

Proyecto Ana Java – Plataforma de Streaming Musical Cliente-Servidor (Parte I)

Introducción

En este proyecto se implementa un sistema en Java que simula una plataforma de streaming musical cliente-servidor. El sistema reproduce el comportamiento de plataformas reales como Spotify, pero empleando únicamente los conceptos fundamentales del lenguaje Java: encapsulación, clases, arrays, enumeraciones, métodos estáticos, manejo de excepciones y composición de objetos. La comunicación cliente-servidor se modela mediante llamadas a métodos entre objetos Java.

Objetivos

- Diseñar una arquitectura modular cliente-servidor.
- Implementar reproducción de música con control de tiempo.
- Gestionar usuarios, playlists, historial, licencias y seguridad.
- Monitorizar estadísticas globales.
- Implementar control parental y auditoría.
- Aplicar arrays, enums, clases y métodos estáticos.
- Controlar errores mediante excepciones.

Arquitectura General

El sistema consta de dos grandes bloques:

- **ServidorStreaming**: almacena canciones, usuarios, licencias, logs y estadísticas.
- **ClienteStreaming**: gestiona la sesión del usuario y la reproducción.

El servidor centraliza toda la lógica crítica. El cliente únicamente accede a través de métodos autorizados.

Enumeraciones del Sistema

RolUsuario

- ADMIN
- STANDARD

EstadoServidor

- ACTIVO
- MANTENIMIENTO
- CAIDO

EstadoReproductor

- REPRODUCIENDO
- PAUSADO
- DETENIDO

Clases del Servidor

Clase Cancion

Representa cada pista musical. Controla:

- título
- artista
- duración
- contador de reproducciones
- indicador de contenido explícito

Clase UsuarioServidor

Contiene:

- nombre
- contraseña
- rol
- estado activo/bloqueado

Clase Licencia

Controla:

- duración de suscripción
- inicio
- fin
- estado activa

Clase LogServidor

Registra eventos en arrays:

- accesos
- reproducción
- errores

Clase ServidorStreaming

Es el núcleo del sistema.

Atributos principales:

- catálogo de canciones
- array de usuarios
- logs
- licencias activas
- contador global de reproducciones (estático)
- usuarios conectados (estático)
- estado del servidor

Funciones esenciales:

- registrarUsuario()
- login()
- registrarCancion()
- solicitarCancion()
- validarLicencia()
- comprobarContenidoExplicito()
- generarRecomendaciones()

- `aplicarRateLimit()`
- `registrarReproduccionGlobal()`
- `resumenEstadisticas()`

Componentes del Cliente

Clase `Playlist`

- modo colaborativo
- arrays sin duplicados

Clase `HistorialCircular`

Almacena las últimas canciones reproducidas.

Clase `ClienteStreaming`

Controla:

- sesión
- playlist
- canción actual
- estado del reproductor
- posición temporal

El método `tic()` simula un segundo real.

Gestión de Errores

Se usan excepciones para:

- intentar reproducir sin login
- intentar reproducir con servidor caído
- intentar reproducir contenido explícito con control parental activo
- alcanzar límite de uso

Sistema de Recomendación

Basado en:

- historial
- canciones más reproducidas

Estadísticas Globales

- canción más escuchada
- número de usuarios activos
- tiempo total reproducido
- reproducciones totales

Auditoría

Cada evento se registra en `LogServidor`.

Ejercicio 2: Simulación Completa

La clase `Main` debe:

1. Crear el servidor.
2. Registrar canciones.
3. Crear usuarios (incluyendo admin).
4. Aplicar licencias.
5. Activar control parental.
6. Iniciar sesión desde el cliente.
7. Crear playlist.
8. Reproducir varias canciones.
9. Pausar y reanudar.
10. Mostrar historial.
11. Mostrar estadísticas del servidor.
12. Registrar logs.

Se simularán múltiples sesiones.