

Práctica 3. Tolerancia a Fallos

1. Objetivos y Requisitos

Esta práctica tiene como objetivo el diseño y la implementación de un sistema distribuido con técnicas de tolerancia a fallos.

1.1 Objetivo

- Familiarización con técnicas de tolerancia a fallos
- Implementación de tolerancia a fallos de tipo *crash*, *omission* y *timing* en una arquitectura *master worker*, tanto en el máster como en el *worker*

1.2 Requisitos

- Elixir y su entorno de desarrollo
- Seguir la guía de codificación de Elixir publicada aquí:

https://github.com/christopheradams/elixir_style_guide/blob/master/README.md

2. Descripción del Problema

Dado el intervalo $[1, 1.000.000]$, se pide encontrar los pares de números amigos contenidos en él. Para ello, se utilizará una arquitectura Master-Worker. No obstante, la ejecución de los *workers* está sujeta a la presencia de fallos de tipo *crash*, *omission* y *timing*.

Dado un número natural n , los *divisores propios* de n son todos los divisores de n salvo sí mismo. Así, por ejemplo, los divisores propios de 6 son 1, 2 y 3.

Dados dos números naturales a y b , se dice que a y b son amigos, si la suma de los divisores propios de a es igual a b y la suma de los divisores propios de b es igual a a .

3. Ejercicio

Se pide implementar un sistema distribuido en Elixir, siguiendo una arquitectura *master-worker*, que permita al *master* encontrar todos los números amigos en el intervalo $[1, 1.000.000]$ aun cuando los *workers* en ejecución presenten fallos de tipo *crash*, *omission* y *timing*. Tendremos además 3 tipos de *workers*: i) un *worker* que dado un número natural n devuelve sus divisores, ii) un *worker* que dado un número natural n devuelve la suma de los divisores propios de n y iii) un *worker* que dada una lista de números calcula la suma de los números contenidos en ella.

Deberéis implementar el máster y los 3 tipos de *workers*. Los fallos los simularemos con el código del *worker* que se adjunta.

Como máximo, se podrá obtener una calificación de A en la solución del problema si se aplican estrategias y mecanismos de tolerancia a fallo en el *master*. Para obtener la calificación máxima A+ en la solución del problema, deberán aplicarse mecanismos de tolerancia a fallos en los *workers* (por ejemplo, el algoritmo del matón).

Respecto de la tolerancia a fallos, podéis utilizar las estrategias que deseéis. Una posibilidad es utilizar los algoritmos 1 o 2 de detección de fallos vistos en clase para el *master*. En cuanto a un *worker*, podríamos utilizar el algoritmo del matón de la siguiente manera. Supongamos que el *master* tiene a su disposición N *workers* w_1, w_2, \dots, w_n . Podríamos replicar la funcionalidad de cada *worker* k veces, por ejemplo, el

worker w_i tiene asociado k réplicas. Si falla el worker w_i , entonces ejecutamos el algoritmo del matón para que una de esas k réplicas ocupe su lugar y ese worker w_i pueda volver a estar operativo.

4. Evaluación

La realización de las prácticas es por parejas, pero los dos componentes de la pareja deberán entregarla de forma individual. En general estos son los criterios de evaluación:

- Deben entregarse *todos* los programas, se valorará de forma negativa que falte algún programa.
- Todos los programas deben compilar correctamente, se valorará de forma muy negativa que no compile algún programa.
- Todos los programas deben funcionar correctamente como se especifica en el problema.
- Todos los programas tienen que seguir el manual de estilo de Elixir, disponible en¹ (un 20% de la nota estará en función de este requisito). Además de lo especificado en el manual de estilo, cada fichero fuente deberá comenzar con la siguiente cabecera:

```
# AUTORES: nombres y apellidos
# NIAs: números de identificaci'on de los alumnos
# FICHERO: nombre del fichero
# FECHA: fecha de realizaci'on
# TIEMPO: tiempo en horas de codificación
# DESCRIPCI'ON: breve descripci'on del contenido del fichero
```

