

1. Cambio GUI para que muestre los videos en lugar de descargarlos

Con la ayuda de la IA se le ha pedido que cambie la API de un endpoint para que muestre el video en lugar de descargarlo. Después de revisar los cambios y adaptarlos un poco para que esté a nuestro gusto (básicamente cambios de formato), se le ha pasado el código de la GUI de todos los endpoints para que replique la manera de mostrar el video.

Ejemplo de GIT DIFF de la GUI del endpoint de modificación de la resolución:

```
Final\front_end\seminario2\modificador_resolución.html
@@ -6,7 +6,6 @@
6      6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      7      <title>Modificador resolución</title>
8      8      <link rel="icon" href="../images/logo.png" type="image/png">
9      -
10     9      </head>
11     10     <body>
12     11     <h1>MODIFICADOR RESOLUCIÓN</h1>
@@ -14,9 +13,9 @@
14     13     <h2>Este endpoint genera una versión del vídeo en la resolución deseada</h2>
15     14     <hr>
16     15     <h2>Selecciona el vídeo al que deseas cambiar la resolución</h2>
17     - <form action="http://localhost:5001/resolution_changer" method="POST" enctype="multipart/form-data"
18     + <form id="resolutionForm" enctype="multipart/form-data">
19     16     <label for="file" class="file-label">Seleccionar vídeo</label>
20     17     <input type="file" id="file" name="file" required><br><br>
21     18     <input type="file" id="file" name="file" required><br><br>
22     19     <p id="file-name" style="display:none"></p>
@@ -28,57 +27,75 @@
28     27     <label for="ancho">Ancho:</label>
29     28     <input type="number" id="ancho" placeholder="Ej: 1280" required><br><br>
30     29
31     - <input type="hidden" id="data" name="data">
32     - <h2>Pulsa el botón para descargar el vídeo</h2>
33     - <button type="submit">Descargar</button>
```

```

33 + <hr>
34 + <div>
35 +     <h2>Vista previa del video procesado:</h2>
36 +     <video id="videoResult" controls style="display: none; max-width: 100%; max-height: 80vh; borde
r: 2px solid #ccc; border-radius: 8px;">
37 +     </video>
38 + </div>
39 +
40 <script>
- // Obtener el input del archivo y el elemento para mostrar el nombre
41 + // Mostrar nombre del archivo seleccionado
42 const fileInput = document.getElementById('file');
43 const fileNameDisplay = document.getElementById('file-name');
-
- // Escuchar el evento de cambio del input de archivo
- fileInput.addEventListener('change', function() {
- // Verificar si se seleccionó un archivo
44 + fileInput.addEventListener('change', function () {
45     if (fileInput.files.length > 0) {
- // Mostrar el nombre del archivo seleccionado
46     fileNameDisplay.textContent = `Archivo seleccionado: ${fileInput.files[0].name}`;
- fileNameDisplay.style.display = 'block'; // Hacer visible el texto
47 + fileNameDisplay.style.display = 'block';
48     }
49 });

```

```

50 - </script>
51 -
52 - </body>
53 50
51 + // Manejar el envío del formulario
52 + document.getElementById('resolutionForm').addEventListener('submit', async function (event) {
53 +     event.preventDefault(); // Evitar el envío tradicional del formulario
54 54
55 - <script>
56 -     function prepararJSON(event) {
57 -         // Verificar si el archivo está vacío (ningún archivo seleccionado)
58 -         const fileInput = document.getElementById('file');
59 -         if (!fileInput.files.length) {
60 -             alert('Por favor, selecciona un video.');
```

```

61 -             event.preventDefault(); // Detiene el envío del formulario
62 -             return false;
63 -         }
55 + // Validar el archivo
56 +         if (!fileInput.files.length) {
57 +             alert('Por favor, selecciona un video.');
```

```

58 +             return;
59 +         }
64 60
65 - // Obtener los valores de los campos "Alto" y "Ancho"
66 - const alto = document.getElementById("alto").value;
67 - const ancho = document.getElementById("ancho").value;
61 + // Obtener valores de ancho y alto
62 + const alto = document.getElementById('alto').value;
63 + const ancho = document.getElementById('ancho').value;

