

Video Recording

Informe estado del proyecto Final Segundo Corte

Integrantes

- Mayerli Paola Bruno Aguirre
- Arturo Flores Causil
- César Luis Briceño Ojeda
- José Álvarez Ortega
- Yerson David Moreno Zapata

¡Excelente pregunta! Profundizar en la justificación de la elección de la API nativa **MediaRecorder** en lugar de librerías externas es importante para entender las decisiones técnicas del proyecto. Aquí te presento un informe de las razones principales:

Informe: Justificación de la Selección de la API Nativa MediaRecorder

Introducción:

Este informe detalla las razones fundamentales por las cuales el equipo de desarrollo optó por utilizar la API nativa del navegador **MediaRecorder** para la implementación de la funcionalidad de grabación de pantalla en el proyecto Video Recording, en lugar de recurrir a librerías de terceros.

Razones Principales para la Selección de MediaRecorder:

1. Naturaleza Nativa y Eliminación de Dependencias:

- La API **MediaRecorder** es una funcionalidad inherente a los navegadores web modernos. Su uso elimina la necesidad de incluir librerías externas al proyecto.
- **Ventajas:**
 - **Reducción del tamaño del paquete:** Al no depender de librerías adicionales, se disminuye el tamaño total de la aplicación, lo que puede traducirse en tiempos de carga más rápidos y un menor consumo de recursos por parte del usuario.
 - **Menor riesgo de vulnerabilidades:** La dependencia de librerías de terceros introduce un riesgo potencial de seguridad si estas librerías contienen vulnerabilidades. Al utilizar una API nativa, se reduce esta superficie de ataque.
 - **Mayor estabilidad a largo plazo:** Las APIs nativas suelen tener un ciclo de vida más largo y una mejor compatibilidad con las futuras

actualizaciones de los navegadores en comparación con las librerías de terceros, que pueden quedar obsoletas o sin mantenimiento.

- **Simplicidad en la gestión de dependencias:** Se evita la complejidad de gestionar versiones, actualizaciones y posibles conflictos entre diferentes librerías.

2. Funcionalidad Específica y Eficiente para la Tarea:

- La API **MediaRecorder** está diseñada específicamente para la captura y codificación de streams de medios (audio y/o video) directamente desde el navegador.
- **Ventajas:**
 - **Optimización para la grabación:** Al ser una API nativa, está optimizada para interactuar con los dispositivos de captura del sistema (pantalla, micrófono, cámara) de manera eficiente.
 - **Control granular sobre la grabación:** Ofrece un control adecuado sobre los parámetros de grabación, como el tipo de códec (aunque la compatibilidad puede variar entre navegadores), la tasa de bits y el formato de salida (en este caso, **.webm**).
 - **Integración directa con otras APIs nativas:** Se integra de manera fluida con otras APIs del navegador como **mediaDevices.getDisplayMedia()** (para la captura de pantalla) y la API **Blob** (para el manejo de los datos del video grabado).

3. Compatibilidad con Navegadores Modernos:

- La API **MediaRecorder** cuenta con un soporte amplio en los navegadores modernos (Chrome, Firefox, Safari, Edge). Dado que el proyecto se enfoca en la funcionalidad dentro del navegador, esta compatibilidad es crucial.
- **Consideraciones:** Si el proyecto tuviera como objetivo una compatibilidad extensiva con navegadores muy antiguos, las librerías podrían ofrecer soluciones con polyfills o implementaciones alternativas. Sin embargo, para un proyecto que prioriza las capacidades actuales de los navegadores, **MediaRecorder** es una opción sólida.

4. Generación Directa del Formato Deseado (.webm):

- La API **MediaRecorder** permite configurar el tipo de **mimeType** para la grabación, y en este caso, genera directamente archivos en formato **.webm**.
- **Ventajas:**
 - **Simplicidad en la descarga y reproducción:** El formato **.webm** es compatible con la mayoría de los reproductores de video modernos y navegadores, lo que facilita la descarga y visualización sin necesidad de conversiones adicionales en el lado del cliente.

Posibles Consideraciones sobre Librerías Externas (y por qué MediaRecorder fue preferible):

Si bien existen librerías de terceros que facilitan la grabación de video en el navegador, a menudo introducen:

- **Mayor complejidad:** Pueden abstraer la API nativa, pero también pueden añadir una capa adicional de configuración y posibles puntos de fallo.
- **Mayor tamaño:** Suelen ser más pesadas que utilizar directamente la API nativa, lo que afecta el rendimiento de la aplicación.
- **Posibles costos (en algunos casos):** Algunas librerías comerciales pueden tener costos asociados.

Conclusión:

La decisión de utilizar la API nativa **MediaRecorder** para la funcionalidad de grabación de pantalla se basó en los beneficios significativos en términos de eliminación de dependencias, eficiencia, compatibilidad con navegadores modernos y la capacidad de generar directamente el formato de archivo deseado. Esta elección simplifica la arquitectura del proyecto, reduce su tamaño, minimiza los riesgos de seguridad asociados con dependencias externas y aprovecha las capacidades intrínsecas de los navegadores web actuales. El equipo consideró que las ventajas de utilizar **MediaRecorder** superaban los posibles beneficios que podrían ofrecer las librerías de terceros para este caso de uso específico.

