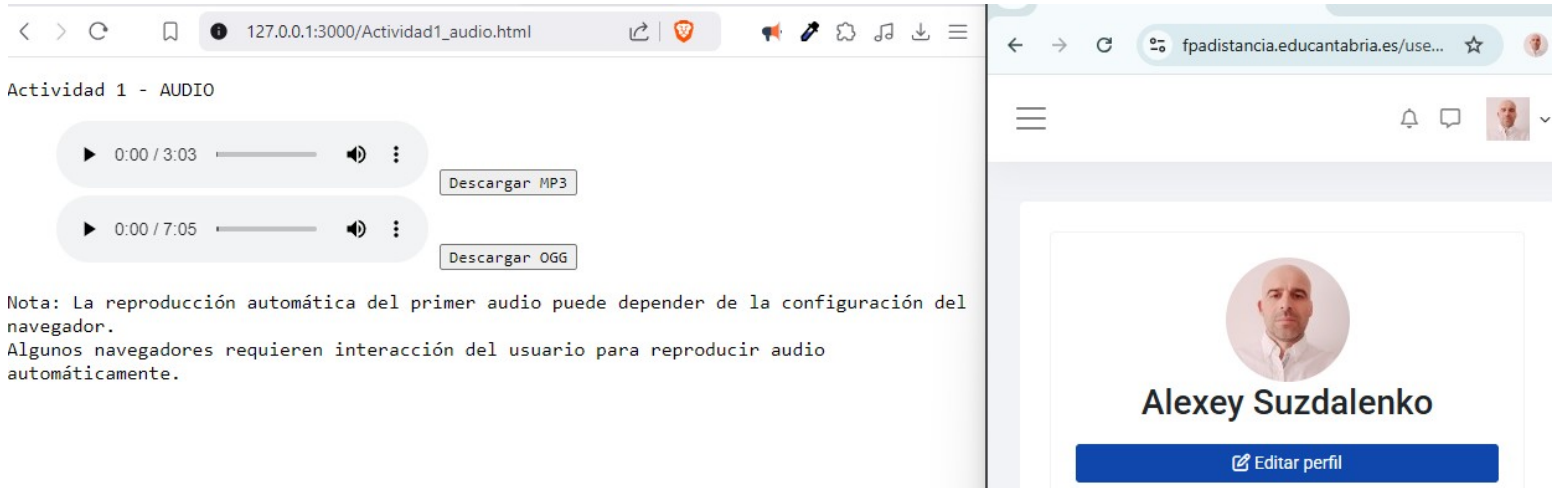


Los archivos “1.mp4” de la tercera actividad y “out.mp4” de la actividad 4 he tenido que alojarlos en nube y en el código indico sus url por que de lo contrario la aplicación de fpadistancia no me dejaba subir la tarea ya que el máximo volumen permitido es 20MB y los 2 archivos de video suman bastante mas.

Entonces ya que los archivos “1.mp4” y “out.mp4” están en la nube, no trabaja correctamente el botón de descargar archivo, ya que en vez de descargarlo le abre en una pestaña y luego hay que proceder a descargarle. Los archivos alojados localmente como por ejemplo “2.ogg” se descargan correctamente.

Actividad 1 - AUDIO

He encontrado en internet 2 audios y he echo la tarea de acuerdo con lo que pide el anunciado, el resultado en el navegador (a través de un servidor de desarrollo http://127.0.0.1:3000/Actividad1_audio.html) captura:



El código correspondiente es:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Actividad 1</title>
</head>

