Laboratorio 4: Sistemas Distribuidos

Profesor: Jorge Díaz
Ayudantes de Lab: Iñaki Oyarzun M. & Débora Alayo A.

Abril 2024

1 Objetivos del laboratorio

- Aprender acerca de la comunicación en sistemas distribuidos.
- Familiarizarse con tecnologías de comunicación como remote procedures **(gRPC)** y Asynchronous Messaging (**RabbitMQ**) en **Golang**.
- Profundizar el uso de **Docker**.
- Implementar y coordinar diferentes componentes de un sistema distribuido.
- Familiarizarse con conceptos de replicación y consistencia de datos.

2 Introducción

Un sistema distribuido es un conjunto de computadores que trabajan de manera conjunta para cumplir algún objetivo. Para realizar esto, es esencial que estos se comuniquen entre sí mediante el intercambio de mensajes. La comunicación entre estos computadores puede ser desarrollada de distintas maneras, por ejemplo, puede existir una comunicación síncrona, en la cual se asume que los mensajes llegarán en determinado lapso de tiempo, o bien una asíncrona, en la que no se toma este supuesto. Para poner esto en práctica, se propone un sistema en el cual deban hacer uso del intercambio de mensajes para cumplir con las tareas propuestas.

Además de lo anterior, en la medida en la que los componentes de los sistemas distribuidos aumentan en complejidad se hace presente un nuevo problema, la coordinación. Servicios que comparten recursos o que se distribuyen tareas, además de poder comunicarse, deben ser capaces de ponerse de acuerdo para trabajar en conjunto.

Para realizar este laboratorio, se utilizarán tecnologías de comunicación como RPC, RabbitMQ y Docker. En la siguiente sección pueden encontrar documentación acerca de estas tecnologías, siguiendo los enlaces.

3 Tecnologías

- El lenguaje de programación a utilizar es **Go.**
- El sistema de comunicación síncrona a utilizar es **gRPC (video)**
- El sistema de comunicación asíncrona a utilizar es **RabbitMQ**
- Para distribuir el programa se utilizará **Docker.**

4 Laboratorio

El laboratorio consiste en implementar un sistema distribuido inspirado en el videojuego Killing Floor 2

4.1 Contexto

Horzine, una vez reconocida como la cúspide de la innovación humana en tecnología genética, ahora es la precursora de un posible apocalipsis. Esta corporación realizó experimentos secretos para desarrollar superhumanos que representaran al soldado perfecto, combinando avanzadas tecnologías genéticas y mecánicas. Sin embargo, sus creaciones superaron las expectativas de control, evolucionando de maneras imprevistas y comenzando una rebelión sangrienta contra sus creadores.

El caos se desató cuando estos seres modificados, lejos de ser los protectores de la humanidad, se convirtieron en monstruos voraces que amenazan con extinguir la vida humana. En un intento desesperado por rectificar sus errores, el director de Horzine busca la colaboración de gobiernos globales, prometiendo compartir información crucial sobre los experimentos a cambio de ayuda para contener la crisis.

Como último recurso, se ha convocado a un grupo élite de mercenarios, provenientes de diversos rincones del planeta. Estos ocho especialistas en combate tienen la misión crítica de infiltrarse en una base secreta de Horzine, ubicada en los confines más oscuros de la corporación. Allí, deberán superar múltiples desafíos, incluyendo dos intensas oleadas de enemigos en los dos primeros pisos del edificio para llegar al corazón del problema: el Patriarca.

El Patriarca, un ex científico de Horzine que llevó los experimentos de auto-modificación al extremo, ahora es un ser despiadado y la principal figura detrás de la producción de estas abominaciones. Consciente de los movimientos de la corporación, ha preparado trampas letales para eliminar a cualquiera que se atreva a detenerlo. En este entorno hostil, los mercenarios no solo lucharán por la recompensa prometida, una suma considerable de dinero ofrecida por los gobiernos a los sobrevivientes, sino también por la posibilidad de restaurar un semblante de paz y seguridad en un mundo al borde del abismo.

4.2 Explicación

El desarrollo de esta misión consiste en el manejo de los **8 mercenarios**, en donde aquellos que logren salir con vida de la misión serán recompensados con una suma de dinero.

Los mercenarios deberán mantenerse en comunicación de forma síncrona con **El Director** de Horzine. Quién será el encargado de manejar la comunicación estratégica y los movimientos del enemigo dentro del campo, se encargará de solicitar y recibir información de los mercenarios

como sus decisiones dentro del transcurso de la misión o si están preparados para el ingreso a cada uno de los pisos del edificio.