```

```

69 - // Validar que ambos valores sean números válidos
70 - if (!alto || !ancho || isNaN(alto) || isNaN(ancho)) {
71 -     alert("Por favor, ingresa valores numéricos válidos para Alto y Ancho.");
72 -     event.preventDefault(); // Detiene el envío del formulario
73 -     return false;
74 - }
65 + // Validar valores
66 + if (!alto || !ancho || isNaN(alto) || isNaN(ancho)) {
67 +     alert('Por favor, ingresa valores numéricos válidos para Alto y Ancho.');
```

```

68 +     return;
69 + }
75 70
76 - // Crear el JSON con el alto y el ancho
77 - const stringData = `{"Alto": ${alto}, "Ancho": ${ancho}}`;
78 - document.getElementById("data").value = stringData;
79 - return true;
80 - }
71 + // Crear FormData para enviar al servidor
72 + const formData = new FormData();
73 + formData.append('file', fileInput.files[0]);
74 + formData.append('data', JSON.stringify({ Alto: parseInt(alto), Ancho: parseInt(ancho) }));
81 75
82 - </script>
76 + try {
77 +     // Enviar solicitud al servidor
78 +     const response = await fetch('http://localhost:5001/resolution_changer', {
79 +         method: 'POST',
80 +         body: formData
81 +     });
```

```

83 +     if (!response.ok) {
84 +         throw new Error('Error al procesar el video.');
```

```

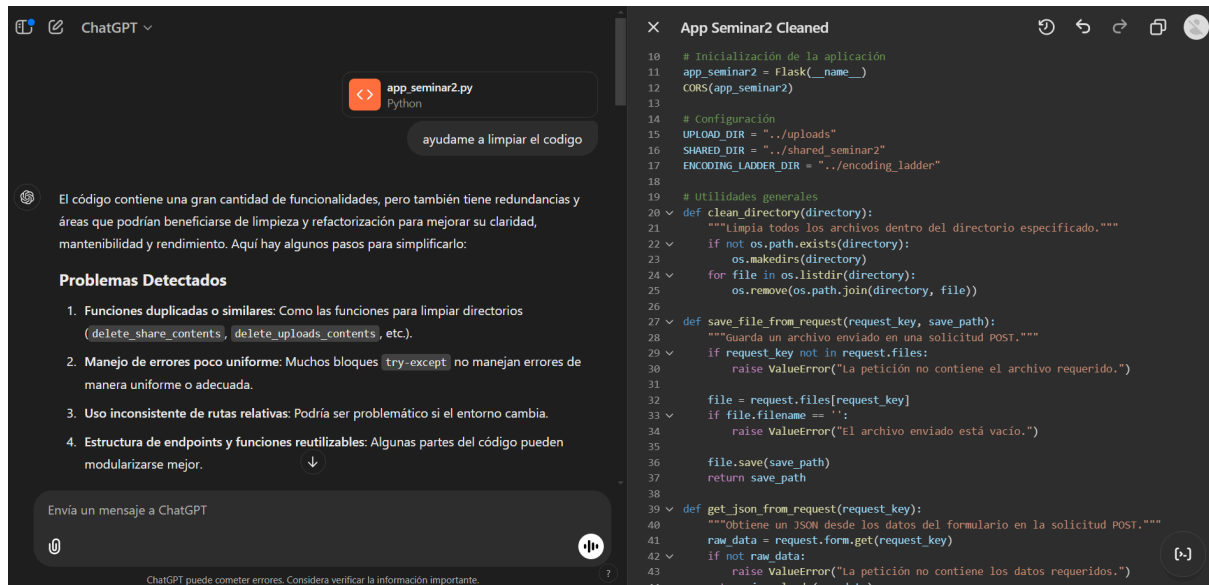
85 +     }
86 +
87 +     // Crear un blob del video resultante y mostrarlo
88 +     const videoBlob = await response.blob();
89 +     const videoUrl = URL.createObjectURL(videoBlob);
90 +
91 +     const videoElement = document.getElementById('videoResult');
92 +     videoElement.src = videoUrl;
93 +     videoElement.style.display = 'block'; // Mostrar el video
94 + } catch (error) {
95 +     console.error(error);
96 +     alert('Ocurrió un error al procesar el video. Por favor, inténtalo nuevamente.');
```

```

97 + }
98 + });
99 + </script>
100 + </body>
84 101 </html>
```

2. Corrección app_seminar2.py

Se le ha pasado el archivo app_seminar2.py a la IA y se le ha pedido que limpie y optimice el resultado.



“El código contiene una gran cantidad de funcionalidades, pero también tiene redundancias y áreas que podrían beneficiarse de limpieza y refactorización para mejorar su calidad, mantenibilidad y rendimiento. Aquí hay algunos pasos para simplificarlo:

Problemas Detectados

1. **Funciones duplicadas o similares:** Como las funciones para limpiar directorios (`delete_share_contents`, `delete_uploads_contents`, etc.).
2. **Manejo de errores poco uniforme:** Muchos bloques `try-except` no manejan errores de manera uniforme o adecuada.
3. **Uso inconsistente de rutas relativas:** Podría ser problemático si el entorno cambia.
4. **Estructura de endpoints y funciones reutilizables:** Algunas partes del código pueden modularizarse mejor.

Propuesta de Limpieza

1. **Consolidar funciones similares:**
 - Crear una única función para limpiar directorios.
2. **Abstraer lógica repetitiva:**
 - Simplificar el manejo de archivos y JSON dentro de una función reutilizable.
3. **Mejorar el manejo de errores:**

- Usar una función central para devolver errores o manejar excepciones.
4. **Agregar documentación y mejorar nombres de variables:**
- Hacer que los nombres sean más descriptivos.”

Dicho esto, ha procedido a aplicar la propuesta.

El resultado ha reducido bastante el número de líneas, concretamente la nueva versión del código tiene **417 líneas menos**. También hay que tener en cuenta que ha eliminado todos los comentarios que consideraba redundantes (casi todos). Sin embargo, se han tenido que hacer varias iteraciones para que el código funcione adecuadamente.

1. No había tenido en cuenta que las rutas relativas para los contenedores y para devolver los archivos con el POST no son las mismas. Para corregir esto ha generado una función que modifica la ruta para que sea correcta.
2. No ha gestionado bien los endpoints de la información del video ni del número de pistas de audio. Al parecer no los entiende bien, hemos decidido mantener los antiguos adaptándolos para que funcionen correctamente con el otro código generado de la IA.
3. Los endpoints de la práctica 2 tampoco funcionan correctamente, así que hemos tenido que recuperar nuestro código.

Se ha mantenido el código antiguo en el archivo `app_seminar2_viejo.py`.