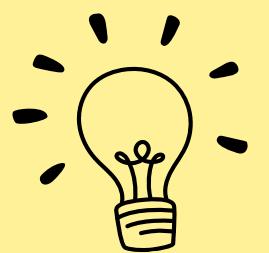




INTEGRACIÓN DE CONTENIDO MULTIMEDIA E INTERACTIVO EN UNA WEB DE EDUCACIÓN ONLINE



LIDIA GARCÍA MUÑOZ



FEB 2025

CONTENIDO

1. INTRODUCCIÓN

2. TECNOLOGÍAS UTILIZADAS

3. INSERCIÓN DE ELEMENTOS MULTIMEDIA

4. INTERACTIVIDAD CON EL USUARIO

5. PRUEBAS DE COMPATIBILIDAD Y
RESPONSIVIDAD

6. CONCLUSIÓN



I. INTRODUCCIÓN

Este proyecto consiste en una web educativa interactiva desarrollada con tecnologías web como HTML, CSS y JavaScript. El objetivo es proporcionar una plataforma dinámica y fácil de usar, que permita a los niños aprender de manera divertida y participativa.



La web incorpora elementos multimedia como videos, audios y animaciones, que enriquecen el contenido educativo. También se han incluido funciones como comentarios y cuestionarios interactivos, lo que fomenta la participación activa y el feedback en tiempo real. De este modo, buscamos crear un entorno donde los niños puedan explorar, aprender y comunicarse de forma dinámica y entretenida.

2. TECNOLOGÍAS UTILIZADAS



HTML

Lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la web a través de un conjunto de etiquetas y atributos.



JAVASCRIPT

Lenguaje de programación que añade interactividad, permitiendo que los usuarios interactúen dinámicamente con los elementos de la página.



CSS

Lenguaje utilizado para describir la apariencia y el diseño de una página web. Se utiliza junto con HTML para controlar cómo se ven los elementos en la pantalla.

3. INSERCIÓN DE ELEMENTOS MULTIMEDIA

Para hacer que la web sea más interactiva y atractiva para los niños, se han incorporado elementos multimedia como videos y audios.

Estos recursos permiten mejorar la experiencia de aprendizaje, facilitando la comprensión del contenido a través de estímulos visuales y sonoros.

En este proyecto, se ha utilizado HTML5 para la integración de estos elementos, junto con CSS y JavaScript para su correcto diseño y funcionalidad.

→ Pestaña "Videos": Se han insertado videos utilizando la etiqueta <video>, permitiendo a los usuarios reproducirlos directamente desde la web.

```
<video id="video-abc" width="80%" controls>
<source src="resources/abc.mp4" type="video/mp4">
Tu navegador no soporta la etiqueta de video.
</video>
```

→ Pestaña "Audio": Se ha agregado un audio con la etiqueta <audio>, brindando una opción adicional para que los niños puedan escuchar contenido educativo.

```
<audio id="audio-saludos" controls>
<source src="resources/saludos_ingles.mp3" type="audio/mp3">
Tu navegador no soporta la etiqueta de audio.
</audio>
```

4. INTERACTIVIDAD CON EL USUARIO

La interactividad permite que los usuarios participen activamente con los contenidos en lugar de ser solo espectadores. En esta plataforma educativa para niños, se han integrado elementos interactivos que mejoran la experiencia de aprendizaje.

En las siguientes diapositivas veremos en profundidad las siguientes funcionalidades que permiten la interacción del usuario:

1

Cuestionario al finalizar
la reproducción de los
vídeos.

2

Navegación
interactiva por
capítulos en el audio



3

Área de
comentarios en
tiempo real

Cuestionario al finalizar la reproducción de los vídeos

Al finalizar cada video, aparece automáticamente un cuestionario para evaluar la memoria visual del usuario. A modo ejemplo, veremos parte del código JavaScript, para ver cómo se implementa esta funcionalidad cuando finaliza el video "ABC":

```
// Mostrar el modal del cuestionario después de que termine el video de ABC
videoABC.addEventListener('ended', function() {
  videoABC.pause();
  modalABC.style.display = 'flex';
});

// Función para verificar la respuesta del cuestionario de ABC
function verificarRespuestaABC() {
  const respuestasABC = document.querySelectorAll('input[name="respuesta-abc"]:checked');

  if (respuestasABC.length === 0) {
    resultadoABC.innerHTML = 'Por favor, selecciona una respuesta.';
    return;
  }

  let respuestaSeleccionadaABC = respuestasABC[0].value;
  if (respuestaSeleccionadaABC === "3") {
    resultadoABC.innerHTML = '|Correcto! El autobús transportaba 3 lápices.';
  } else {
    resultadoABC.innerHTML = 'Respuesta incorrecta. Intenta nuevamente.';
  }
}
```

