociona a un nodo copia a nodo primario y a su vez uno de espera a copia. La copia si era consistente con el nodo primario, por lo que la promoción puede darse sin problemas, pero no la nueva copia. No se confirmará la vista hasta que la transferencia de datos finalice y la consistencia vuelva. Se puede volver a relanzar el nodo primario caído, que se añadiría al a espera.

A continuación se muestran 4 diagramas de secuencia para las situaciones descritas (suponiendo Nodo1 como primario y los otros dos como copia):

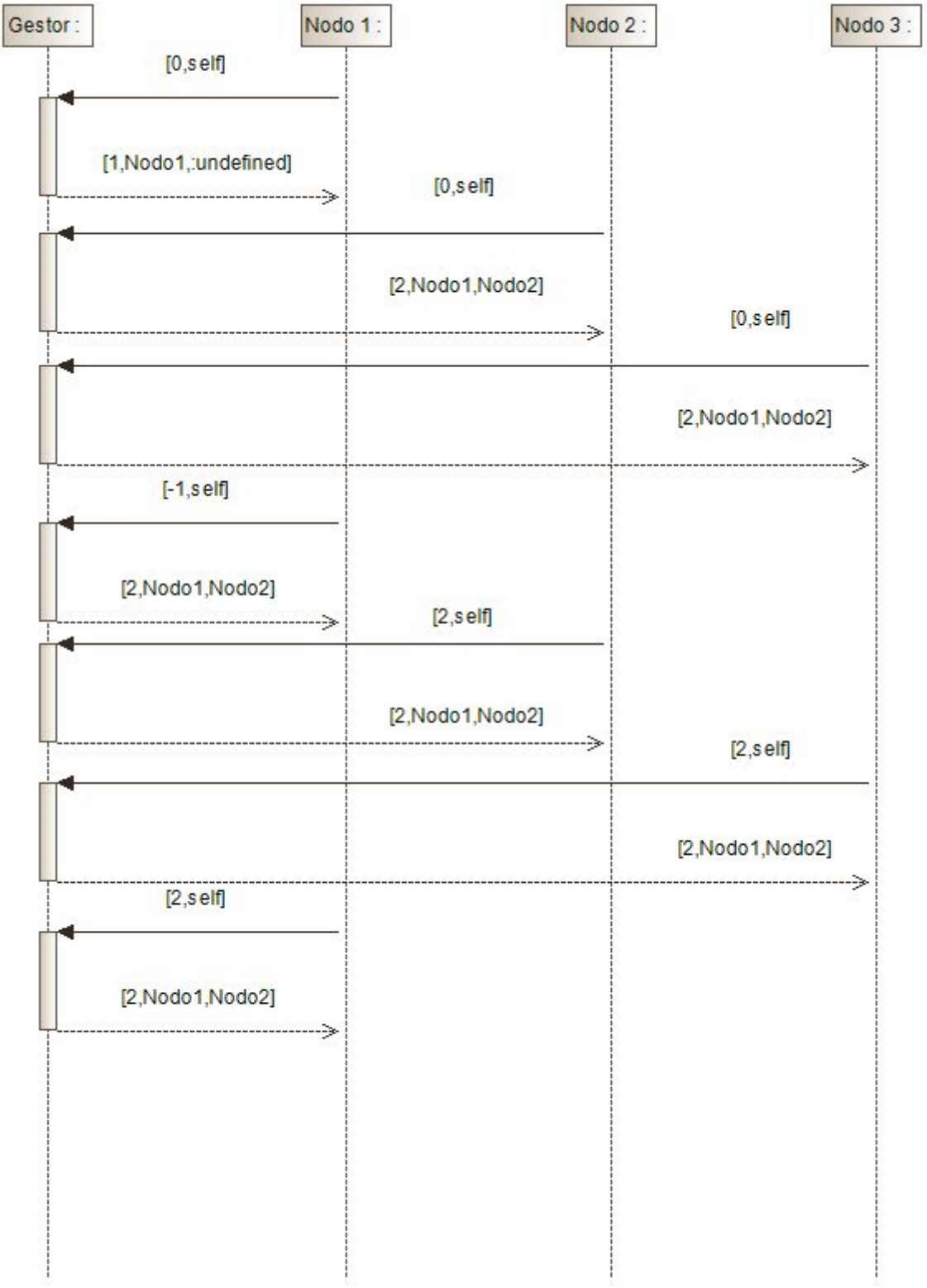

Figura 1. Diagrama de secuencia inicialización del sistema.