Junto con eso **El Director** para complacer a sus líderes y Gobiernos, deberá llevar un registro sobre las decisiones de los mercenarios dentro de la misión para poder entregar dicha información, para esto deberá mantener el contacto con un proceso llamado **NameNode** al cual le informa todas las decisiones tomadas por los mercenarios. Este NameNode deberá decidir la forma de registrar estas elecciones tomadas por los mercenarios distribuyendo estos datos a otros 3 procesos llamados **DataNode**. Estos procesos DataNode, deberán crear un archivo por cada mercenario asignado en el piso del edificio y registrar sus acciones.

Además, **El Director** deberá mantener la comunicación asíncrona con un proceso llamado **Dosh Bank**, cuando un mercenario sea eliminado, el Director deberá informar al Dosh Bank para que la entidad aumente en 100 millones de libras el pago a los mercenarios que salgan con vida.

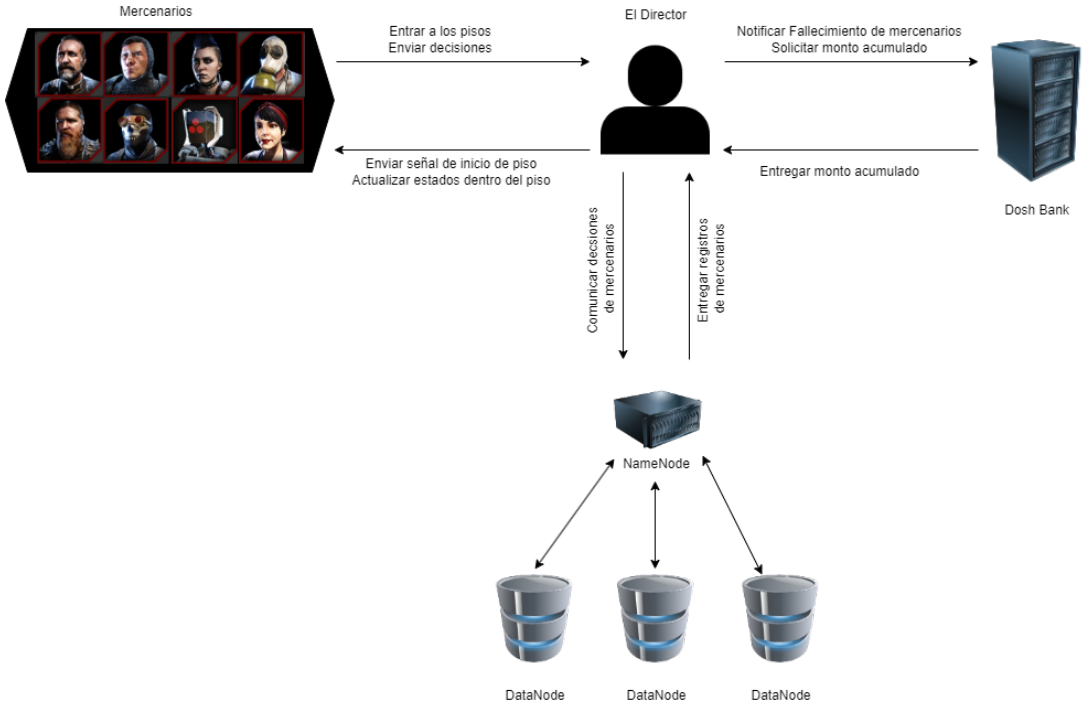

Figure 1: Relaciones del sistema

5 Especificación por proceso

Se entrega a continuación una información más detallada sobre las funcionalidades que deberán disponer cada una de estas entidades.

5.1 Mercenarios

Son los participantes en la misión que se llevará a cabo. Harán de *clientes* en este sistema. Uno de estos mercenarios deberá ser una persona, es decir, que pueda ser controlado por consola. El

resto de mercenarios pueden ser bots, es decir, que de manera automática tomen las decisiones. Entre las funcionalidades de estos mercenarios se encuentran:

- Enviar una petición para informar su estado de preparación de la misión.
- Ser capaces de tomar las decisiones en los 3 pisos a explorar, es decir, su código debe estar implementado para poder enviar las respuestas requeridas por el Director en cada piso (que será una dinámica diferente en cada uno, siendo explicado en secciones de más adelante)
- En caso de fallecer, debe terminar su código.
- Pedir al Director ver el monto que se tiene acumulado en el Dosh Bank.