The screenshot shows a web browser window for 'Curso Interactivo para Niños'. At the top, there's a navigation bar with 'Home', 'Videos', and 'Audios'. Below it, a message says 'Al final de cada video deberás responder a una pregunta. ¡Ánimo!' (At the end of each video you will have to answer a question. Good luck!). A video player is visible, showing a cartoon about counting letters A, B, and C. To the right, a modal window titled 'Memoria Visual: ¿De qué color es...?' (Memory Visual: What color is...) asks '¿Cuántos lápices transportaba el autobús?' (How many pencils did the bus transport?). It lists three options: 3, 7, and 5. The option '3' is selected. Below the modal, a message says '|Correcto! El autobús transportaba 3 lápices.' (Correct! The bus transported 3 pencils.). At the bottom of the page, there's a copyright notice: '© 2025 Curso Interactivo para Niños'.

Navegación interactiva por capítulos en el audio

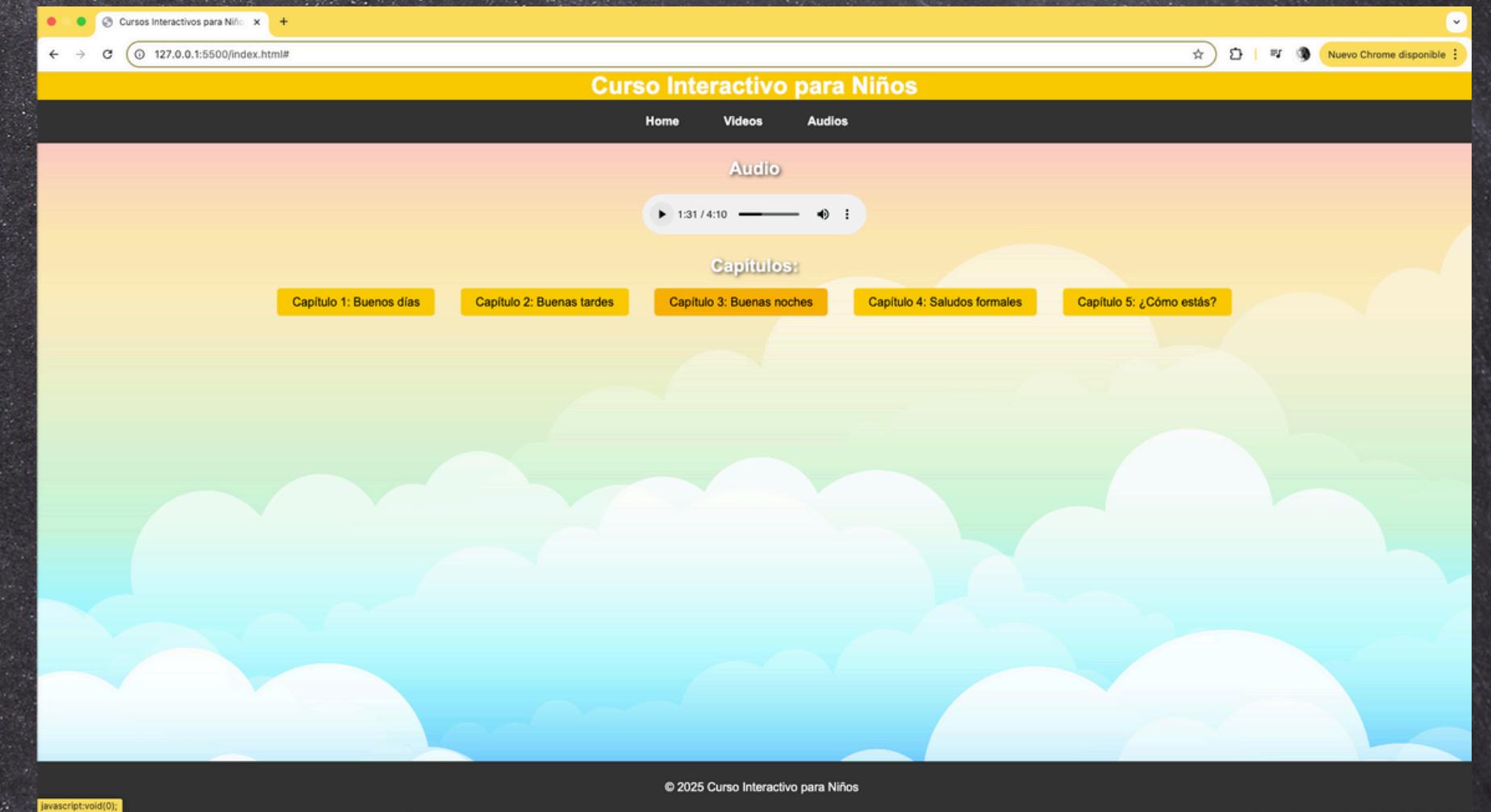
Los usuarios pueden saltar a secciones específicas del audio educativo seleccionando capítulos.

