

# Práctica 3. Tolerancia a fallos

Jorge Aznar López

Jorge Fernández Muñoz

22 de Noviembre de 2018

# Introducción

Esta práctica tiene como objetivo plantear una solución a un sistema máster-worker con fallos en sus trabajadores, los cuales realizan la tarea de encontrar los divisores de un número (tipo uno), sumarlos (tipo tres) o ambas (tipo dos), dependiendo del tipo de trabajador, con la finalidad de proporcionar esos datos al máster cuando los solicite a través de una petición con la finalidad de encontrar los números amigos del 1 al 1000000.

El máster primero solicita a los trabajadores de tipo dos que realicen el cálculo de la lista y la suma a la vez pues es lo más eficiente, y en caso de que todos los trabajadores de tipo dos fallen manda el trabajo a los trabajadores de tipo uno y tres de manera secuencial, primero el cálculo de la lista (uno) y luego la suma (tres).

La solución a los fallos de caída, omisión o retraso se basa en encolar los distintos tipos de trabajadores e ir solicitando los datos necesarios al siguiente si el actual cae (se considerará caído cuando no responda a la petición después de un timeout preestablecido por el máster). Por supuesto, problemas indirectos producidos por el fallo se explicarán con más detalle posteriormente.

## Tratamiento de los fallos (máster)

Los fallos que se pueden producir abarcan la caída del trabajador, el cual no responderá más a las peticiones del máster, la omisión en el envío del mensaje de manera ocasional o el retraso en el propio envío. Los tres tipos de fallos son gestionados encolando a los trabajadores activos, los cuales una vez fallan son retirados de la lista y se procede a enviar la petición al siguiente trabajador disponible en la lista. Se explican los problemas causados de manera indirecta por separado.

### Crash

El fallo de caída es el que menos problemas ocasionaría, pues el trabajador no llega nunca a enviar la respuesta, por lo cual después de un tiempo de espera (timeout) se procede a enviar la petición al siguiente trabajador disponible.

### Timing

El fallo por retraso en el envío produce que el máster reciba una repuesta que no se corresponde con la petición que el ha generado a otro trabajador, nuestra solución propone comprobar que el resultado esperado coincide con el recibido, si el resultado es correcto se tratará de manera normal para el cálculo de los números amigos con el resultado recibido.

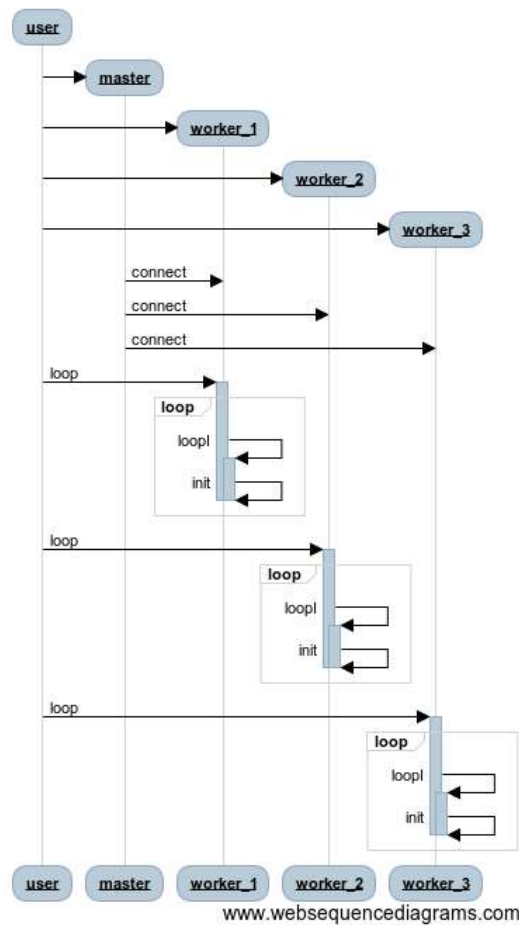
### Omission

Un fallo de omisión se gestiona como uno de caída, ese trabajador pasa a ser considerado inactivo en cuanto deja de enviar mensajes al máster, es decir, mientras el trabajador responda a las peticiones será considerado un trabajador activo.

