

Arquitectura (35)			Modelo de Dominio (40)				Persistencia (25)		Final
A (10)	B (15)	C (10)	A (15)	B (10)	C (10)	D (5)	A (15)	B (10)	

Apellido y Nombre:Legajo:

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.

VideoTracker

Contexto

VideoTracker es una empresa que ofrece servicios de análisis del comportamiento de los usuarios al reproducir videos en línea. A través de un plugin integrado en los reproductores de video de los sitios que contratan el servicio, se recopila información en tiempo real sobre la interacción del espectador: en qué momento del video se encuentra, cuántas veces pausó la reproducción, cuántas veces cambió de pestaña, entre otros datos relevantes. Con esta información, se generan análisis detallados que permiten, a los creadores de contenido y plataformas, evaluar la recepción de sus videos, optimizar la producción y mejorar la experiencia de la audiencia.

Publicadores de contenido

Los publicadores de contenido son organizaciones o personas que crean y exponen contenido (videos) y que quieren saber cómo son visualizados por los usuarios. Nuestro Sistema procesa la información obtenida y ofrece un informe automático con distintas propiedades sobre los videos. Por ejemplo, un cliente podría ser la cátedra de DDS, quien podría generar videos y quiere ver la recepción de los alumnos.

Visualizadores de contenido

Los visualizadores de contenido son los usuarios que ven los videos generados por los publicadores. Para registrar su interacción, el reproductor de video en la plataforma del publicador debe contar con un plugin que envía datos de seguimiento cada pocos segundos a nuestro sistema. Luego, estos datos son procesados para analizar el comportamiento de los espectadores.

Por ejemplo, los alumnos pueden ver videos en Webcampus, donde el reproductor cuenta con el plugin que permite el seguimiento.

Información de seguimiento

La información de seguimiento consiste en el **user_id** para poder identificarlo univocamente, un **video_id** con el mismo propósito, **timestamp** del día, **momento** en el video, e información asociada por ejemplo a cuántas veces se pausó en ese tiempo, a qué velocidad va el video, o cuántas veces se cambió de pestaña mientras se visualizaba.

Visualización y propiedades

Para este dominio, una **"visualización"** es el tiempo que una persona dedica a ver un video. Un usuario puede ver un video de principio a fin en una sola sesión a velocidad normal, mientras que otro puede pausarlo y retomarlo en distintos momentos. Incluso, puede reproducirlo en paralelo con otras actividades, como leer un periódico en línea. Las propiedades de estas visualizaciones se calculan a partir de la información de seguimiento. Actualmente, se consideran dos propiedades principales: **"atención"** y **"continuidad"**. La lógica de su cálculo es compleja, pero independiente del resto del sistema.

Análisis de seguimiento

Consiste en seleccionar un conjunto de usuarios y videos y procesar las correspondientes visualizaciones, para obtener un resultado por video.

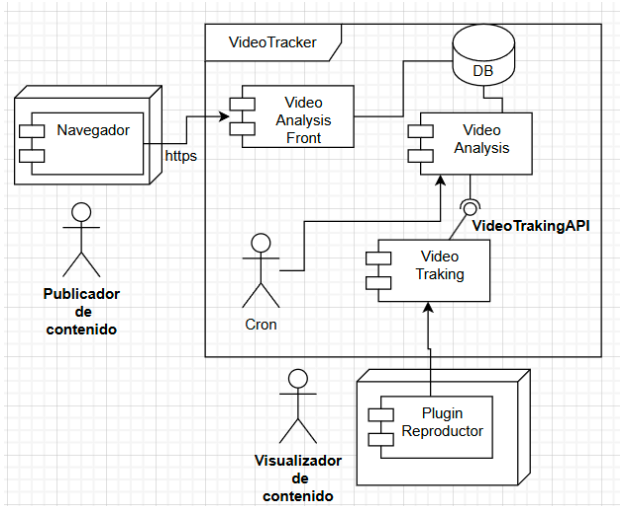
Resultado del análisis de un video

Una vez obtenidas las visualizaciones y sus propiedades, se agrupan bajo ciertos criterios, como el promedio de cada propiedad. Sin embargo, como estas propiedades son categóricas, la forma en que se agrupan es parte de la lógica de negocio del sistema. Esto significa que se debe definir una metodología específica para interpretar los datos. Por ejemplo, si todos los alumnos ven un video de principio a fin sin pausas, la propiedad "continuidad" se categorizará como

