

# Laboratorio 2: Sistemas Distribuidos

**Profesores:** Jorge Díaz & Sebastián Godinez  
**Ayudantes de Lab:** Iñaki Oyarzun M. & Sebastián Martínez C.

Octubre 2022

## 1 Objetivos del laboratorio

- Aprender acerca de la comunicación en sistemas distribuidos.
- Familiarizarse con los conceptos asociados a **Naming y Coordinación** en Sistemas Distribuidos
- Profundizar el uso de **Golang y gRPC**

## 2 Introducción

Dentro de los sistemas distribuidos, el manejo de información se realiza en la mayoría de las situaciones a partir de la centralización en una entidad o sistema que conoce y distribuye estos datos. Para estos casos, la entidad mantiene un registro de donde esta almacenada la información para en caso de que sea solicitada pueda buscar y saber a quién preguntar, esto se conoce como **namining**. Generalmente se observa también en sistemas como Hadoop.

Para poner esto en práctica, se propone el siguiente problema en donde haciendo uso de gRPC se aplicará un caso de naming y coordinación entre entidades para subir y obtener información.

## 3 Tecnologías

- El lenguaje de programación a utilizar es **Go**
- Para las comunicaciones se utilizará **gRPC**

## 4 Laboratorio

### 4.1 Contexto

Es el año 2XXX, el complejo de investigación Black Mesa, ubicado en Nuevo Mexico, ha causado un desastre que significaría el fin de la humanidad como se conocía, la llamada "*Cascada de Resonancia*". Producida en el Sector C de pruebas, la cual investigaba la creación de portales para viajes a larga distancia.

Un error científico provocó una reacción en cadena de aparición de portales hacia otras dimensiones, como XEN, habitadas por alienígenas monstruosos y entidades altamente nocivas para el planeta y entre ellas, igualmente a la peor entidad conocida en el multiverso entero.

Los Combine, un imperio interdimensional disperso entre una cantidad infinita de universos y sistemas solares. Cuyo único objetivo es dominar el multiverso, junto con controlar y esclavizar a todas las especies que existen.

La humanidad ante esta gran amenaza, decidió unir fuerzas contra este gran mal común, sin embargo, todo fue en vano, en tan solo 7 horas logró sucumbir el planeta completo. Pasando a ser anexada la tierra como parte de los Combine y los humanos esclavizados.

Sin embargo, aún existe una luz de esperanza, sobrevivientes al gran desastre del laboratorio de Black Mesa crean una gran fuerza de resistencia que busca como objetivo luchar desde las sombras por la tan anhelada libertad de los humanos. Esta resistencia liderada y enardecida por su actualmente desaparecido líder de iniciales G.F. Actualmente la resistencia sobrevive de la infiltración en los sistemas de datos en la central de los Combine (La Ciudadela) para conocer sus planes y poder reaccionar ante posibles ataques. Últimamente han detectado estas infiltraciones y han actualizado completamente los sistemas, quedando obsoletas todas las tecnologías diseñadas actualmente. Su grupo, como grandes programadores de la época dorada de la humanidad han sido llamados para diseñar y estudiar formas de la obtención de datos vitales para la resistencia. Quedando en sus manos el futuro de la humanidad.

## 4.2 Explicación

### 4.2.1 Diagramas de sistemas

Para ayudar a los Rebeldes, se le hace entrega de la siguiente información enviada minutos antes de la muerte de un escuadrón enviado a investigar:

- Los nuevos sistemas funcionan a partir de 3 nodos de información (llamados DataNodes) los cuales almacenan datos de manera desordenada de 3 tipos (Logística, Financiera y Militar), teniendo una entidad común que se encarga de coordinar y a la vez conoce la forma en que se distribuye la información (llamado NameNode). Los computadores de la red combine se conectan a este NameNode para subir sus archivos y este se encarga de distribuirla. (Revisar Diagrama adjunto para mayor detalle.)