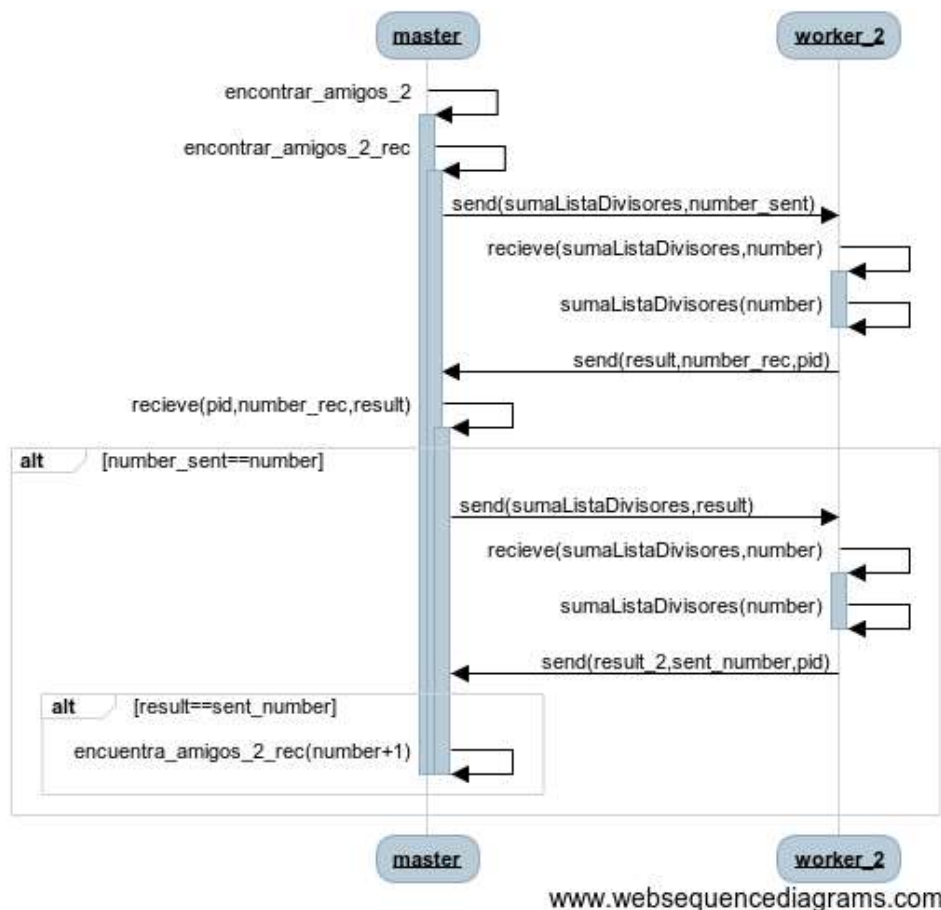
## Diagrama secuencial del sistema

Se ha diseñado un diagrama secuencial que explica visualmente el funcionamiento del sistema, el diagrama abarca desde la conexión por parte del máster y los trabajadores hasta el final de su funcionamiento. Se ha obviado la producción de fallos pues las herramientas utilizadas para generar los diagramas no permitían reflejar un comportamiento tan complejo como una caída o una probabilidad de omisión o retraso en el mensaje. Asimismo, se ha optado por separar el comportamiento del máster con los trabajadores y los uno y tres.