<body>
  <p>Actividad 1 - AUDIO</p>
  <ul>
    <li>
      <!-- primer audio con reproduccion automatica MP3 -->
      <audio controls autoplay>
        <source src="1.mp3" type="audio/mpeg">
        Tu navegador no soporta el formato MP3.
      </audio>
      <!-- boton de descarga del archivo mp3 -->
      <a href="1.mp3" download="1.mp3"><button>Descargar MP3</button></a>
    </li>
    <li>
      <!-- primer audio con reproduccion automatica OGG -->
      <audio controls>
        <source src="2.ogg" type="audio/ogg">
        Tu navegador no soporta el formato OGG.
      </audio>
      <!-- boton de descarga de archivo OGG -->
      <a href="2.ogg" download="2.ogg"><button>Descargar OGG</button></a>
    </li>
  </ul>
  <p>Nota: La reproducción automática del primer audio puede depender de la configuración del navegador.
```

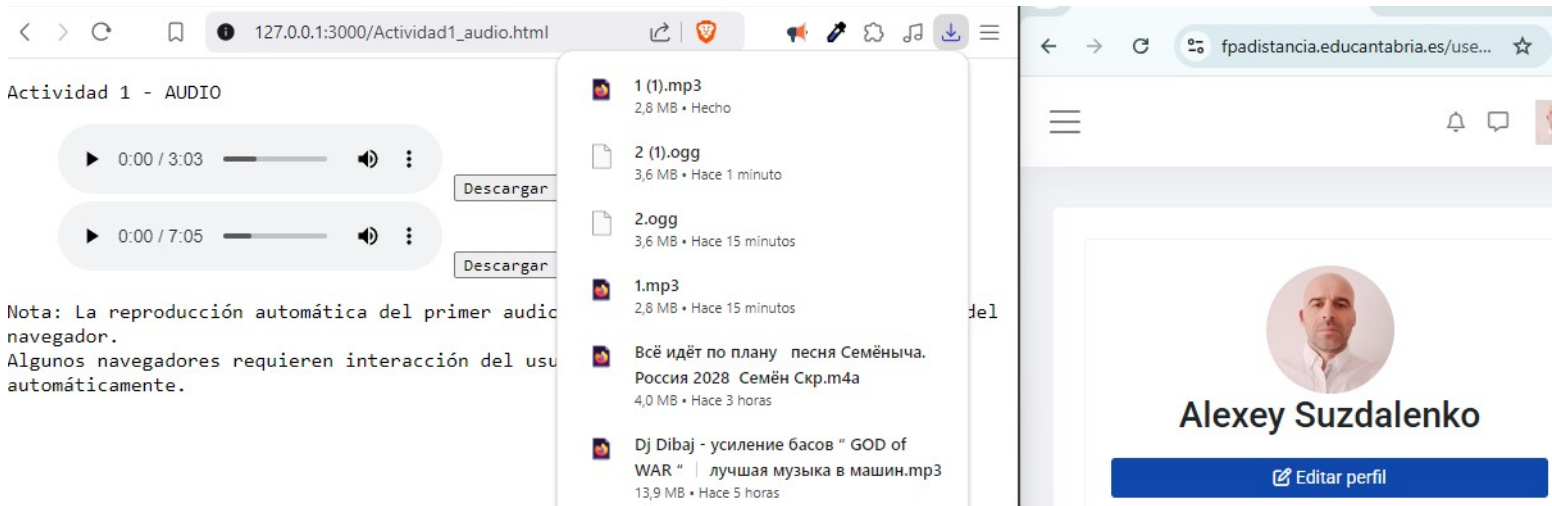
```

