

Clase 06: Ejercicio - Reproductor Multimedia

Propósito: Practicar el diseño de una jerarquía de clases a partir de **funcionalidades** (no desde la estructura). Se espera que vayas **introduciendo interfaces (traits), clases abstractas e implementaciones concretas** de forma incremental.

1. Contexto

Una aplicación necesita manipular “medios” (audio y video) que pueden **reproducirse** y **buscar posiciones** (seek). Además, algunos medios **requieren buffer** (p.ej., streaming) antes de poder reproducir/avanzar con normalidad.

La empresa también quiere un reproductor que pueda **reproducir** un medio, **detenerlo**, **cambiar su posición de reproducción** y **mostrar información básica** (título y duración). Dicho reproductor debe ser capaz de realizar estas acciones sobre cualquier objeto que ofrezca las capacidades necesarias.

2. Objetivos de aprendizaje

1. Derivar **interfaces y clases** a partir de **requisitos funcionales**.
2. Separar responsabilidades entre **medio** (estado del contenido) y **player** (opera sobre el contrato recibido).
3. Introducir **clases abstractas** para factorizar lógica común y **hooks** simples para extensión.
4. Añadir un caso de **buffer** con política de búsqueda controlada.

3. Requisitos

3.1. Reproductores

3.1.1. Basic Player

- Debe **reproducir, detener, cambiar la posición y mostrar información** de un medio.
- Ejecuta las acciones **sin mensajes adicionales** propios; solo se esperan los mensajes naturales de cada operación del medio (p. ej., reproducir o detener).

3.1.2. Verbose Player

- Debe **reproducir, detener, cambiar la posición y mostrar información** de un medio.
- **Antes y después** de cada operación imprime un mensaje de registro.

3.2. Medios básicos

3.2.1. Audio

- Debe poder **reproducirse, detenerse y cambiar su posición**.
- Expone **título y duración total**.
- Al reproducir, informa al usuario el **título** y la **posición actual**.

3.2.2. Video

- Igual que Audio, y además expone **ancho y alto** (resolución).
- Al reproducir, informa al usuario el **título**, la **posición actual** y la **resolución** (p. ej., 3840x2160).

3.3. Posicionamiento y límites

3.3.1. Seek y consulta

- Debe existir una operación para **cambiar la posición** en milisegundos.
- Debe exponer la **posición actual** y la **duración total**.
- La nueva posición siempre se **limita** al rango $[0, \text{duración}]$.

3.4. Variantes Local y Streaming

3.4.1. Local

- El contenido está **completamente disponible** desde el inicio.
- El posicionamiento puede alcanzar **cualquier valor** dentro de la duración.

3.4.2. Streaming

- Mantiene una cantidad de tiempo **bufferizado** (en milisegundos).
- El posicionamiento solo puede avanzar hasta el **tiempo bufferizado**.
- Al reproducir, debe **garantizar al menos 3000 ms** bufferizados por delante de la posición actual.
- Debe existir una operación para **simular** la carga de buffer **hasta un objetivo** e **informar el avance**.

3.4.3. Variantes requeridas

- Implementar: **Audio Local, Audio Streaming, Video Local, Video Streaming**.