Informe del Sistema Distribuido: Arquitectura, Funcionalidades y Procesos

Equipo RAFT

November 30, 2025

Abstract

Este informe describe el diseño, organización, y ejecución de un sistema distribuido basado en la arquitectura RAFT, que incluye la replicación de datos, manejo de eventos en tiempo real, y el uso de WebSockets para notificaciones. Se detallan las principales funcionalidades del sistema, los procesos involucrados, y las decisiones tomadas para garantizar la consistencia, disponibilidad, y seguridad del sistema. El informe también aborda aspectos relacionados con la tolerancia a fallos y la replicación de datos en un entorno distribuido.

Contents

1 Arquitectura del Sistema

El sistema distribuido se basa en una arquitectura **RAFT** que permite la replicación de datos entre varios nodos. La arquitectura se organiza en varios *shards*, cada uno con un líder y seguidores, siguiendo el principio de consenso de RAFT. El objetivo principal de la arquitectura es garantizar la consistencia de los datos en presencia de fallos, mediante la replicación de los logs de eventos.

El sistema también incluye un *layer* de notificaciones en tiempo real basado en el patrón **Pub/Sub** utilizando WebSockets. Cada usuario está suscrito a un canal de eventos de su interés, lo que permite una comunicación eficiente entre los usuarios del sistema.

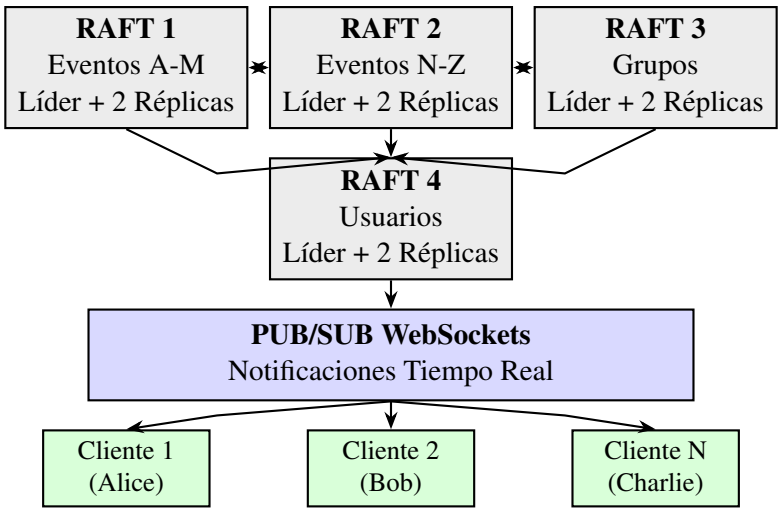

Figure 1: Arquitectura del Sistema Distribuido RAFT con WebSockets

2 Organización del Sistema Distribuido

El sistema distribuido se organiza en varios componentes clave:

- **Coordinador Inteligente:** Este servicio se encarga de decidir a qué *shard* va cada operación y balancea la carga entre los *shards*.
- **Shards Especializados:** Cada shard gestiona un subconjunto de los datos, como eventos, usuarios, y grupos. Cada shard sigue el protocolo RAFT con un líder y múltiples réplicas.
- **WebSocket Manager:** Administra las conexiones en tiempo real y las notificaciones push.

2.1 Detalle de los Shards

3 Roles del Sistema

Cada nodo en el sistema tiene un rol específico dentro del protocolo RAFT. Los roles incluyen:

Shard	Tipo -Datos	Particionamiento	Nodos
Shard 1	Eventos	A-M	3 (Líder + 2 Réplicas)
Shard 2	Eventos	N-Z	3 (Líder + 2 Réplicas)
Shard 3	Grupos	Todos	3 (Líder + 2 Réplicas)
Shard 4	Usuarios	Todos	3 (Líder + 2 Réplicas)

Table 1: Distribución de shards en el sistema

- **Líder:** El nodo que coordina la replicación de logs y garantiza la consistencia del sistema.
- **Seguidor:** Nodos que siguen al líder y replican sus logs.
- **Candidato:** Nodos que pueden ser elegidos como líderes si el líder actual falla.

4 Distribución de Servicios en Docker

El sistema distribuido está implementado en contenedores Docker. Cada grupo de RAFT se ejecuta en contenedores separados, con la comunicación entre los nodos facilitada por Docker Swarm.

4.1 Redes Docker

- **raft-network:** Red interna para comunicación entre nodos RAFT
- **api-network:** Red para comunicación frontend-backend
- **public-network:** Red para acceso externo de clientes

5 Procesos del Sistema

El sistema consta de varios procesos que trabajan en conjunto para garantizar el funcionamiento adecuado:

- **Proceso de Elección de Líder:** Los nodos RAFT eligen un líder para garantizar la consistencia de los datos.
- **Replicación de Logs:** El líder replica sus logs a los nodos seguidores.
- **Notificación en Tiempo Real:** Los clientes se suscriben a canales de eventos y reciben notificaciones a través de WebSockets.

6 Comunicación en el Sistema

El sistema utiliza varios tipos de comunicación:

- **RPC (Remote Procedure Call):** Usado en el protocolo RAFT para la replicación de logs y la elección de líder.
- **WebSockets:** Usados para la comunicación en tiempo real entre el servidor y los clientes.
- **REST API:** Usada para interactuar con la base de datos del sistema y para la creación y gestión de eventos.

6.1 Flujo de Comunicación

1. Cliente envía solicitud vía WebSocket/REST
2. Coordinador enruta al shard correspondiente
3. Líder del shard procesa y replica la operación
4. Confirmación enviada al cliente
5. Notificaciones broadcast a clientes interesados

7 Coordinación del Sistema

El sistema garantiza que todos los servicios estén sincronizados:

- **Sincronización de Acciones:** A través de RAFT, el líder asegura que las operaciones se apliquen en todos los nodos.
- **Acceso Exclusivo a Recursos:** Se asegura que solo un nodo líder pueda modificar los datos, evitando condiciones de carrera.

8 Tolerancia a Fallos

El sistema es resiliente a fallos de nodos:

- **Respuesta a Errores:** Si un nodo falla, otro nodo puede ser elegido como líder automáticamente.
- **Nivel de Tolerancia a Fallos:** El sistema puede tolerar la caída de un nodo por cada shard sin interrumpir el servicio.

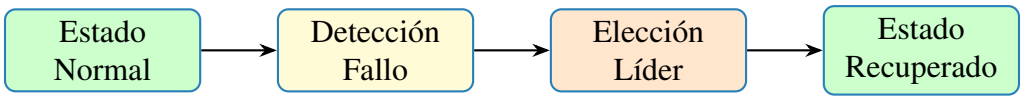

Figure 2: Proceso de recuperación ante fallos

8.1 Mecanismo de Failover

9 Seguridad

La seguridad del sistema se aborda en dos niveles:

- **Seguridad en la Comunicación:** Las comunicaciones entre nodos y clientes están protegidas mediante WebSockets seguros (wss://).
- **Autenticación y Autorización:** Los usuarios deben autenticar su sesión para interactuar con el sistema, y los accesos están restringidos según los roles (líder, miembro, etc.).

9.1 Medidas de Seguridad Implementadas

- Autenticación JWT para usuarios
- Encriptación TLS/SSL en todas las comunicaciones
- Validación de entrada en todos los endpoints
- Logs de auditoría para operaciones críticas