

# Laboratorio: Sistemas Distribuidos

**Profesora:** Erika Rosas Olivos - Juan Calderón Maureira

**Ayudantes:** Jorge Díaz M. [jorge.diazma@sansano.usm.cl](mailto:jorge.diazma@sansano.usm.cl)

Noviembre 2021

## 1. Objetivos

- Familiarizarse con conceptos de replicación y consistencia de datos.
- Aplicar consistencia eventual en un sistema distribuido.
- Aplicar modelos de consistencia centrados en el cliente

## 2. Introducción

La replicación es una excelente opción que permite mejorar la tolerancia a fallos de un sistema distribuido. Además, permite mejorar el rendimiento, dividiendo la carga entre las distintas réplicas o bien acercando geográficamente los datos a los clientes que los utilizarán. Sin embargo, la replicación tiene un trade-off y es que introduce problemas de consistencia, ya que cuando se actualiza una réplica, entonces se vuelve diferente de las otras. Para solucionar esto, se han propuesto modelos de consistencia.

En este laboratorio se propone implementar un sistema en el cual deban poner en práctica la replicación, donde además deberán aplicar modelos de consistencia centrados en los datos y centrados en el cliente. Para la tarea deberán utilizar el lenguaje de programación GO y en particular deberán utilizar un modelo de consistencia eventual y Monotonic Reads junto a Read your Writes.

## 3. Tecnologías

- El lenguaje de programación a utilizar es **Go**
- Para la comunicación se utilizará **gRPC**

## 4. Laboratorio



El laboratorio consiste implementar un sistema distribuido inspirado en el universo de Star Wars.

### 4.1. La Chispa de la Rebelión

Nos encontramos en un  
periodo de guerra civil. Las  
naves espaciales rebeldes,  
atacando desde una base  
oculta, han logrado su  
primera victoria contra  
el malvado Imperio Galáctico.

El Almirante Thrawn,  
un fiel servidor del emperador se  
entera que la ex-jedi Ahsoka Tano  
está apoyando a los rebeldes  
enviando información respecto  
a la cantidad de los rebeldes  
que hay en cada planeta,  
por lo que busca intervenir  
en este sistema para  
falsificar la información.

Mientras tanto, perseguida  
por los siniestros agentes del  
Imperio, la Princesa Leia  
vuela hacia su patria, a  
bordo de su nave espacial,  
reuniendo la mayor cantidad  
de información de los rebeldes  
para que puedan salvar a su pueblo  
y devolver la libertad a la  
galaxia....

## 4.2. Arquitectura del sistema

El sistema distribuido estará compuesto por 5 entidades: La Princesa Leia, Ahsoka Tano, el Almirante Thrawn, el Broker Mos Eisley y los servidores Fulcrum. En la siguiente imagen se representa las interacciones entre las entidades.

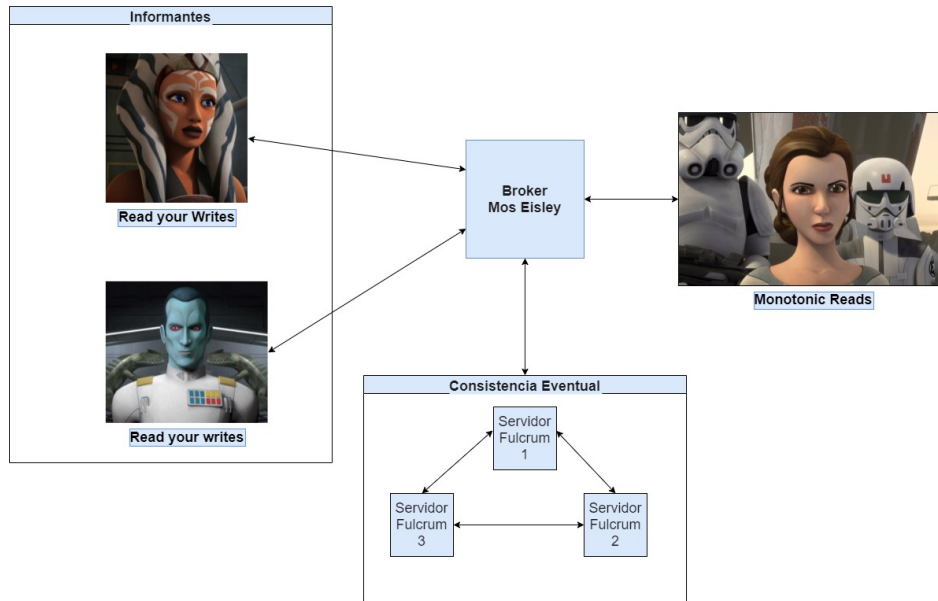


Figura 1: Relaciones del Sistema

### 4.2.1. Servidores Fulcrum

Estos servidores serán tres y todos cumplen las mismas funciones. Son réplicas que permiten mejorar la tolerancia a fallas del sistema. Estos deberán hacer lo siguiente:

- Almacenar **Registros Planetarios**, los cuales deben contener información con el siguiente formato:

nombre\_planeta nombre\_ciudad cantidad\_soldados\_rebeldes

Ejemplo:

Tatooine Mos\_Eisley 5

Notar que debe crearse un archivo por cada planeta.

