



PRÁCTICA FINAL – ENERO 2021


SISTEMAS DISTRIBUIDOS.


GRADO EN INGENIERÍA TELEMÁTICA – ETSIT, URJC

 **Profesor:** Felipe Ortega.

 **Versión:** 1.0.

 **Lenguaje de programación:** Go.

 **Fecha tope de entrega:** 22:00 del lunes, 1 de febrero de 2021.

 **Envío:** Esquema del planteamiento de solución y código fuente **que incluya comentarios**, a través del espacio de entrega en Aula Virtual.

1 OBJETIVOS

- Comprender las estrategias y primitivas de sincronización entre gorutinas.
- Comprender los problemas de concurrencia asociados al uso de gorutinas.
- Aplicar conceptos y estrategias para implementación de protocolos de mantenimiento de coherencia en sistemas distribuidos.

EL PROBLEMA DE SANTA CLAUS

Este problema está basado en el problema homónimo de ¹ (cf. sec. 2.3.7). Este es, exactamente, el enunciado planteado para la Práctica 2 de la asignatura. Se incluye aquí únicamente a efectos prácticos, para que toda la información útil para el desarrollo de la práctica final se encuentre en un solo lugar.

¹ David Vallejo, C. G. and Jiménez, J. A. A. (2016). *Programación concurrente y tiempo real*. CreateSpace, 3 edition

2 ENUNCIADO DEL PROBLEMA

Próximamente ya las fechas de Navidad, Santa Claus está en su casa cerca del Polo Norte. Para poder despertarlo de su descanso, se debe cumplir una de las dos condiciones siguientes:

- Que los 9 renos disponibles para tirar del trineo hayan regresado de vacaciones.

- Que alguno de los elfos ayudantes necesiten que le eche una mano para fabricar un juguete.

Para intentar evitar importunar al jefe más de lo debido, los elfos han acordado despertarle si 3 de ellos tienen problemas. En el caso de que un grupo de 3 duendes ya le haya despertado, el resto de duendes con problemas deben esperar a que Santa Claus acabe de ayudar al primer equipo.

Sin embargo, en caso de que en algún momento todos los renos vuelvan de vacaciones, Santa Claus acaba de atender a los elfos a los que estaba ayudando y deja a los demás en espera y prepara el trineo. El reparto es más importante que ayudar a fabricar juguetes, que se pueden entregar el año siguiente. El último reno en llegar es el encargado de avisar a Santa Claus, mientras que cuando todos los anteriores renos van llegando se quedan en espera, antes de ser enganchados al trineo.

2.1 CONDICIONES ADICIONALES

Para concretar la resolución del problema, se establecen las siguientes condiciones adicionales:

- En la fábrica de juguetes hay $D = 12$ elfos trabajando.
- Los renos irán llegando aleatoriamente, con un intervalo de, al menos, 5 segundos entre la llegada de dos renos consecutivos.
- Cada elfo fabrica un juguete. Uno de cada tres juguetes da problemas.
- Santa Claus invierte un tiempo aleatorio de entre 2 y 5 segundos en ayudar a un equipo de 3 elfos con problemas.
- En el caso de que Santa Claus ayude a todos los elfos y todavía no hayan llegado todos los renos del trineo, se vuelve a dormir hasta que le vuelvan a despertar.

3 PLANTEAMIENTO

Además de la función principal, se recomienda crear, al menos, 3 funciones para gorutinas específicas:

- Una gorutina que modela a Santa Claus.
- Una gorutina que modela el comportamiento de un duende.
- Una gorutina que modela el comportamiento de un reno.

Respecto a los eventos que deben ser sincronizados, será preciso tener elementos para despertar a Santa Claus, notificar a los renos que se han enganchado al trineo y controlar la espera de los elfos cuando otro grupo de 3 elfos esté siendo ayudado por Santa Claus.

4 EVALUACIÓN

Se deberá entregar un esquema y planteamiento teórico, donde se describa el problema así como las estrategias y mecanismos de sincronización que se han puesto en práctica para resolver todos los aspectos operativos solicitados en el enunciado.

Además, se entregará el código fuente en lenguaje de programación Go, comentado, implementando el esquema propuesto.

- Corrección del planteamiento teórico para resolver el problema → **3 puntos**.
- Corrección funcional del programa → **4.5 puntos**.
- Corrección de estructura del programa y estilo → **2.5 puntos**.

LOS ALMACENES DE SANTA CLAUS

5 INTRODUCCIÓN

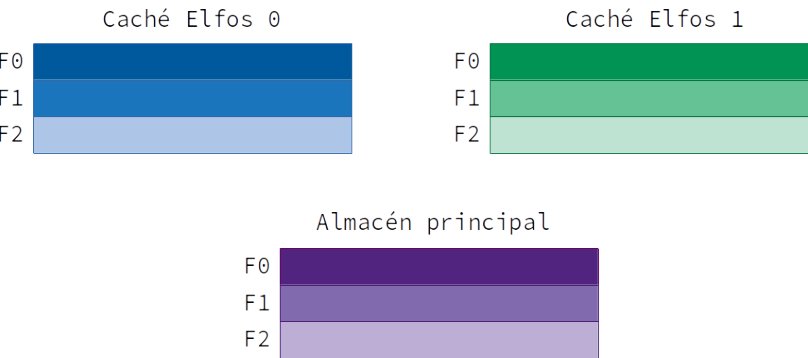
Esta parte del documento presenta el enunciado correspondiente a la Práctica Final de la asignatura. El problema que se plantea se basa, en cierta medida, en el enunciado propuesto para la Práctica 2, por lo que el alumno puede consultar los detalles que necesite refrescar, presentados en la parte I.