<br>
Algunos navegadores requieren interacción del usuario para reproducir audio automáticamente.
</p>
<style>
* {
    font-family: Consolas, "sans-serif";
}
button:hover {
    cursor: pointer;
}
li{
    list-style: none;
}
</style>
</body>
</html>

```

Los audios incluyen controles etiqueta “controls” y el primero se reproduce automáticamente etiqueta “autoplay”. Dependiendo de la configuración del navegador podemos tener bloqueada la reproducción automática.

Con el código `<button>Descargar MP3</button>` y `<button>Descargar OGG</button>` añado los botones de poder descargar el audio. Los archivos se descargan al presionar los botones, captura:



Como ya había comentado antes la aplicación está desplegada en un servidor de desarrollo en el URL http://127.0.0.1:3000/Actividad1_audio.html lo cual se puede ver en la captura.

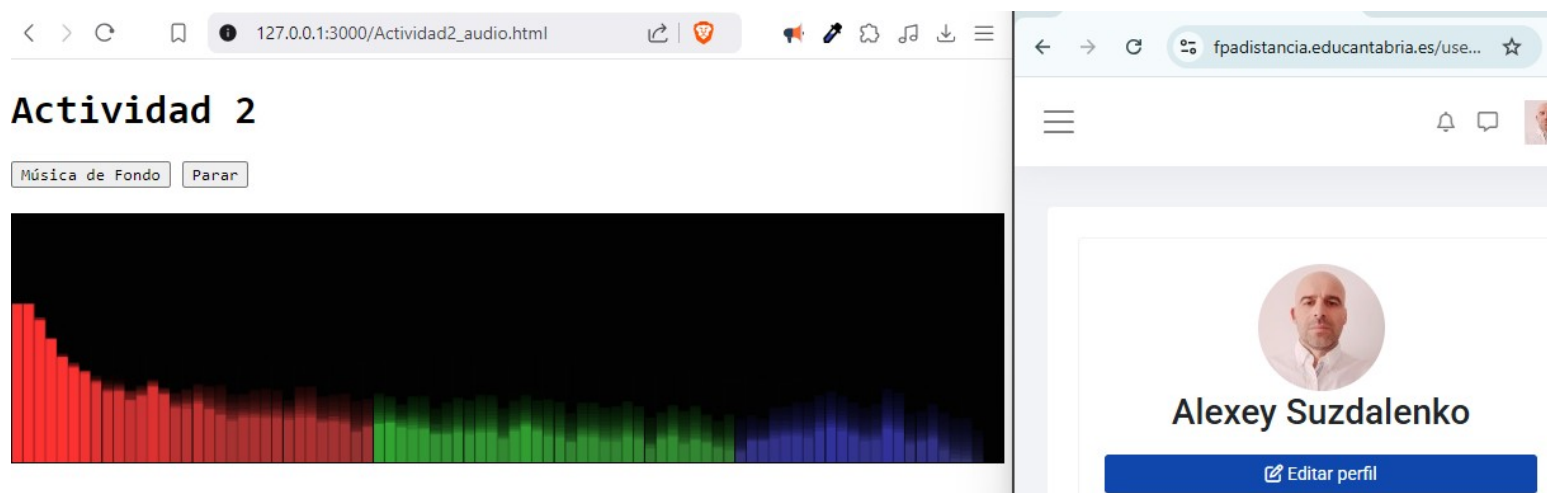
La etiqueta “download” es la que hace que el archivo se descargue.

Actividad 2 - AUDIO

He creado archivo html con en el botón “Música de Fondo” y al pulsarlo se reproduce la música en bucle desde la función “play()”; he añadido el segundo botón para parar la reproducción “Parar” con la función “pause()”.

Adicionalmente he añadido una vista de equalizador de frecuencia como una practica mas de JavaScript y audio..

Captura de ejecución del archivo html:



Código completo de esta actividad:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Actividad 2</title>
</head>
<body>
  <h1>Actividad 2</h1>
  <!-- Botón para reproducir música de fondo -->
  <button id="play-button">Música de Fondo</button>
  <!-- Botón para parar la música de fondo -->
  <button id="stop-button">Parar</button>
  <!-- Elemento de audio para la música de fondo -->
  <audio id="background-audio" src="1.mp3" type="audio/mpeg">