4.1. Rúbrica

Con el objetivo de que, tanto los profesores como los estudiantes de esta asignatura por igual, puedan tener unos criterios de evaluación objetivos y justos, se propone la siguiente rubrica en la Tabla 1. Los valores de las celdas son los valores mínimos que hay que alcanzar para conseguir la calificación correspondiente y tienen el siguiente significado:

A+ (excelente). En el caso de software, conoce y utiliza de forma autónoma y correcta las herramientas, instrumentos y aplicativos software necesarios para el desarrollo de la práctica. Plantea correctamente el problema a partir del enunciado propuesto e identifica las opciones para su resolución. Aplica el método de resolución adecuado e identifica la corrección de la solución, sin errores. En el caso de la memoria, se valorará una estructura y una presentación adecuadas, la corrección del lenguaje, así como el contenido explica de forma precisa los conceptos involucrados en la práctica. En el caso del código, este se ajusta exactamente a las guías de estilo propuestas.

A (bueno). En el caso de software, conoce y utiliza de forma autónoma y correcta las herramientas, instrumentos y aplicativos software necesarios para el desarrollo de la práctica. Plantea correctamente el problema a partir del enunciado propuesto e identifica las opciones para su resolución. Aplica el método de resolución adecuado e identifica la corrección de la solución, con ciertos errores no graves. Por ejemplo, algunos pequeños casos (marginales) no se contemplan o no funcionan correctamente. En el caso del código, este se ajusta casi exactamente a las guías de estilo propuestas.

B (suficiente). En el caso de software, conoce y utiliza de forma autónoma y correcta las herramientas, instrumentos y aplicativos software necesarios para el desarrollo de la práctica. No plantea correctamente el problema a partir del enunciado propuesto y/o no identifica las opciones para su resolución. No aplica el método de resolución adecuado y / o identifica la corrección de la solución, pero

¹ https://github.com/christopheradams/elixir_style_guide/blob/master/README.md

con errores. En el caso de la memoria, bien la estructura y / o la presentación son mejorables, el lenguaje presenta deficiencias y / o el contenido no explica de forma precisa los conceptos importantes involucrados en la práctica. En el caso del código, este se ajusta a las guías de estilo propuestas, pero es mejorable.

B- (suficiente, con deficiencias). En el caso de software, conoce y utiliza de forma autónoma y correcta las herramientas, instrumentos y aplicativos software necesarios para el desarrollo de la práctica. No plantea correctamente el problema a partir del enunciado propuesto y/o no identifica las opciones para su resolución. No se aplica el método de resolución adecuado y/o se identifica la corrección de la solución, pero con errores de cierta gravedad y/o sin proporcionar una solución completa. En el caso de la memoria, bien la estructura y / o la presentación son manifiestamente mejorables, el lenguaje presenta serias deficiencias y / o el contenido no explica de forma precisa los conceptos importantes involucrados en la práctica. En el caso del código, hay que mejorarlo para que se ajuste a las guías de estilo propuestas.

C (deficiente). El software no compila o presenta errores graves. La memoria no presenta una estructura coherente y/o el lenguaje utilizado es pobre y/o contiene errores gramaticales y/o ortográficos. En el caso del código, este no se ajusta exactamente a las guías de estilo propuestas.

Calificación	Solución al problema	Código	Memoria
10	A+ (TF en master y worker)	A+	A+
9	A+ (TF en master y worker)	A	A
8	A (Tolerancia a Fallos solo en master)	A	A
7	A (Tolerancia a Fallos solo en master)	B	B
6	B	B	B
5	B-	B	B-
suspense	C	C	C

Tabla 1. Rúbrica

5. Entrega

Deberéis entregar un fichero zip que contenga: (i) los fuentes: master.exs y worker.exs y (ii) la memoria en pdf (4 páginas). La entrega se realizará a través de moodle2 en la actividad habilitada a tal efecto. La fecha de entrega será no más tarde del jueves anterior al comienzo de la siguiente práctica, esto es, el 15 de noviembre de 2018 para los grupos A y el 22 de noviembre de 2018 para los grupos B.

Los días 16 y 23 de noviembre durante la sesión de prácticas se realizará una defensa "in situ" de la práctica.