El siguiente código JavaScript permite cambiar el tiempo del audio al hacer clic en un capítulo:

```
// Función para ir a un capítulo específico del audio
function irACapítulo(segundos) {
  var audio = document.getElementById('audio-saludos'); // Obtén el reproductor de audio
  audio.currentTime = segundos; // Establece el tiempo en segundos
  audio.play(); // Reproduce el audio desde ese punto
}
```

Cada enlace de capítulo llama a esta función con el tiempo específico. Esta parte de código la implementamos en el documento HTML con la correspondiente etiqueta anchor de enlace dentro de una lista no ordenada.

```
<ul class="lista-capitulos">
<li><a href="javascript:void(0); onclick="irACapítulo(35)">Capítulo 1: Buenos días </a></li>
<li><a href="javascript:void(0); onclick="irACapítulo(65)">Capítulo 2: Buenas tardes </a></li>
<li><a href="javascript:void(0); onclick="irACapítulo(90)">Capítulo 3: Buenas noches </a></li>
<li><a href="javascript:void(0); onclick="irACapítulo(124)">Capítulo 4: Saludos formales</a></li>
<li><a href="javascript:void(0); onclick="irACapítulo(150)">Capítulo 5: ¿Cómo estás?</a></li>
</ul>
```



**en este ejemplo vemos como al hacer clic sobre "Capítulo 3: Buenas noches" nos lleva directamente al segundo 90 (l'31")

Agregar comentarios en tiempo real

Los usuarios pueden dejar comentarios en la pestaña principal Home. Estos se almacenan en el navegador con localStorage, permitiendo que se mantengan aunque se recargue la página. Además, tendrán la opción de borrar el comentario.