Los procesos de los jugadores pueden estar ejecutándose en diferentes máquinas y deben ser capaces de conectar con el Director. **Toda comunicación con el Director debe realizarse por gRPC.**

5.2 El Director

Es el encargado de la logística y planificación estratégica de la misión. Coordina y maneja las decisiones de los mercenarios además de controlar las entidades de Dosh Bank y NameNode. **Debe esperar a que todos los mercenarios confirmen su preparación a la misión para comenzar.** El Director debe ser capaz de:

- Disponer de una interfaz por consola
- Mostrar por consola cada vez que fallece un mercenario
- Mostrar por consola los mercenarios que quedan vivos al término de un piso
- Mostrar por consola los mercenarios que finalizaron con éxito la misión (en caso de lograrlo).
- Mediante la interfaz dar comienzo al siguiente piso en la misión
- Se debe poder consultar mediante la interfaz, todas las decisiones en todas las rondas de un determinado mercenario.
- Debe detectar si un mercenario falleció, e informar al proceso del mercenario junto al Dosh Bank
- Debe enviarle todas las decisiones de todos los mercenarios al NameNode

Este proceso debe estar corriendo solamente en una de las máquinas virtuales. Debe comunicar de manera asíncrona, mediante RabbitMQ, al Dosh Bank cuando un mercenario es eliminado, pero debe poder preguntar de manera síncrona el monto acumulado en el Dosh Bank usando gRPC. Debe comunicarse de manera síncrona con el NameNode utilizando gRPC.

5.3 Dosh Bank

Es el encargado de mantener el conteo del monto acumulado por los mercenarios en la misión. Debe crear un archivo txt donde se registre cada uno de los mercenarios eliminados y el monto acumulado actual. El Dosh Bank debe ser capaz de:

- Registrar cada uno de los mercenarios eliminados en el archivo txt de la siguiente forma:
  - Mercenario Numero_piso Monto_acumulado_actual

  - ...
  - D.A.R. Piso_1 100000000
  - Mr.Foster Piso_2 200000000

- Responder a las peticiones sobre el monto actual acumulado

Este proceso debe estar corriendo solamente en **una** de las máquinas virtuales. Debe procesar de manera asíncrona, mediante RabbitMQ, el registro de mercenarios eliminados, pero de manera síncrona responder a la petición del monto acumulado.

5.4 NameNode

Es el encargado de distribuir el registro de las elecciones de los mercenarios en cada piso. El NameNode debe ser capaz de:

- Debe distribuir, de manera aleatoria entre los 3 DataNode, la tarea de registrar las decisiones de los mercenarios en cada piso.
- Registrar en un archivo txt donde están almacenadas las elecciones de cada mercenario en cada piso de la siguiente forma:
  - Mercenario Piso_numero IP_datanode
  - ...
  - D.A.R Piso_1 10.0.0.15

Este proceso debe estar corriendo solamente en una de las máquinas virtuales. Debe procesar de manera síncrona el registro de las decisiones de los mercenarios y la petición de ver deciciones almacenadas de mercenarios.

5.5 DataNode

Son los procesos encargados de almacenar el registro de las decisiones de los mercenarios. Los DataNode deben ser capaces de:

- Debe registrar, en archivos txt, las decisiones de cada mercenario en cada piso de la siguiente manera:
  - **Archivo:** Mercenario_NumeroPiso.txt
    - decisión
  - **Archivo:** DAR_1.txt
    - 3

Este proceso debe estar corriendo en tres máquinas virtuales diferentes, pero no en la máquina donde se esté ejecutando el proceso NameNode. Debe procesar de manera síncrona, el registro de las decisiones de los mercenarios y la petición de obtener el registro de las decisiones de los mercenarios.

6 Pisos

6.1 Piso 1: Entrada al infierno

Este piso consiste en el manejo de probabilidades dispuestas por el Director para la supervivencia, dependiendo de los enemigos detectados por el radar.

Luego de que cada mercenario este confirmado, cada mercenario deberá elegir una de estas 3 armas:

- (1) Escopeta
- (2) Rifle automático
- (3) Puños eléctricos

El Director, al recibir todas las elecciones de los mercenarios, calculará la probabilidad de éxito de cada arma, utilizando la partición aleatoria:

- Generar dos números entre 0 y 100 y diferentes entre sí, por ejemplo X e Y
- Asumiendo que $X < Y$ se calculan las probabilidades tal que:
  - Probabilidad 1: X
  - Probabilidad 2: $Y - X$
  - Probabilidad 3: $100 - Y$

Luego del cálculo de probabilidades el Director utilizará los valores para determinar la supervivencia o muerte del mercenario de acuerdo con el arma elegida. Aquellos mercenarios que de acuerdo a la probabilidad del arma elegida hayan fallecido. Quedan eliminados de la misión. Pasando con esto al siguiente piso.

6.2 Piso 2: Trampas y Traiciones

Este piso contiene una trampa preparada por El Patriarca!, El Director detecta que existe una separación de los caminos en el edificio, uno de ellos llevará al Patriarca, mientras que el otro llevará solamente a la perdición.

En este piso, los mercenarios deberán decidir a cual de los pasillos accederán (A o B), El director una vez recibidas las decisiones de los mercenarios, deberá calcular la salida correcta eligiendo al azar uno de los dos caminos con una probabilidad del 50% cada uno. Aquellos mercenarios que hayan elegido la misma salida que la obtenida por El Director, pasarán al piso final, mientras que el resto quedará eliminado de la misión.