    Tu navegador no soporta el elemento de audio.
  </audio>
  <!-- Canvas para el visualizador de audio -->
  <canvas id="visualizer" width="800" height="200"></canvas>
  <script>
    let backgroundAudio, audioContext, analyser, dataArray, canvas, canvasContext;
    /* referencia al elemento <audio> que contiene el archivo de audio */
    backgroundAudio = document.getElementById("background-audio");
    canvas = document.getElementById("visualizer");
    canvasContext = canvas.getContext("2d");
    const playButton = document.getElementById("play-button");
    const stopButton = document.getElementById("stop-button");
    let isVisualizerInitialized = false;
    // Event listeners for play and stop buttons
    playButton.addEventListener("click", () => {
```

```

    play();
  });
  function play() {
    backgroundAudio.loop = true;
    backgroundAudio.play();
    if (!isVisualizerInitialized) {
      initializeVisualizer();
      drawVisualizer();
      isVisualizerInitialized = true;
    }
  }
  function pause(){
    backgroundAudio.pause();
    backgroundAudio.currentTime = 0; // Reinicia el audio al inicio
  }
  stopButton.addEventListener("click", () => {
    pause();
  });

  function initializeVisualizer() {
    /* representa el contexto de audio.
       Es una instancia de AudioContext, una API de Web Audio que permite trabajar con audio en tiempo
real,
       incluyendo análisis, reproducción y modificación. */
    audioContext = new (window.AudioContext || window.webkitAudioContext)();
    let source = audioContext.createMediaElementSource(backgroundAudio);
    /* es un nodo de análisis del contexto de audio.
       Permite acceder a datos sobre la frecuencia y el tiempo del audio que pasa por él