6 ENUNCIADO

Considera un **problema de coherencia** que involucra a **dos cachés de duendes**, cada una con idénticas condiciones y restricciones de funcionamiento a las ya indicadas en la Parte I.

Existe un **almacén principal** de regalos por preparar por parte de cada uno de los dos batallones de duendes. Para simplificar un poco el problema, supongamos que el almacén cuenta con **90 espacios en total** para guardar regalos, repartidos en **3 filas de 30 espacios** cada una. La Figura 1 representa esquemáticamente este entorno.

Figura 1: Esquema de las *cachés* de cada batallón de elfos, para que puedan controlar la coherencia entre sus respectivas remesas de regalos, y el *almacén principal*, del que se van sacando progresivamente los regalos para prepararlos.



6.1 CONDICIONES ESPECÍFICAS

Existen **2 batallones de 16 duendes** cada uno, que colaboran con Santa Claus en la preparación de los regalos de Navidad. En cada batallón, hay un duende líder que es el encargado de ir recuperando regalos del almacén principal. El resto de duendes de ese batallón preparan los regalos recuperados por su líder, siguiendo el procedimiento descrito en la parte I (Práctica 2).

Al comienzo del programa el **almacén principal** está **completamente lleno** de regalos (no hay ningún espacio vacío).

El procedimiento que siguen los duendes de cada batallón es el siguiente:

- ▶ Al comienzo, la caché de cada batallón de duendes está vacía. Así que el líder elige aleatoriamente traer un regalo de una de las tres filas del almacén principal. Seguidamente, el **duende líder elige** aleatoriamente uno de los regalos disponibles en esa fila para prepararlo.
- ▶ Inmediatamente después de elegir un regalo en particular, el líder avisa de que se ha traído dicho regalo de la posición seleccionada, siguiendo un protocolo *write through*. El aviso debe actualizar la información tanto en el *almacén principal* como en la *caché* del otro batallón, tal y como se muestra en la Figura 2.
- ▶ En la siguiente iteración en la que el líder de ese batallón necesite otro regalo, lo trae aleatoriamente de otra de las dos filas **que aún no ha usado**. Seguidamente, se repite el procedimiento descrito en los puntos anteriores para preparar un regalo elegido al azar, de entre los de la fila seleccionada, y seguidamente actualizar el almacén principal y la caché del otro batallón, de modo que reflejen el cambio de valor para la posición elegida.
- ▶ En la siguiente iteración del líder de ese batallón, carga otro regalo de la **fila restante que no se ha elegido todavía** en su caché, y repite las operaciones antes descritas.
- ▶ A partir de que se hayan cargado regalos de las tres filas en la caché de ese batallón, en las siguientes iteraciones el líder elige una fila al azar para traer un regalo de una posición de la misma, elegida también al azar. Se sigue de nuevo el protocolo *write through* para actualizar el almacén principal y las otras cachés.
- ▶ El **programa termina** cuando ya no quedan más regalos para preparar en el almacén principal, o bien cuando se cumplen las condiciones de finalización de la parte I, si es que todos los renos llegan a engancharse al trineo de Santa antes de que se acaben los regalos que preparar.

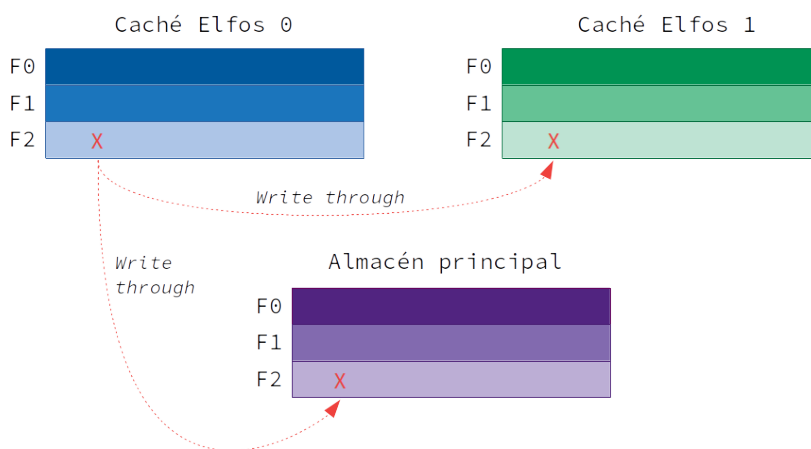


Figura 2: Esquema del procedimiento de actualización *write through* que deben seguirse para mantener la coherencia entre las cachés locales de cada batallón de duendes y el almacén principal.

7 EVALUACIÓN

Los criterios de evaluación para esta práctica serán completamente análogos a los de la Práctica 2.

Se deberá entregar un esquema y planteamiento teórico, donde se describa el problema así como las estrategias y mecanismos de sincronización que se han puesto en práctica para resolver todos los aspectos operativos solicitados en el enunciado.

Además, se entregará el código fuente en lenguaje de programación Go, **comentado adecuadamente**, implementando el esquema propuesto.

- Corrección del planteamiento teórico para resolver el problema → **3 puntos**.
- Corrección funcional del programa → **4.5 puntos**.
- Corrección de estructura del programa y estilo → **2.5 puntos**.

REFERENCIAS

David Vallejo, C. G. and Jiménez, J. A. A. (2016). *Programación concurrente y tiempo real*. CreateSpace, 3 edition.

