



PRÁCTICA 1 – NOVIEMBRE 2020


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 miércoles, 18 de noviembre de 2020.

 **Envío:** Código fuente **que incluya comentarios**, a través del espacio de entrega en Aula Virtual.

1 OBJETIVOS

- Comprender mejor el funcionamiento de las *'goroutines'* explicadas en clase.
- Comprender los problemas de concurrencia asociados al uso de *goroutines*.
- Comprender la utilización de *channels* para comunicación y sincronización entre *goroutines*.
- Consideración de patrones para diseño de programas concurrentes.

2 PUNTO DE PARTIDA: SERVIDOR CHAT

El punto de partida para esta práctica es el ejemplo *Chat server*, que se puede encontrar en el Capítulo 8 (§8.10), del libro “The Go Programming Language”, indexado en la bibliografía básica de la asignatura.

En primer lugar, el alumno debe compilar y ejecutar dicho ejemplo completo, comprendiendo su estructura y el funcionamiento del código y todos los elementos que lo componen en detalle. A continuación, se propone realizar las mejoras descritas en las secciones más abajo, sobre la funcionalidad básica del ejemplo.

- La puntuación máxima que se puede obtener al implementar cada mejora aparece claramente reflejada al comienzo de la sección correspondiente.
- El alumno puede escoger qué mejoras implementa para superar la práctica.
- Cada mejora está, en la medida de lo posible, desligada de las restantes, de forma que la resolución de una mejora no bloquee la de otras secciones.

¹ No es obligatorio implementar todas las mejoras solicitadas en los diferentes apartados del enunciado. Sin embargo, todas las mejoras implementadas deberán entregarse en un único archivo.

Se entregará una sola versión de la solución, que incluya simultáneamente todas las mejoras implementadas sobre el código del ejemplo inicial ¹.

- Para superar la práctica, el alumno debe obtener como mínimo una nota de 5 puntos sobre 10.

3 MEJORA 1: INFORMACIÓN LISTA DE CLIENTES

Puntuación: 2.5 puntos

Modifica el código del servidor de chat para que el `broadcaster` anuncie la lista actualizada de todos los clientes que están conectados al servidor, después de cada llegada de un nuevo cliente.

El conjunto `clientes` y los canales `entering` y `leaving` deben registrar también el identificador de cada cliente.

4 MEJORA 2: NOMBRES DE CLIENTE LEGIBLES

Puntuación: 2.5 puntos

Modifica el código del servidor de chat, cambiando el protocolo de comunicación por red, de forma que cada cliente proporcione un nombre legible al conectarse (que se debe solicitar al cliente por consola en el arranque).

Utiliza este nombre en lugar de la dirección de red y puerto cuando añadas el prefijo de identificación con la identidad del remitente en cada mensaje.

5 MEJORA 3: ENVÍO MULTICAST

Puntuación: 2 puntos

Modifica el código del servidor de chat, de forma que la rutina `broadcaster` reenvíe los mensajes enviados por un cliente a todos los demás clientes excepto a sí mismo.

En caso de haber implementado la mejora 1, este reenvío selectivo no afecta al anuncio de la lista completa de clientes, solo a los mensajes enviados tras la conexión inicial al servidor.

6 MEJORA 4: CANALES PRIVADOS

Puntuación: 3 puntos

Modifica el código del servidor de chat, de manera que un cliente pueda solicitar al servidor la apertura de un canal de comunicación privado con otro cliente en particular.

Por supuesto, la implementación de la mejora 1 puede ser muy útil en este caso para facilitar que un cliente conozca la lista de los demás clientes a los que puede pedir abrir un canal privado, pero no es imprescindible.

En caso de no haber implementado la mejora 1, para pruebas se puede asumir que el cliente en cuestión conoce “de antemano” el identificador del otro cliente.