       crea un nodo de análisis. Este nodo extrae datos del audio que pasa por él.
    */
    analyser = audioContext.createAnalyser();
    source.connect(analyser);
    analyser.connect(audioContext.destination);
    analyser.fftSize = 256;
    let bufferLength = analyser.frequencyBinCount;
    /* un arreglo donde se almacenan las amplitudes de las frecuencias analizadas por el analyser. */
    dataArray = new Uint8Array(bufferLength);
  }
  function drawVisualizer() {
    requestAnimationFrame(drawVisualizer);
    analyser.getByteFrequencyData(dataArray);
    canvasContext.fillStyle = "rgba(0, 0, 0, 0.2)";
    canvasContext.fillRect(0, 0, canvas.width, canvas.height);
    const barWidth = (canvas.width / dataArray.length) * 1.3;
    let barHeight;
    let x = 0;

    for (let i = 0; i < dataArray.length; i++) {
      barHeight = dataArray[i];
      // Determinar el color según la posición de la barra
      if (i < dataArray.length / 4) {
        // Primera tercera parte: rojo
        canvasContext.fillStyle = "rgb(" + (barHeight + 100) + ",50,50)";
      } else if (i < (2 * dataArray.length) / 4) {
        // Segunda tercera parte: verde
        canvasContext.fillStyle = "rgb(50," + (barHeight + 100) + ",50)";
      } else {

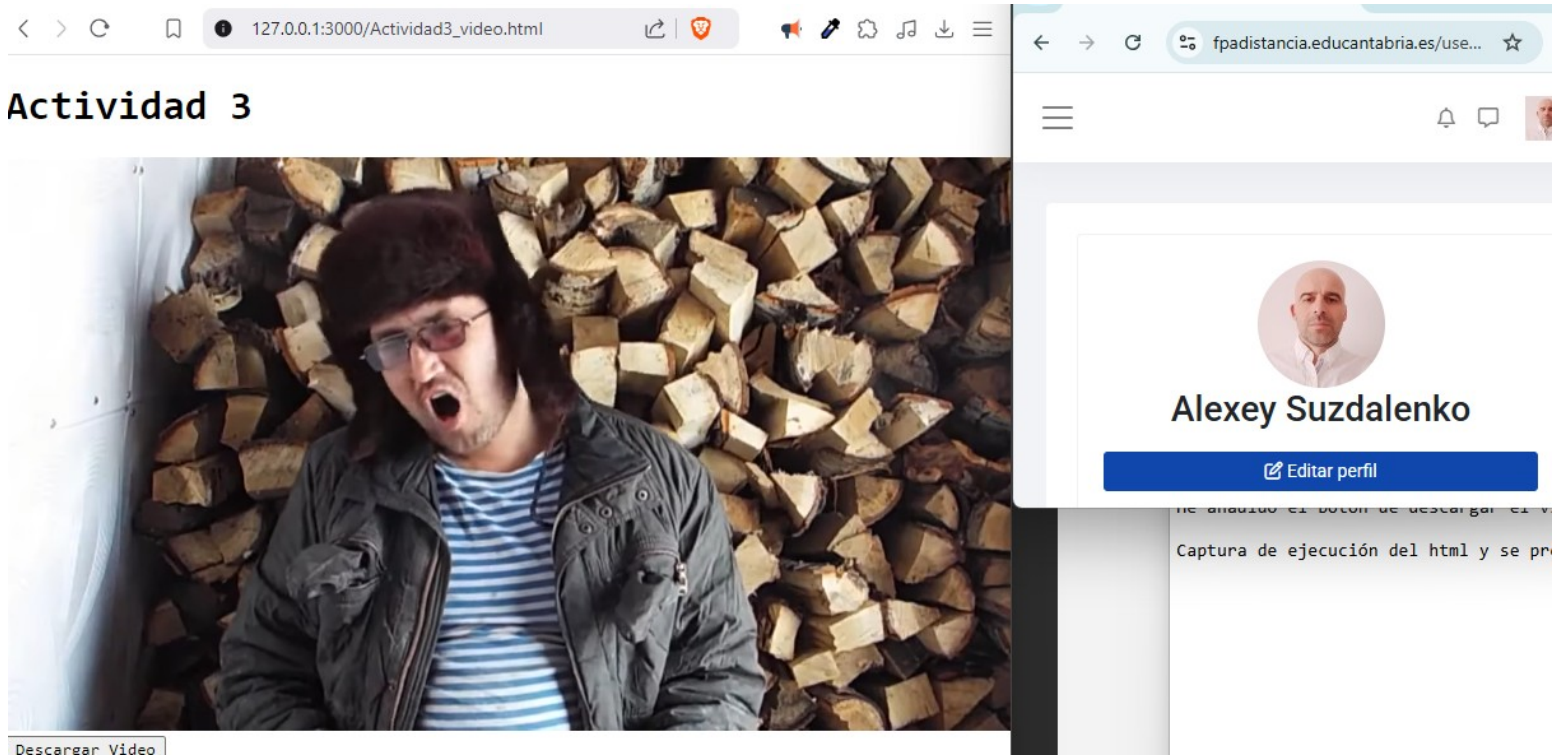
```

```
        // Última tercera parte: azul
        canvasContext.fillStyle = "rgb(50,50," + (barHeight + 100) + ")";
    }
    canvasContext.fillRect(x, canvas.height - barHeight / 2, barWidth, barHeight / 2);
    x += barWidth + 1;
}
}
</script>
<style>
    * {
        font-family: Consolas, "san-serif";
    }
    audio {
        display: none;
    }
    canvas {
        display: block;
        border: 1px solid black;
        display: block;
        width: 100%;
        margin: 20px auto;
    }
</style>
</body>
</html>
```