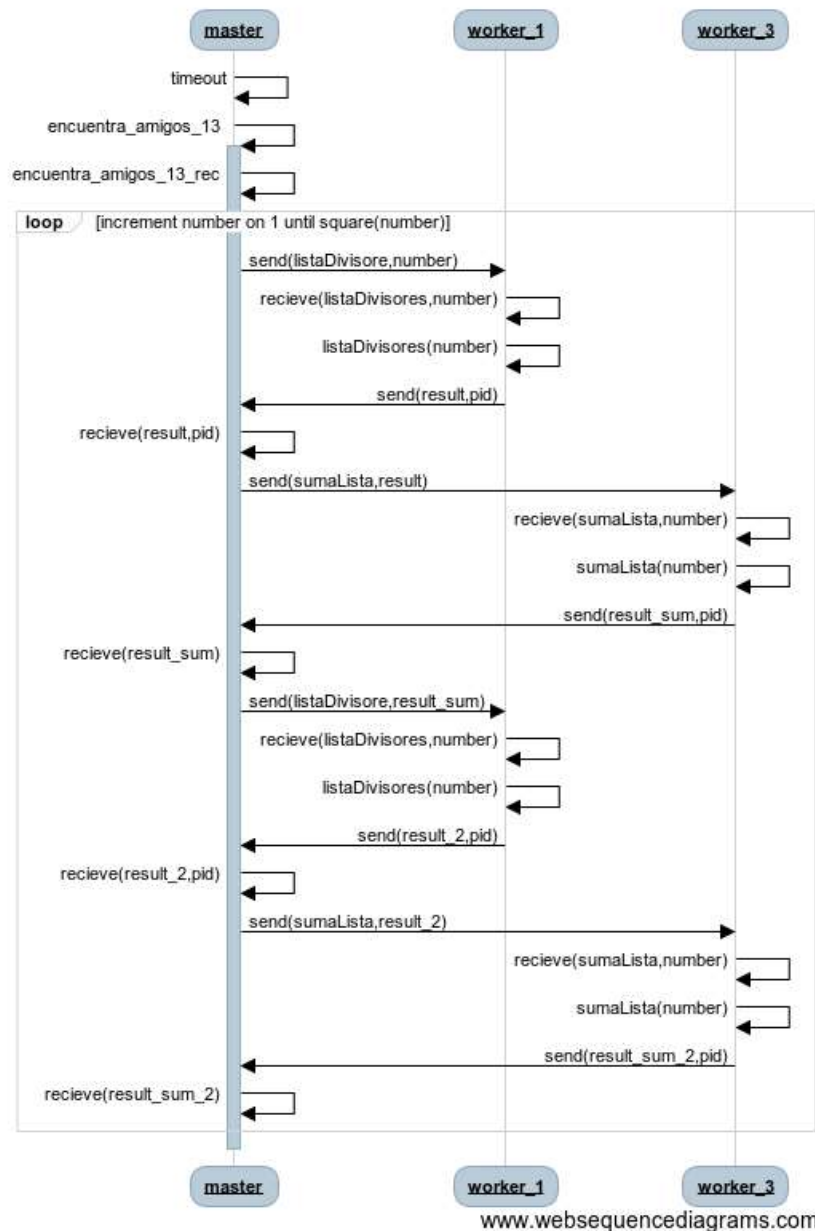
## Process Creation and Connection



## Process Communication



## Process Communication



## Validación Experimental

La validación del funcionamiento del sistema se ha realizado con el máster y cuatro trabajadores de tipo dos y tres de tipo uno y tres. La prueba resultó en que todos los trabajadores fallaban en algún momento y por lo tanto eran excluidos de la lista de trabajadores activos. Por lo cual sería necesario ampliar la cantidad de trabajadores para lograr que alguno trabajara de manera constante sin errores. Sin embargo, se procedió a comprobar que el sistema funcionaba correctamente si se dejaba un trabajador sin fallos de cada tipo por separado (en el caso del uno y tres ambos juntos).

## Conclusiones

Se concluyó que el sistema funciona correctamente con los fallos producidos, sin embargo, a la hora de ejecutarlo con trabajadores con posibilidad de fallo se necesita una gran cantidad de trabajadores para garantizar que el sistema funcionará correctamente.

Una solución alternativa comprendería considerar los trabajadores con fallos de tipo retraso como trabajadores activos cuando envían su respuesta encolándolos de nuevo al final de la lista de trabajadores, esta solución evitaría que la posibilidad de fallo total del sistema fuera menor pero añadiría un mayor retraso al cálculo total de números amigos.