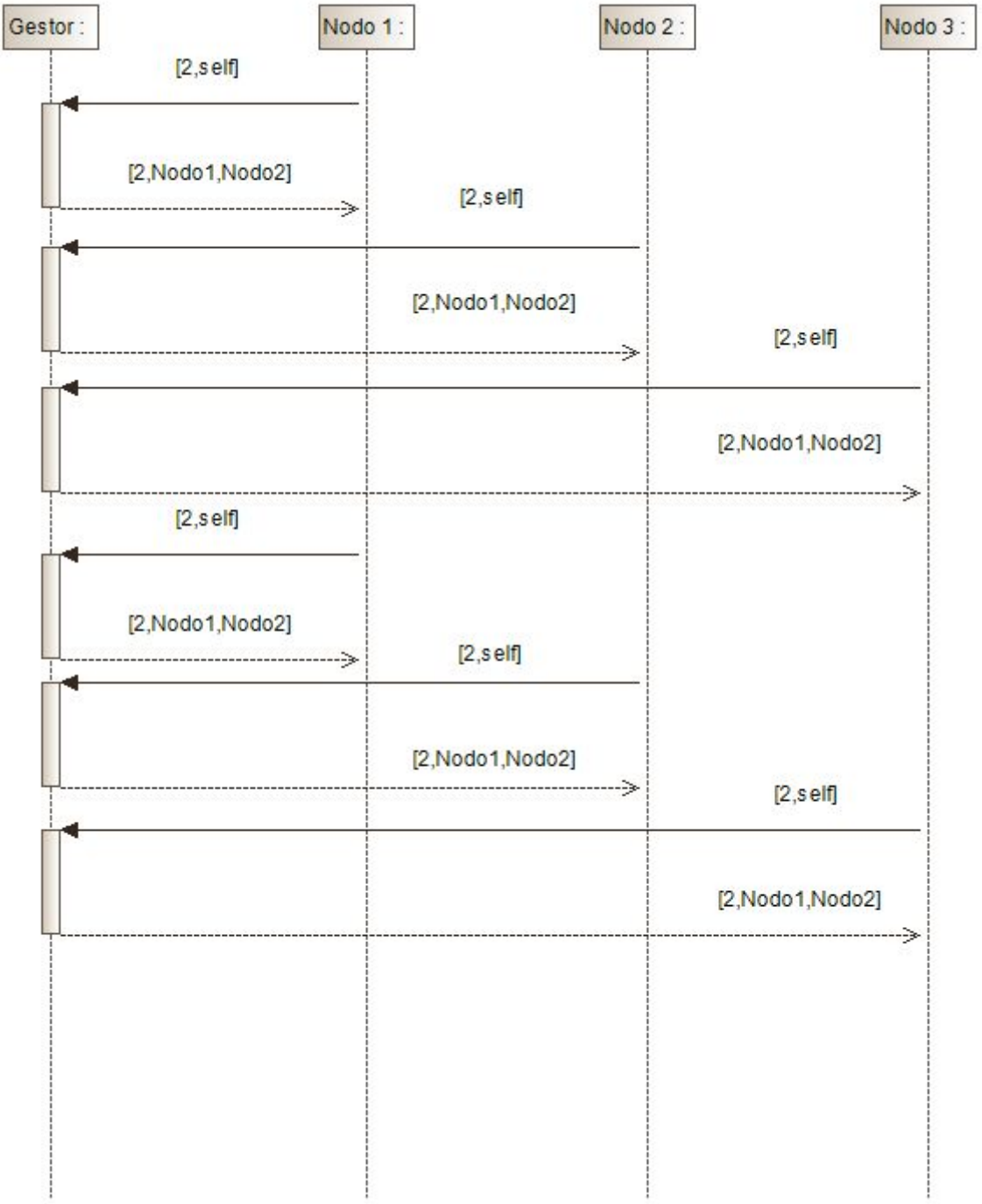

Figura 2. Diagrama de secuencia de la ejecución del sistema.