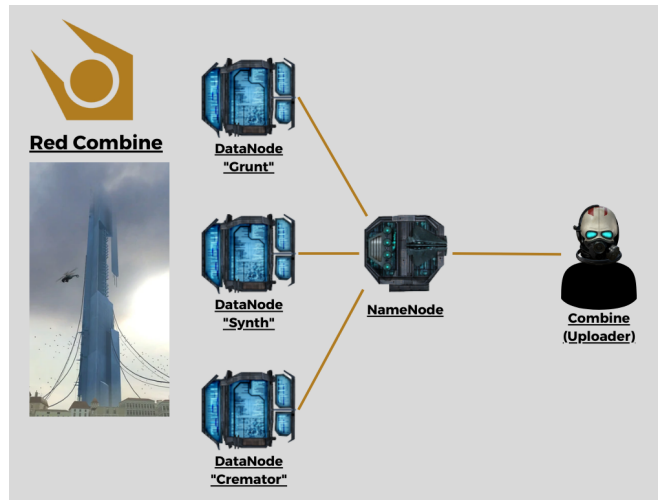


Figure 1: Estructura de los sistemas actuales

- Se requerirá establecer una conexión con el NameNode para poder descargar los datos correctamente ordenados, generando la siguiente distribución de conexiones.

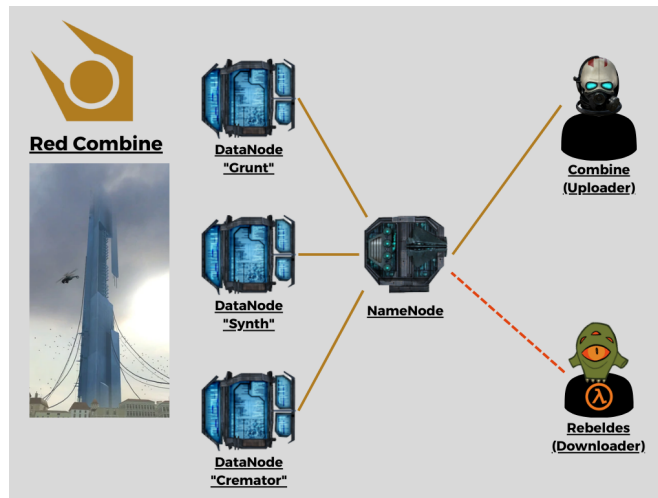


Figure 2: Estructura de los sistemas, cuando se genere el ingreso al sistema

### 4.2.2 Explicación entidades presentes

Esta sección explicará el funcionamiento de cada una de las entidades a crear para desarrollar el laboratorio.

- **Combine:**

- Conoce y se conecta solamente la IP del NameNode
- Se conectan de manera síncrona al NameNode para ingresar información a la base y según un tipo (**Logística, Financiera o Militar**)
- Se tendrá una interfaz por consola para realizar el ingreso de un texto al NameNode y para terminar la interfaz.
- El formato del texto a solicitar para ingresar es en el siguiente formato:  
<TIPO> : <ID> : <DATA> por ejemplo: MILITAR : 122 : LLEGADA DE SUMINISTROS A DEPOSITO CITADELA
- El id de cada input no puede ser igual en ninguna de las categorías, es decir, no se puede ingresar 2 veces un mismo ID.

- **NameNode:**

- Conoce y se conecta a la IP de los DataNodes, Combine y Rebeldes (Se insertó su IP en un ataque realizado anteriormente)
- Al recibir una información a almacenar de parte del Combine, el NameNode de manera aleatoria selecciona un DataNode para entregarle la información, luego de elegirlo, registra en una nueva línea dentro de un archivo llamado **DATA.txt** la categoría, el ID y el DataNode en donde fue almacenado el dato. Luego de ello le envía el mensaje completo al DataNode seleccionado para que sea almacenado.
- El NameNode solo puede recibir una consulta por todos los registros de una categoría. (*Hint: Se puede crear más de una función para ser usada en gRPC*)
- Al recibir la consulta de los registros de una categoría, el NameNode ingresará a su registro **DATA.txt** para ubicar donde está guardado cada mensaje (de acuerdo al id de la información) y solicitarlo al DataNode para finalmente enviar todos los mensajes de dicha categoría ordenados por los mensajes más antiguos a los más nuevos, indicando solo el ID del mensaje y la DATA del mensaje (*Hint: Observar el orden en que se agregan los registros a este archivo DATA*).

- **DataNodes:**

- Conocen y se conectan solamente a la IP del NameNode
- Almacenan la información entregada por el NameNode en un archivo local, llamado DATA.txt
- Cuando el NameNode lo solicite, los DataNodes deben entregar la información correspondiente al mensaje.

- **Rebeldes:**

- Conocen y se conectan solamente a la IP del NameNode
- Poseen una consola para seleccionar y consultar al NameNode por la información de una categoría en específica a la base de datos a partir de la función que posee el NameNode.
- Cuando se muestre por pantalla cada mensaje obtenido del NameNode, será por cada línea separado.

## 5 Restricciones

- Todo uso de librerías externas que no se han mencionado en el enunciado debe ser consultado en aula.