"unaTanda". Si algunos lo ven en tres partes, otros en dos y otros de una sola vez, la clasificación seguirá una lógica predefinida dentro del dominio.

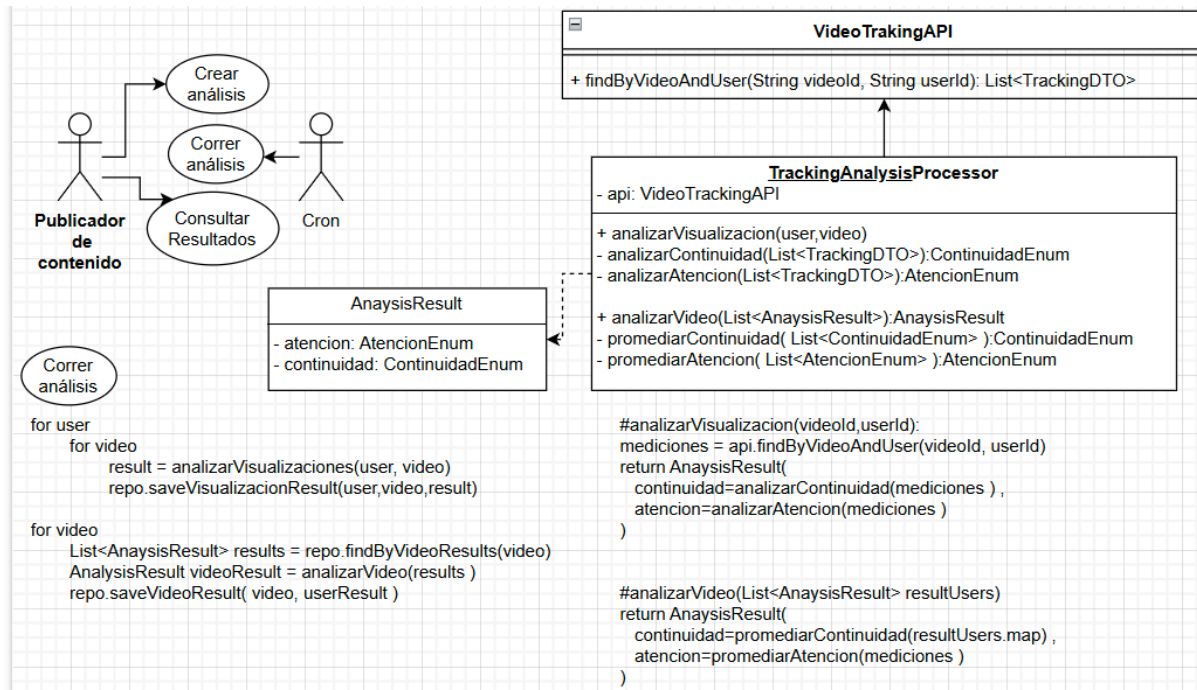
Por otro lado, lo anterior también significa que los mismos valores de las propiedades de visualización se usan para el resultado del análisis de un video.

Arquitectura Actual



- **Video Analysis Front** sirve para dar de alta los análisis y visualizar los resultados. Está implementado con un cliente liviano / render server.
- **Video Analysis** es NUESTRO componente, tiene como responsabilidad implementar la lógica de las visualizaciones y el cálculo de propiedades.
- **Video Tracking** se encarga de recibir la información de seguimiento y almacenarla. Aún desconocemos cómo va a funcionar porque pertenece a otro equipo, solo sabemos que nos provee una API de consulta.
- El **Cron** se encarga de llamar cada 24hs al procesamiento, que consiste en un job que toma todos los análisis solicitados en el día y los procesa para obtener los resultados de los videos.