- Cada uno de estos archivos deberá estar relacionado con un reloj de vector. Este reloj, debe tener la forma  $[x,y,z]$ , donde la posición en el vector indicará el Servidor Fulcrum sobre el cual se está modificando un archivo, y el valor ubicado en cada posición indicará la cantidad de cambios que ha realizado el servidor correspondiente a aquella posición.
- Llevar un **Log de registro** junto a cada Registro Planetario, cuya función será registrar los cambios (AddCity/DeleteCity/UpdateName/UpdateNumber) que se hayan realizado sobre el. Una vez se hayan propagado los cambios de los Servidores Data, estos archivos deben ser eliminados y creados nuevamente vacíos. La idea de estos archivos es facilitar la resolución de conflictos durante la propagación de cambios. Este log debe llevar la siguiente estructura:

accion planeta-afectado ciudad-afectada [nuevo-valor]

Ejemplo:

UpdateNumber Tatooine Mos\_Eisley 100

AddCity Tatooine Mos\_Pelgo 8

UpdateName Tatooine Mos\_Eisley Mos\_Gamos

DeleteCity Tatooine Mos\_Eisley

Notar que en el comando DeleteCity no se debe guardar el nuevo valor ya que no es necesario para ese comando. Además, para el comando AddCity este puede contener o no el nuevo-valor.

- Mantener **consistencia eventual**, realizando propagación de los cambios cada 2 minutos.
- Cuando se encuentren problemas de consistencia entre las versiones de las réplicas, entonces se deberá realizar un merge entre estas, lo cuál debe ser realizado de manera automática. Para esto, se utilizarán los Logs de registro mencionados anteriormente.

#### 4.2.2. Broker Mos Eisley

Es el componente encargado de balancear la carga entre los diversas réplicas. Esta entidad debe cumplir con las siguientes características:

- Redirige a los Informantes a una de las réplicas de forma aleatoria. En el caso de la princesa Leia, se comporta como un intermediario entre estos y los Servidores Fulcrum, por lo que se encarga de transmitir los mensajes enviados entre ambos componentes.
- Redirige a la Princesa Leia y los Informantes a una réplica en específico cuando estas tengan un conflicto con las versiones de los Registros Planetarios.

#### 4.2.3. Informantes

Los informantes son 2: Ahsoka Tano y el Almirante Thrawn. Son los encargados de agregar información a los registros planetarios de los servidores Fulcrum, además de actualizar o elimina dichos registros. Para lo anterior dispone de los siguiente cuatro comandos:

- **AddCity nombre\_planeta nombre\_ciudad [nuevo\_valor]** : esto creará una nueva línea en el registro planetario correspondiente. Si dicho planeta aún no posee un archivo de registro planetario debe crearse uno. Este comando puede o no ingresarse con el nuevo\_valor. En caso de no escribirse uno, debe guardarse esa ciudad con valor 0.
- **UpdateName nombre\_planeta nombre\_ciudad nuevo\_valor** : Con este comando se debe cambiar el nombre a la ciudad en el registro planetario.
- **UpdateNumber nombre\_planeta nombre\_ciudad nuevo\_valor** : Con este comando se debe actualizar el número de rebeldes presentes en dicha ciudad en el registro planetario.
- **DeleteCity nombre\_planeta nombre\_ciudad** : Con este comando se debe eliminar la linea de dicha ciudad en el registro planetario.

Los comandos se envían al Broker Mos Eisley, el cual responde con la dirección del los servidores Fulcrum aleatoriamente seleccionado. Posterior a esto, los informantes se conectan al Servidor Fulcrum y vuelven a enviar el mismo comando que le envió al Broker Mos Eisley de manera automática. La respuesta del Servidor Fulcrum es el reloj de vector del registro planetario del planeta que se modificó.

Los informantes deben utilizar el modelo de consistencia **Read your Writes**. Para poder llevar a cabo la consistencia, los informantes deben mantener en memoria la información de los registros planetarios ha modificado, junto con el reloj de vector del archivo del planeta y la dirección del servidor Fulcrum al que se conectó por última vez a ese registro.