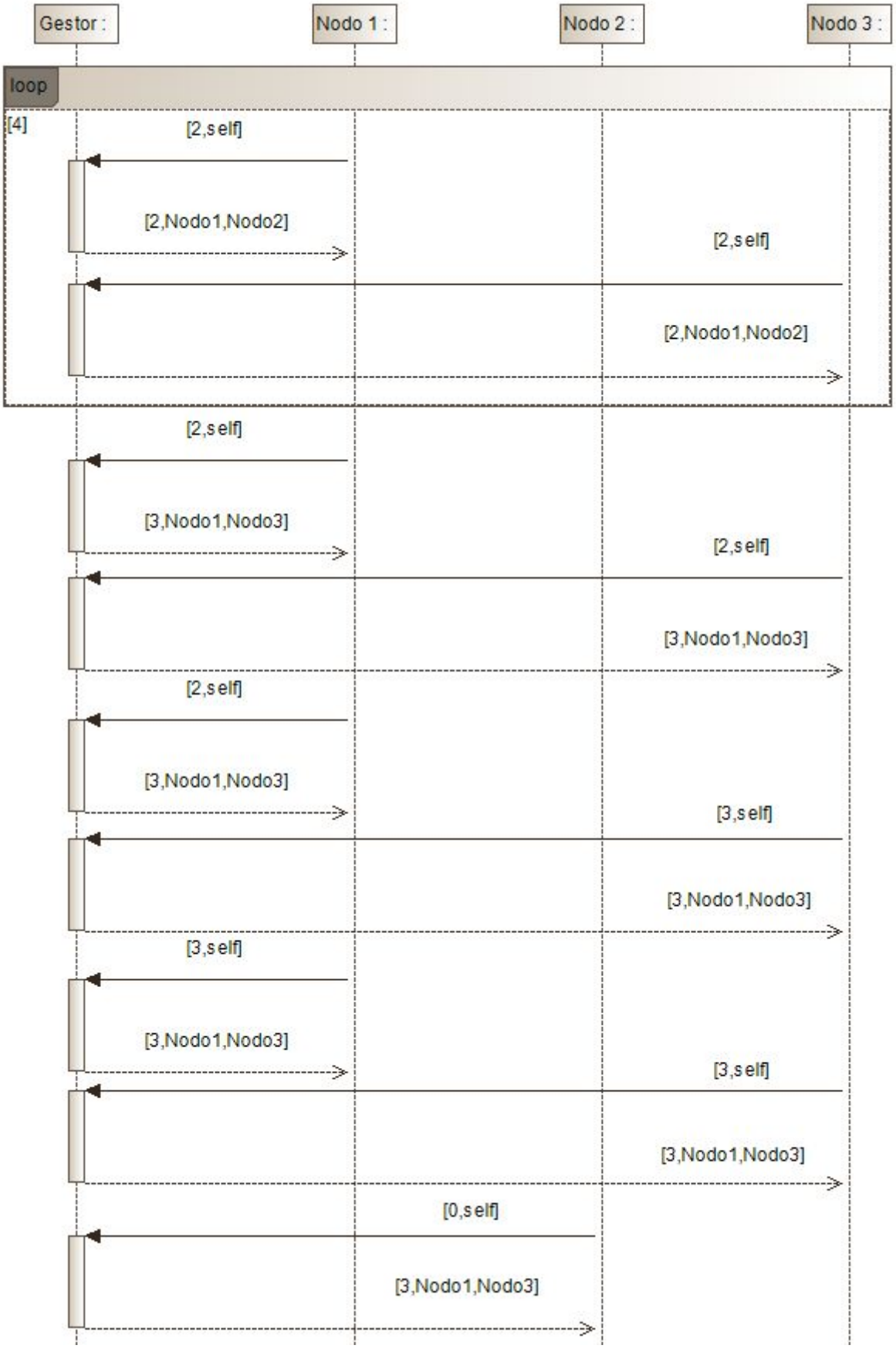

Figura 3. Diagrama de secuencia fallo de copia.