Implementación Actual de VideoAnalysis (NUESTRO COMPONENTE)



Importante

- La implementación actual FUNCIONA
- VideoTrackingAPI y TrackingDTO forman parte de la SDK provista por el equipo de VideoTracking.

Punto 1 – Arquitectura (35 puntos)

Actualice el diagrama de Despliegue del enunciado (y realice cualquier diagrama o explicación auxiliar que necesite) teniendo en cuenta, y contestando, los siguientes puntos:

1. **(10 puntos)** Del equipo de Video Analysis Front, están analizando pasar a una arquitectura de cliente pesado, y generar una Single Page Application para la visualización de resultados. Dado que no se prevén otros casos de uso más que los que se representan más arriba, ¿cree que es el camino indicado? ¿Por qué? Justifique su respuesta.
2. **(15 puntos)** Actualmente el trabajo que procesa todos los análisis comienza a ejecutar las 12 de la noche y tarda algunas de horas en terminar. Sin embargo, en ciertas ocasiones ha consumido más de 24hs. Esos días se detectó que se realizaron más análisis de los soportados, pero contratar instancias (o comprar hardware) más caro que el que se está contratando supondría un salto muy grande en los costos. ¿Cómo puede resolver esta situación? GRAFIQUE y explique su funcionamiento muy brevemente.
3. **(10 puntos)** Del equipo de VideoTracking están debatiendo ideas de cómo el plugin se debe integrar con ellos. Un colega propone que el plugin almacene todas las mediciones y cuando el usuario está por abandonar la página, las envíe todas juntas. De esta forma se ahorran varias idas y vueltas al servidor. Teniendo en cuenta los atributos de calidad “*tolerancia a fallos*” y “*eficiencia*”, compare la alternativa de acumular todos los resultados hasta el final vs ir enviando cada N segundos/minutos.

Punto 2 - Modelado de dominio (40 puntos)

La implementación que se tiene actualmente funciona, sin embargo:

1. Se quieren agregar más propiedades al análisis. Las mismas se deben poder incorporar lo más fácilmente posible.
 2. Los análisis deben indicar qué propiedades se quieren calcular, es decir, no siempre se quieren calcular todas.
 3. El equipo de VideoTracking, aún no tiene algunas definiciones y puede cambiar su firma. Por otro lado, también se está evaluando fuertemente la posibilidad de contratar un servicio externo que provea la misma funcionalidad.
- A. **(15 puntos)** Realice los diagramas que considere necesarios (diagrama de clases obligatorio) para realizar un refactor del **dominio** del componente **VideoAnalysis** que incorpore los puntos planteados.
 - B. **(10 puntos)** Explique cómo se agregaría una propiedad “**Ansiedad**” la cual procesa las mediciones y, entre otras cosas, obtiene la velocidad de reproducción de cada trackeo.
 - C. **(10 puntos)** Explique qué pasos se deberían seguir para reemplazar al componente **VideoTracking**.
 - D. **(5 puntos)** Explique cómo se aplicó el concepto Open-Close / Abierto-Cerrado a los puntos B y C.

Punto 3 – Datos (25 puntos)

Actualmente se está persistiendo en una base documental, pero se decidió cambiar a un esquema relacional. Diseñar el modelo de datos del componente al sistema para poder persistir en una base de datos relacional a través de un ORM.

- A. **(15 Puntos)** Armar la especificación usando un DER físico. Indicando las entidades, sus campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(10 puntos)** Justificar
 - Qué elementos del modelo es necesario persistir y cuáles no (**atención con esto**)
 - Cómo resolvió los impedance mismatches.
 - Las estructuras de datos que deban ser desnormalizadas, si corresponde.