Veamos cómo implementar el área de comentarios con código JavaScript:

```
// Función para cargar los comentarios almacenados en el localStorage
function cargarComentarios() {
    let comentariosLista = document.getElementById('comentarios-lista');
    comentariosLista.innerHTML = ""; // Limpiar la lista antes de cargar

    let comentariosGuardados = JSON.parse(localStorage.getItem('comentarios')) || [];
    comentariosGuardados.forEach(function(comentario) {
        let comentarioDiv = document.createElement('div');
        comentarioDiv.classList.add('comentario');

        let comentarioContenido = document.createElement('p');
        comentarioContenido.textContent = comentario.texto;

        let botonEliminar = document.createElement('button');
        botonEliminar.textContent = 'Eliminar';
        botonEliminar.classList.add('eliminar');
        botonEliminar.onclick = function() {
            eliminarComentario(comentarioDiv, comentario.texto);
        };

        comentarioDiv.appendChild(comentarioContenido);
        comentarioDiv.appendChild(botonEliminar);
        comentariosLista.appendChild(comentarioDiv);
    });
}
```

```
// Función para agregar un comentario
function agregarComentario() {
    let comentarioInput = document.getElementById('comentario-input');
    let comentarioTexto = comentarioInput.value.trim();

    if (comentarioTexto === "") {
        alert("Por favor, escribe un comentario.");
        return;
    }

    let comentariosLista = document.getElementById('comentarios-lista');

    // Crear el div del comentario
    let comentarioDiv = document.createElement('div');
    comentarioDiv.classList.add('comentario');

    // Crear el contenido del comentario
    let comentarioContenido = document.createElement('p');
    comentarioContenido.textContent = comentarioTexto;

    // Crear el botón de eliminar
    let botonEliminar = document.createElement('button');
    botonEliminar.textContent = 'Eliminar';
    botonEliminar.classList.add('eliminar');
    botonEliminar.onclick = function() {
        eliminarComentario(comentarioDiv, comentarioTexto);
    };

    comentarioDiv.appendChild(comentarioContenido);
    comentarioDiv.appendChild(botonEliminar);
    comentariosLista.appendChild(comentarioDiv);
}
```

```
// Agregar el contenido y el botón de eliminar al div del comentario
comentarioDiv.appendChild(comentarioContenido);
comentarioDiv.appendChild(botonEliminar);

// Añadir el comentario a la lista de comentarios
comentariosLista.appendChild(comentarioDiv);

// Guardar el comentario en el localStorage
let comentariosGuardados = JSON.parse(localStorage.getItem('comentarios')) || [];
comentariosGuardados.push({ texto: comentarioTexto });
localStorage.setItem('comentarios', JSON.stringify(comentariosGuardados));

// Limpiar el campo de texto
comentarioInput.value = "";

// Función para eliminar un comentario
function eliminarComentario(comentarioDiv, textoComentario) {
    let comentariosLista = document.getElementById('comentarios-lista');

    // Eliminar el comentario de la interfaz
    comentariosLista.removeChild(comentarioDiv);

    // Eliminar el comentario del localStorage
    let comentariosGuardados = JSON.parse(localStorage.getItem('comentarios')) || [];
    comentariosGuardados = comentariosGuardados.filter(function(comentario) {
        return comentario.texto !== textoComentario;
    });
    localStorage.setItem('comentarios', JSON.stringify(comentariosGuardados));
}

// Cargar los comentarios al cargar la página
window.onload = cargarComentarios;
```

Agregar comentarios en tiempo real

This screenshot shows a web browser window for the 'Curso Interactivo para Niños' website. The page features a yellow header with the title 'Curso Interactivo para Niños' and navigation links for 'Home', 'Videos', and 'Audios'. Below the header is a colorful background with clouds and a photo of children walking. A comment input field is visible with the placeholder 'Deja tu comentario:' and a message 'Hola, estoy encantad@ con esta web didáctica 😊😊😊'. A yellow 'Añadir Comentario' button is located below the input field. The footer contains the copyright notice '© 2025 Curso Interactivo para Niños'. A large pink arrow points from the left side of the screen towards the right, indicating the flow of the process.

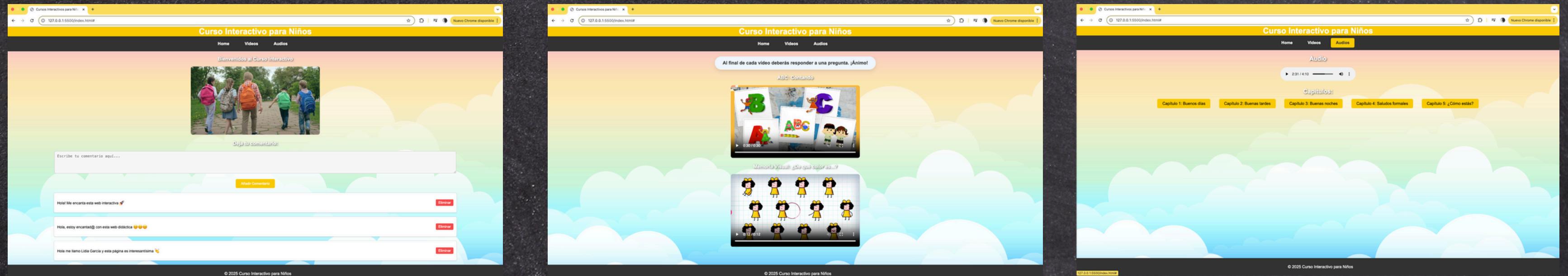
This screenshot shows the same 'Curso Interactivo para Niños' website after the comment has been posted. The comment 'Hola! estoy encantad@ con esta web didáctica 😊😊😊' is now listed in the comment section along with another existing comment 'Hola! Me encanta esta web interactiva 🎉'. Each comment has a small red 'Eliminar' (Delete) button next to it. The rest of the page layout remains the same, including the header, background image, and footer.