Actividad 3 - VIDEO

He creado el archivo html con el video que he encontrado en internet. El video se inicia al cargar la pagina, se muestran los controles, el video esta mateado y se reproduce en bucle “controls autoplay muted loop”; He añadido el botón de descargar el video.

Captura de ejecución del html y se presiono el botón de descargue el video se descarga.



Codigo completo de esta actividad:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Actividad 3 - Video</title>
</head>
<body>
  <h1>Actividad 3</h1>

  <!-- Video con reproducción automática, muteado, en bucle y controles -->
  <video id="sample-video" width="800" controls autoplay muted loop>
    <source src="https://suzdalenko.github.io/suzdalenko/1.mp4" type="video/mp4">
    Tu navegador no soporta el elemento de video.
  </video>

  <!-- Botón para descargar el video -->
  <br>
  <a href="https://suzdalenko.github.io/suzdalenko/1.mp4"
download="https://suzdalenko.github.io/suzdalenko/1.mp4">
    <button>Descargar Video</button>
  </a>
<style>
  * {
```



```
font-family: Consolas, "san-serif";
```

```
}
```

```
</style>
```

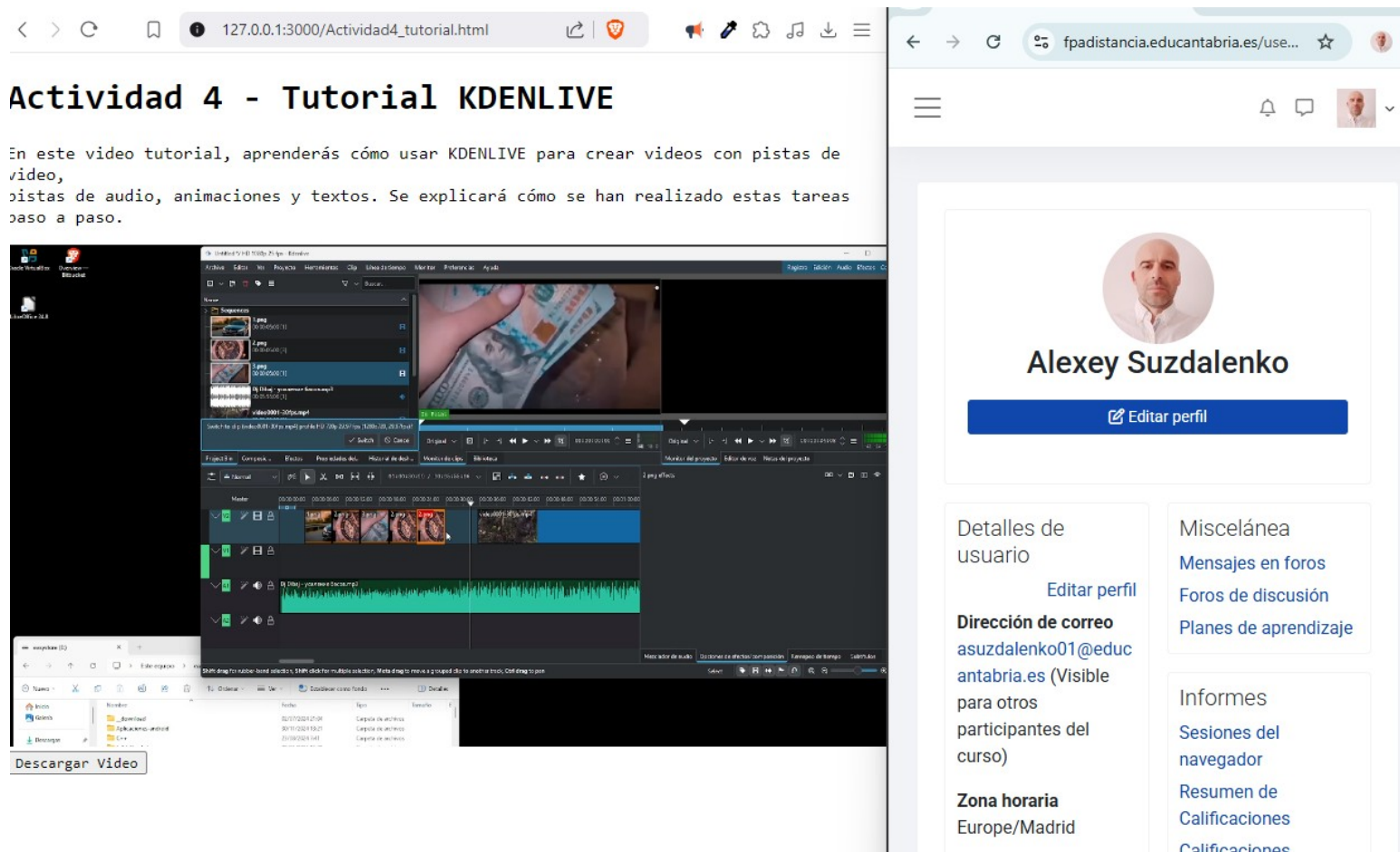
```
</body>
```

```
</html>
```

Actividad 4 - AUDIO Y VIDEO

He creado el archivo html en el que he incluido el video tutorial breve, he intentado que no sea mas de 3 minutos de utilización de KDENLIVE, lo primero tenemos que descargar y instalar este editor desde su pagina oficial. Después en el video insertado en el html intento explicar como a través de un video y una pista de audio y unos fotos se puede crear otro video gracias a la herramienta KDENLIVE. Al final del código html incluyo el botón para descargar el video tutorial.

Captura de ejecución del html:



Código html completo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Actividad 4 - Tutorial KDENLIVE</title>
</head>
<body>
  <h1>Actividad 4 - Tutorial KDENLIVE</h1>
  <p>En este video tutorial, aprenderás cómo usar KDENLIVE para crear videos con pistas de video, <br>
    pistas de audio, animaciones y textos. Se explicará cómo se han realizado estas tareas paso a paso.</p>
  <!-- Video del tutorial -->
  <video id="tutorial-video" width="800" controls muted>
    <source src="http://suzdalenko-dev.web.app/out.mp4" type="video/mp4">
    Tu navegador no soporta el elemento de video.
  </video>
  <!-- Botón para descargar el video -->
```

```
<br>
<a href="http://suzdalenko-dev.web.app/out.mp4" download="http://suzdalenko-dev.web.app/out.mp4">
  <button>Descargar Video</button>
</a>
<style>
  * {
    font-family: Consolas, "san-serif";
  }
</style>
</body>
</html>
```