6.3 Piso 3: Confrontación Final

Es El Patriarca!, finalmente lo han encontrado los mercenarios.

En este piso, los mercenarios deberán elegir un número entre el 1 y 15 de manera aleatoria en cada una de las 5 rondas y deberán acertar en al menos 2 ocasiones el valor obtenido por el Patriarca (de 1 a 15) que es calculado por el Director una vez obtenidas las decisiones para poder obtener la victoria dentro de la misión.

Si dentro de las 5 rondas, ninguno de los mercenarios ha podido acertar al menos 2 veces el número del Patriarca, estarán eliminados de la misión, mientras que el resto habrá salido victorioso, terminando todo el caos ocurrido por Horzine.

7 Comunicación y Máquinas Virtuales

Para la ejecución del laboratorio se utilizarán 4 máquinas virtuales, las cuales serán provistas por los ayudantes de laboratorios a los integrantes de los equipos, junto con las instrucciones correspondientes para el acceso a estas máquinas que serán publicadas en aula.

Para los procesos **Director, NameNode y Dosh Bank** deberán estar ejecutándose en máquinas diferentes cada uno.

Para la comunicación entre los **Mercenarios y el Director**, **Director y NameNode** y **NameNode y los DataNode**, deberán utilizar llamadas a procedimientos remotos, para esto deben emplear gRPC como la tecnología que se haga cargo de esta parte.

Por otro lado, para la comunicación entre el **Dosh Bank y el Director** sobre la notificación de un mercenario eliminado, debe ser de manera asíncrona utilizando como tecnología base RabbitMQ. Sin embargo, las peticiones sobre la cantidad de dinero acumulado debe ser utilizando gRPC.

8 Uso de Docker

Para la ejecución del laboratorio en las máquinas virtuales, se implementará el uso de Docker como herramienta encargada de aislar las instancias de cada uno de los programas a desarrollar exceptuando los mercenarios, es decir, **El director, NameNode, los DataNode y Dosh Bank** deberán estar en contenedores de Docker dentro de cada una de las máquinas virtuales que haya asignado.

9 Restricciones

- Todo uso de librerías externas que no se han mencionado en el enunciado debe ser consultado dentro del foro de laboratorio.

10 Consideraciones

- Si en algún momento de cualquier piso queda un solo mercenario, se detiene la misión y se considera como fallida, entregando el monto al sobreviviente.
- Para el manejo de consultas, deberán ser canalizadas por medio de **aula** para que el resto de equipos pueden estar al tanto de cualquier actualización o aclaración.
- Todas las entidades conocen las direcciones IP y puertos entre sí.
- Las librerías de **Golang** permitidas son:
  - time
  - strconv
  - strings
  - math
  - math/rand
  - net
  - context

- fmt
- log
- os
- os/signal
- sync
- bufio
- grpc
- amqp

11 Reglas de Entrega:

- El laboratorio se realiza en los grupos ya publicados en aula.
- La fecha de entrega es el día **Viernes 17 de Mayo hasta las 23:59hrs**
- El laboratorio se revisará en las máquinas virtuales, por lo que los archivos necesarios para la correcta ejecución de esta debe estar en ellas. El código deberá estar correctamente identado, comentado, sin Warnings y sin errores.
- Se aplicará un descuento de **5 puntos** al total de la nota por cada Warning, Error o Problema de Ejecución.
- Además de los códigos en las máquinas virtuales deberá subir un archivo comprimido que contenga todos los códigos desarrollados en carpetas separadas por entidad en formato **.zip** con el nombre **Grupo-XX-Lab4.zip**.
- **Debe** dejar un **MAKEFILE** o similar en la entrega para la ejecución de cada entidad.
- **Debe** dejar un **README** en la entrega asignada a su grupo con nombre y rol de cada integrante, además de la información necesaria para ejecutar, la cual debe ser clara pues **con eso se evaluará el código**.
- No se aceptan entregas que no puedan ser ejecutadas desde una consola de comandos. Incumplimiento de esta regla, significa **nota 0**.
- Cada hora o fracción de atraso se penalizará con un descuento de **10 puntos**.
- Copias serán evaluadas con **nota 0** y serán notificadas a los profesores y autoridades pertinentes.
- El uso de **Docker** es obligatorio. Teniendo nota **0** aquellas implementaciones que no hagan uso de Docker.

12 Consultas:

Para hacer las consultas, recomendamos hacerlas dentro del foro habilitado en aula. De esta forma los demás grupos pueden beneficiarse en base a la pregunta. **Se responderán consultas hasta 48hrs. antes de la fecha y hora de entrega.**