Informe sobre el Proyecto Distributed Spotify

Introducción

El proyecto Distributed Spotify tiene como objetivo desarrollar una versión minimalista de Spotify, centrada en la distribución eficiente de datos y la resiliencia del sistema. Este informe describe la arquitectura, organización, patrones de diseño, comunicación y coordinación del sistema, así como aspectos relacionados con consistencia, replicación, tolerancia a fallos y seguridad.

Arquitectura del Sistema

El sistema está compuesto por un frontend desarrollado en React y un backend basado en .NET, distribuidos en un entorno Docker con dos redes separadas: una para la comunicación interna entre servicios del backend y otra para la interacción con el frontend.

- **Diseño del sistema:** Se emplea una arquitectura de microservicios para el backend.
- **Roles del sistema:**
    - Frontend: Interfaz de usuario.
    - Backend: Lógica de negocio y acceso a datos.

Distribución de Servicios en Redes Docker

- **Red interna:** Conecta los microservicios del backend para comunicación segura.
- **Red externa:** Permite la interacción entre el frontend y el backend.

Cada microservicio se despliega en su propio contenedor Docker para asegurar independencia y escalabilidad.

Procesos del Sistema

- **Tipos de procesos:** Procesos asincrónicos para streaming y almacenamiento de canciones.
- **Agrupación:** Los procesos se agrupan en instancias según su funcionalidad para optimizar recursos.
- **Patrón de diseño:** Se emplea un enfoque asíncrono basado en hilos y tareas concurrentes (async/await).

Comunicación en el Sistema

- **Problema:** Enviar información de manera eficiente entre componentes distribuidos.
- **Tipo de comunicación:**
  - REST para interacción cliente-servidor así como para microservicios.
  - Patrones de mensajería para sincronización entre procesos.
  - Uso de patrón Outbox para asegurar la entrega de los mensajes.
- **Comunicación cliente-servidor:** El frontend realiza peticiones REST al backend.
- **Comunicación servidor-servidor:** Los microservicios del backend se comunican mediante HTTP y eventos distribuidos.

Nombrado y Localización

- **Identificación:** Cada recurso (canción, álbum, cantante) tiene un identificador único (GUID).
- **Ubicación:** Los datos se almacenan en servidores distribuidos con índices centralizados para facilitar su localización.
- **Localización:** Los servicios utilizan un registro de servicios para resolver la ubicación de otras componentes del sistema distribuido.

Consistencia y Replicación

- **Distribución de datos:** Las canciones se almacenan en servidores distribuidos.
- **Replicación:** Cada canción tiene copias en los demás servidores para garantizar disponibilidad.
- **Confiabilidad:** Uso de un token ring para asegurar las actualizaciones.

Tolerancia a Fallos

- **Respuesta a errores:** Los servicios registran y reintentan operaciones fallidas.
- **Nivel de tolerancia:** El sistema sigue funcionando con fallos parciales mediante redundancia.
- **Gestión de nodos:** Se integran nuevos nodos automáticamente, y los nodos caídos se reconfiguran dinámicamente.

Mecanismo de Líder Dinámico con IP Flotante

Para garantizar la continuidad del sistema, se implementa una estrategia de líder dinámico con IP flotante:

1. **IP Flotante:** Un nodo del sistema se designa como líder y se le asigna una dirección IP específica (IP del gateway).
2. **Monitorización del Líder:** Los nodos secundarios monitorizan el estado del líder mediante ping o consultas a un endpoint del líder.

3. **Cambio de Líder:** Si el líder falla:
    - Un nodo secundario asume el rol de líder.
    - Se reasigna la IP flotante al nodo que asume el rol.
    - El nodo que asume el rol de líder actualiza sus configuraciones de red:
        - Quitar su IP actual.
        - Asignarse la IP del líder fallido.

Automatización con Keepalived

El sistema también puede emplear herramientas como `keepalived` para gestionar automáticamente el cambio de líder y la reasignación de IPs sin intervención manual.

Seguridad

- **Seguridad en la comunicación:** Uso de cifrado TLS para todas las comunicaciones fuera de la red interna.
