

Plataforma de Streaming Musical

Cliente-Servidor (Parte II: Implementación en Red y Arquitectura)

Resumen

En este documento se desarrolla la Parte II del sistema de streaming musical. A diferencia de la Parte I, orientada a un modelo académico simplificado, aquí se plantea una solución real basada en una arquitectura cliente-servidor distribuida. El servidor expone servicios a través de la red y el cliente consume dichos servicios remotamente. El enfoque aplicado sigue criterios de ingeniería de software: modularidad, concurrencia, seguridad, tolerancia a fallos, escalabilidad, trazabilidad, pruebas y despliegue controlado.

Arquitectura Global del Sistema

El sistema se estructura en tres capas principales:

- **Capa de Presentación (Cliente):** interfaz de usuario y control de reproducción.
- **Capa de Lógica de Negocio (Servidor):** control de usuarios, licencias, distribución de audio, auditoría y recomendaciones.
- **Capa de Persistencia:** almacenamiento persistente del catálogo, estadísticas y usuarios.

El cliente y el servidor se comunican mediante red utilizando sockets TCP para control y, opcionalmente, RTP/UDP para el transporte eficiente del flujo de audio.

Modelo Cliente-Servidor en Red

Se definen los siguientes elementos:

Servidor Central

- Acepta conexiones entrantes.
- Mantiene sesiones activas.
- Autentica usuarios.
- Controla el acceso al catálogo.
- Envía los flujos de audio solicitados.
- Registra actividad en logs.
- Maneja concurrencia mediante sincronización.

Cliente Distribuido

- Se conecta al servidor mediante sockets TCP para control.
- Solicita canciones.
- Recibe el flujo de audio mediante RTP/UDP o TCP.
- Gestiona la reproducción local (play, pausa, detener, etc.).
- Presenta la información al usuario.

Protocolo de Comunicación

Las comunicaciones siguen un formato de mensajes normalizado. Cada petición enviada al servidor incluye:

- Tipo de operación.
- Token de sesión.
- Parámetros asociados.

Las respuestas contienen:

- Código de estado.
- Mensaje descriptivo.
- Datos opcionales.

Se definen operaciones como:

- LOGIN
- CONSULTAR_CATALOGO
- SOLICITAR_CANCION
- INICIAR_STREAMING
- PAUSA
- CONTINUAR
- DETENER
- DESCONECTAR

Concurrencia y Control de Sesiones

El servidor crea un hilo independiente por cada cliente conectado, de forma que múltiples usuarios pueden utilizar la plataforma simultáneamente.

Cada cliente obtiene un identificador de sesión. La sesión expira por inactividad, cierre voluntario o desconexión inesperada. Los recursos compartidos se protegen mediante mecanismos de sincronización para evitar condiciones de carrera.

Un cliente no autenticado no puede acceder a recursos protegidos.

Streaming de Audio en Red

El proceso de reproducción funciona de la siguiente manera:

- El cliente selecciona una canción.
- El cliente envía una solicitud de streaming al servidor.
- El servidor localiza el fichero de audio correspondiente.
- El servidor envía el flujo de datos al cliente.
- El cliente reproduce el audio localmente y controla la reproducción.

Para el transporte del audio se puede emplear:

- **TCP**: transmisión fiable.
- **RTP/UDP**: menor latencia y mayor fluidez.

El **control de la reproducción (play, pausa, detener)** reside en el cliente, que envía las órdenes oportunas al servidor cuando es necesario modificar el flujo de datos.

Persistencia

Los datos persistentes incluyen:

- catálogo de canciones,
- usuarios,
- licencias,
- logs,
- estadísticas de uso.

Se garantiza consistencia y recuperación tras reinicio del servidor.

Seguridad y Autenticación

Se implementan las siguientes medidas:

- Control de credenciales.
- Roles (ADMIN, STANDARD).
- Tokens de sesión.
- Bloqueo tras intentos fallidos.
- Control parental.
- Validación de contenido explícito.

Los mensajes se validan antes de su procesamiento para evitar ataques por datos mal formados.

Sistema de Auditoría y Logs

Cada evento relevante se almacena en un registro estructurado en el servidor:

- conexiones,
- reproducciones,
- cambios de estado,
- incidencias y errores.

Los logs permiten analizar el comportamiento del sistema y detectar anomalías.

Sistema de Recomendaciones Avanzado

Las recomendaciones se generan mediante:

- análisis del historial del usuario,
- popularidad global,
- afinidad por artista,
- similitud entre canciones.

El algoritmo se ejecuta exclusivamente en el servidor.