5. PRUEBAS DE COMPATIBILIDAD Y RESPONSIVIDAD

Para asegurarnos de que la web se vea y funcione correctamente en diferentes dispositivos, hemos realizado pruebas de compatibilidad y responsividad usando la herramienta Chrome Dev Tools.

Adaptar la web a distintos tamaños de pantalla es clave para mejorar la experiencia del usuario, permitiendo que se visualice bien tanto en computadoras como en tablets y móviles.

A continuación, podemos observar el resultado de la UI en un monitor 1080 píxeles de alto:

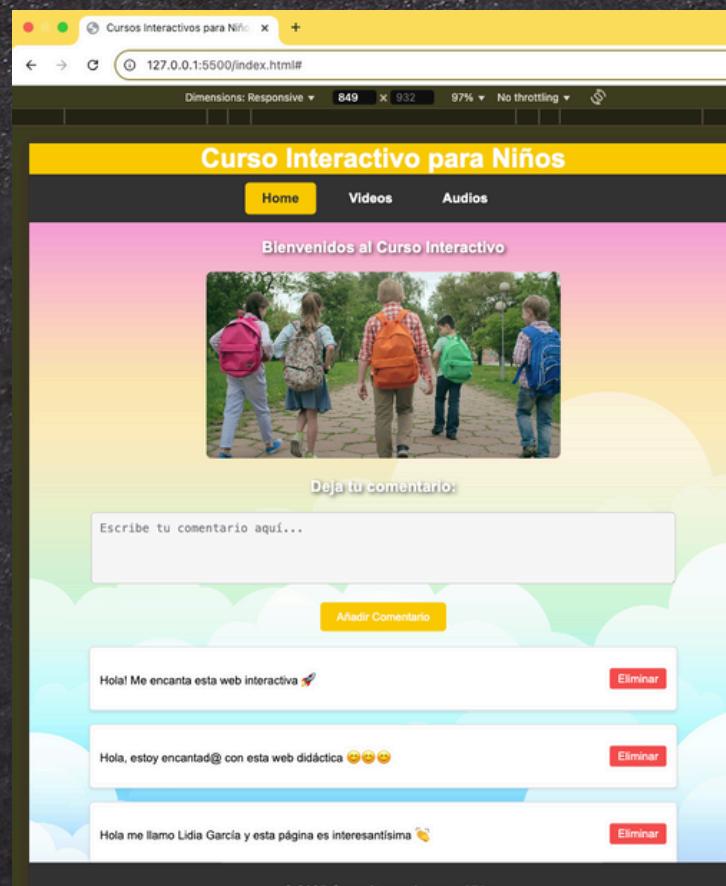


En las siguientes diapositivas veremos cómo se adapta nuestra aplicación web a los distintos dispositivos mediante el uso de la herramienta Dev Tools de Chrome.