#### 4.2.4. Leia Organa

Es quien realiza las consultas al Broker Mos Eisley para saber la cantidad de rebeldes que hay por ciudad en cada planeta. Para realizar esta consulta, para ello se define el siguiente comando:

- **GetNumberRebelds nombre\_planeta nombre\_ciudad** : este mensaje se envía al Broker Mos Eisley y recibe por respuesta la cantidad de rebeldes que hay en dicha ciudad de ese planeta, junto con el reloj vectorial del registro planetario del planeta consultado.

Leia Organa debe utilizar el modelo de consistencia **Monotonic Reads**. Para poder llevar a cabo la consistencia, Leia debe mantener en memoria la información de las ciudades que ha solicitado, junto con el reloj de vector del archivo del planeta y el servidor Fulcrum que le respondió por última vez al solicitar por esa ciudad en ese planeta.

## 5. Consistencia Eventual

Para realizar las consistencias solicitadas en este laboratorio se deben hacer uso de “relojes de vectores”. Como se mencionó anteriormente, estos deben ser de la forma  $[x, y, z]$ , donde la posición en el vector indicará el Servidor Fulcrum, y el valor en cada posición indicará la cantidad de cambios que ha realizado cada servidor en un archivo en particular. Cada vez que se detecte que hubo cambios concurrentes en los archivos a partir de los relojes de vectores al momento de propagar los cambios, se debe someter a un proceso de merge.

### 5.1. Merge

Al momento de replicar los cambios entre los servidores Fulcrum, estos deben deben asimilar los cambios que se han ido generando entre sí en lo que se denomina un proceso de merge. Para ello, se define un nodo dominante el cual se utilizará de base para aplicarle cambios que se hayan efectuado en los otros nodos.

Utilizando los log de cambios de los otros servidores Fulcrum, se podrán comparar las operaciones realizadas por cada máquina, para así luego aplicar los cambios en el archivo del nodo dominante, el cuál sería la nueva versión que debe ser propagada a todas las réplicas. Cabe mencionar que cuando ocurre un merge, los relojes de vectores deben ser actualizados para coincidir con la nueva versión, por ejemplo, si se tienen los vectores  $[1,0,0]$ ,  $[0,1,0]$  y  $[0,0,0]$ , entonces tras el proceso de merge el reloj de vectores en todas las máquinas debiera ser  $[1,1,0]$ .

Puesto que es posible que hayan conflictos por casos particulares, por ejemplo, que se encuentre registrado una misma combinación planeta-ciudad con valores distintos entre las máquinas; el método de resolución de estos inconvenientes es libre, sin embargo, debe obedecer al principio de consistencia, es decir, que finalizado el proceso de merge no existan diferencias entre los servidores y a la estructura de los registros planetarios. Puesto que se está trabajando con el paradigma de consistencia eventual, el hecho de que se “pierda” información no es un inconveniente.

La resolución de problemas por casos particulares debe quedar documentado puesto que corresponden a decisiones de diseño que cada grupo determinó en su implementación.

## 6. Restricciones

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

La distribución de las máquinas debe seguir las siguientes reglas:

- Los tres procesos de servidores Fulcrum deben estar en 3 máquinas diferentes.
- El broker Mos Eisley debe estar en un servidor diferente a todos los servidores Fulcrum.
- Los procesos Leia Organa, los informantes y el broker Mos Eisley deben estar en máquinas diferentes.

## 7. Consideraciones

- Debido a la cantidad de ayudantes este semestre, las dudas serán **exclusivamente** canalizadas por Aula en foros creados para el laboratorio.

## 8. Reglas de Entrega

- La tarea se debe realizar en los grupos ya publicados.
- La fecha de entrega es el día 10 de diciembre a las 23:55.
- La tarea se revisará en las máquinas virtuales, por lo que los archivos necesarios para la correcta ejecución de esta debe estar en ellas. Recuerde que el código debe estar indentado, 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 esto, debe subir un archivo comprimido `.zip` o `.tar.gz`, y debe indicar el número de su grupo siguiendo el patrón: L3-Grupo[N°Grupo].zip, Ejemplo: L3-Grupo01.zip
- Debe dejar un `MAKEFILE` o similar en cada máquina virtual asignada a su grupo para la ejecución de cada entidad.
- Debe dejar un `README` en cada máquina virtual asignada a su grupo con nombre y rol de cada integrante, además de la información necesaria para ejecutar los archivos.
- 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 5 puntos.
- Copias serán evaluadas con nota 0 y serán notificadas a los profesores y las autoridades pertinentes.

## 9. Consultas

Para hacer sus consultas recomendamos hacerlas en el foro del ramo en Aula. Sin embargo, en caso de algo personal o urgente, el correo es: `jorge.diazma@sansano.usm.cl`. Cabe destacar que se responderán consultas vía Aula y/o correo electrónico hasta 48 hrs antes de la fecha y hora de entrega (en este caso, hasta el 8 de diciembre a las 23:55 hrs).