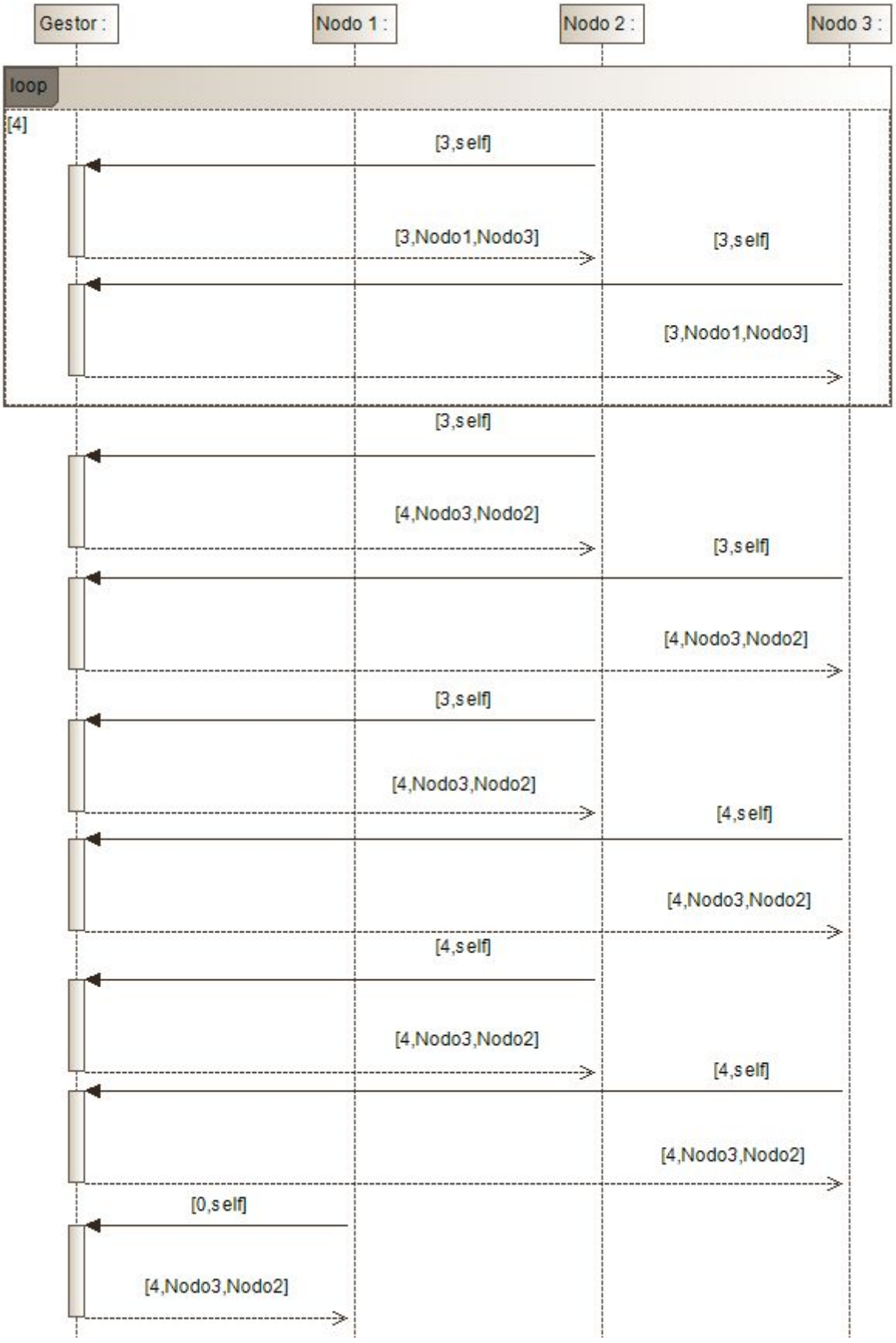

Figura 4. Diagrama de secuencia fallo de primario.

Validación del sistema

Para validar el correcto funcionamiento del sistema, se realizan 9 test tanto como en local como con los servidores en máquinas distintas:

1. No hay nodo primario al iniciar.
2. Hay un nodo primario con vista correcta.
3. Hay un nodo copia.
4. El gestor promociona a la copia si el primario falla.
5. El servidor rearrancado se convierte en nodo copia.
6. El servidor en espera se convierte en nodo copia si el primario falla.
7. El nodo primario rearrancado se trata como caido y se convierte en nodo en espera.
8. El servidor no recibe la confirmación de la vista del nodo primario.
9. Si el servidor anterior se cae, un nuevo servidor no inicializado no puede convertirse en nodo primario.

El resultado de la ejecución de validar_servicios_vistas ejecuta los tests listados e informa por pantalla que se han superado todos.

Conclusiones

El permitir tolerancia a fallos en un sistema distribuido es complejo en un sistema como es el de primario-copia. Hay que tener en cuenta que clase de fallos se van a dar y cuándo pueden ocurrir afectando al resto de la arquitectura del sistema. Además se tiene que tener en cuenta cuando se diseña el sistema si se prefiere tener disponibilidad con la tolerancia a fallos o priorizar la consistencia ya que no se puede tener todo normalmente.

Bibliografía

[1] Material de la asignatura de Sistemas Distribuidos del grado de Ingeniería Informática de UNIZAR.

[2] Documentación del lenguaje Elixir https://elixir-lang.org/docs.html