## 6 Consideraciones

- **Prints por pantalla:** Para ver el desarrollo y a la vez como apoyo para el debugging dentro del laboratorio, se solicitará que se realicen los siguientes prints por pantalla como mínimo:
  - **(DataNode):** Mostrar por pantalla el mensaje recibido y respuesta enviada.  
(Ej: `Solicitud de NameNode recibida, mensaje enviado: X`)
  - **(NameNode):** Mostrar por pantalla el mensaje recibido y respuesta enviada.  
(Ej: `Solicitud de Y recibida, mensaje enviado: X`)
  - **(Combine):** Mostrar por pantalla el mensaje recibido al enviar o subir un mensaje.  
(Ej: `Mensaje enviado Exitosamente`)
- Se utilizarán las 4 máquinas virtuales entregadas a cada equipo del laboratorio anterior.
- Los DataNodes y el NameNode deben estar en máquinas diferentes, Rebeldes y Combine no deben estar juntos en una máquina.
- Se realizará una ayudantía para poder resolver dudas y explicar la tarea. Será notificado por aula.
- Consultas sobre la tarea se deben realizar en Aula o enviar un correo a **Sebastián o Iñaki** con el asunto **Consulta grupo XX - Lab 2**
- Las librerías de **Golang** permitidas son:
  - `time`
  - `strconv`
  - `strings`
  - `math`
  - `rand`
  - `net`
  - `context`
  - `fmt`
  - `os`
  - `bufio`
  - `log`
- **Sobre Github:** Cada grupo tendrá acceso a un repositorio privado (se ha creado uno nuevo dentro de la organización a la que fueron agregados) el cual le permitirá actualizar y entregar el código de su laboratorio. Para ello, el día de la entrega, deberá dejar en la rama principal (main) todos sus archivos finales.
- **Bono docker:** Existirá una bonificación de **10 ptos.** para aquellos grupos que puedan hacer uso de Docker para encapsular su código y ejecutarlo en cada una de las máquinas virtuales provistas (dejando evidencia de aquello igualmente en el repositorio) **OJO:** Solo se entregará dicho bono para aquellos que realicen la implementación para todas las entidades (NameNode, DataNodes, Combine y Rebeldes).

- **Término de la ejecución:** **DEBE** ser implementado un cierre de todos los programas por medio de la consola de los Rebeldes con una función hacia el NameNode, para notificar a los DataNodes del cierre (El Combine no necesita cerrarse debido a que es una interfaz manejada por usuario), luego de ello la consola Rebelde recibe el mensaje de confirmación del cierre.

## 7 Reglas de Entrega:

- El laboratorio se entrega en **grupos de 2 personas**
- La fecha de entrega es el día **2 de Noviembre a las 23:55 hrs.**
- La tarea se revisará en las máquinas virtuales, por lo que, los archivos necesarios para la correcta ejecución de esta, deben estar en ellas. Recuerde que el código debe estar indentado, comentado, sin Warnings ni errores.
- Se aplicará un descuento de **5 puntos** al total de la nota por cada Warning, Error o Problema de Ejecución.
- Debe dejar un **MAKEFILE** o similar en cada máquina virtual asignada a su grupo para la ejecución de cada entidad. Este debe manejarse de la siguiente forma:
  - `make datanode`: Iniciará el código para el DataNode
  - `make namenode`: Iniciará el código para el NameNode
  - `make combine`: Iniciará el código para Combine
  - `make rebeldes`: Iniciará el código para Rebeldes
  - `make docker-datanode`: **(Opcional)** Iniciará el código hecho en Docker para el DataNode.
  - `make docker-namenode`: **(Opcional)** Iniciará el código hecho en Docker para el NameNode.
  - `make docker-combine`: **(Opcional)** Iniciará el código hecho en Docker para Combine.
  - `make docker-rebeldes`: **(Opcional)** Iniciará el código hecho en Docker para los Rebeldes.
- Debe dejar un **README** en el repositorio, que indique las instrucciones de ejecución y lo asumido para el desarrollo del laboratorio.
- No se aceptan entregas que no se encuentren dentro del repositorio, tampoco de las máquinas y que no puedan ser ejecutadas desde una consola de comandos. Incumplimiento de estas reglas, significa **nota 0**.
- No se revisará de manera local los archivos, solo los almacenados en las máquinas virtuales.
- Cada hora o fracción de atraso se penalizará con un descuento de **5 puntos**.
- Copias serán evaluadas con **nota 0** y serán notificadas a los profesores y autoridades pertinentes.

## 8 Consultas:

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