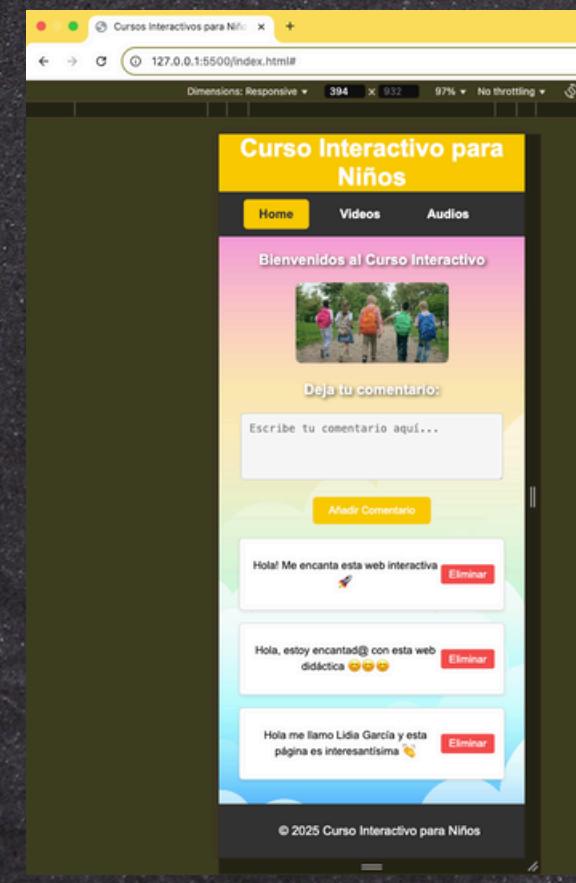
5. PRUEBAS DE COMPATIBILIDAD Y RESPONSIVIDAD

Para comprobar cómo se adapta la web a distintos dispositivos, hemos utilizado el Modo Responsive de Chrome DevTools. Esto nos ha permitido simular la visualización en móviles, tablets y pantallas personalizadas, ajustando estilos y corrigiendo posibles errores.

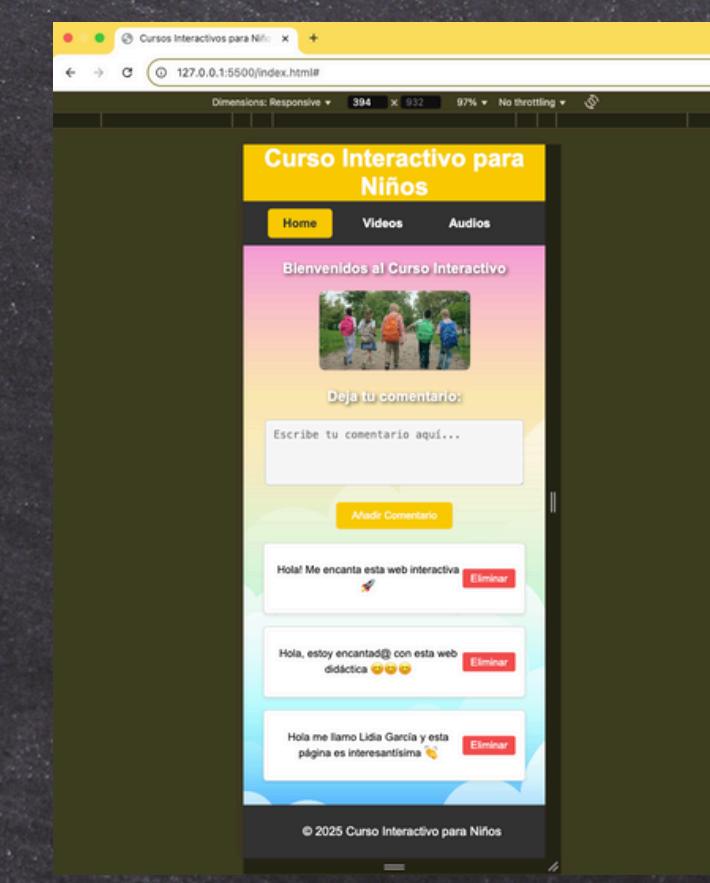
A continuación, se muestran capturas de la web en diferentes tamaños de pantalla:



849x932



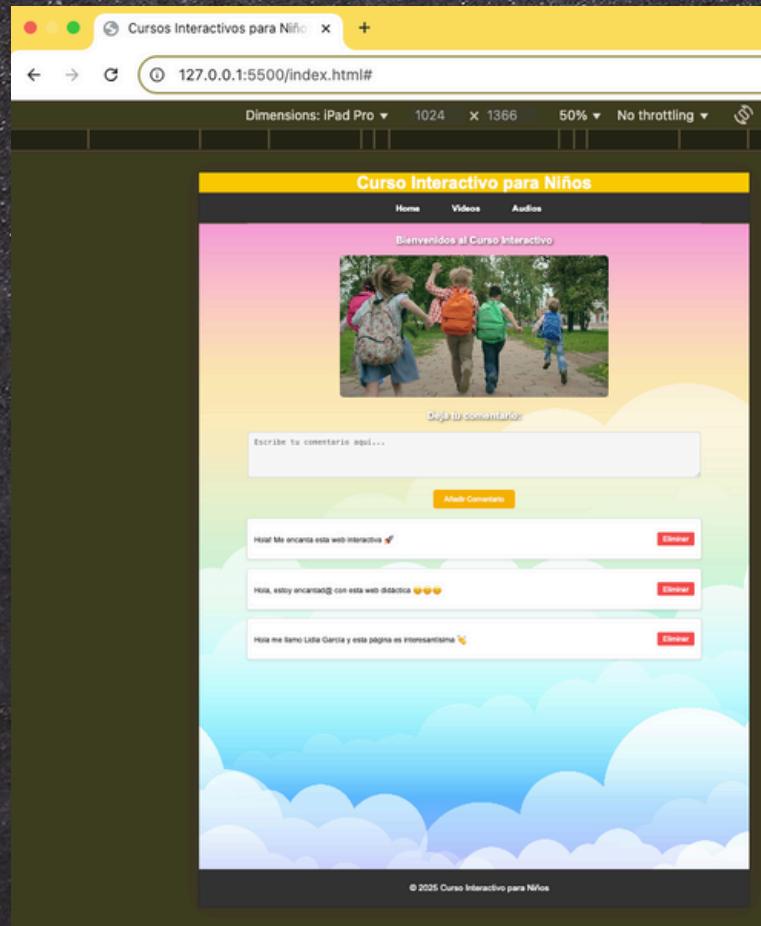
394x932



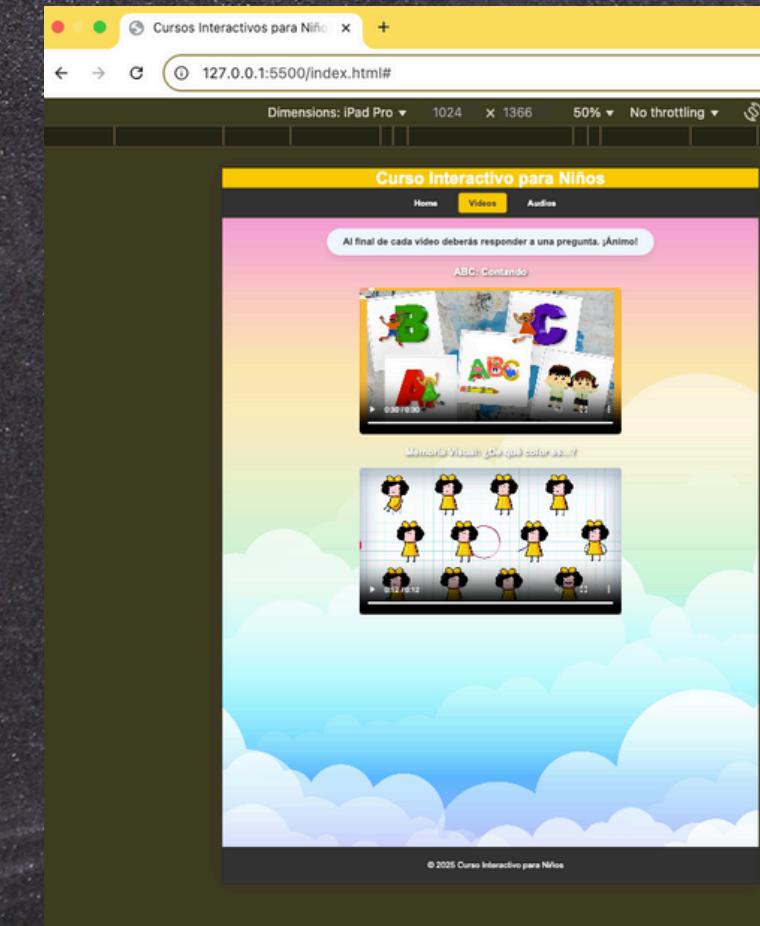
504x932

5. PRUEBAS DE COMPATIBILIDAD Y RESPONSIVIDAD

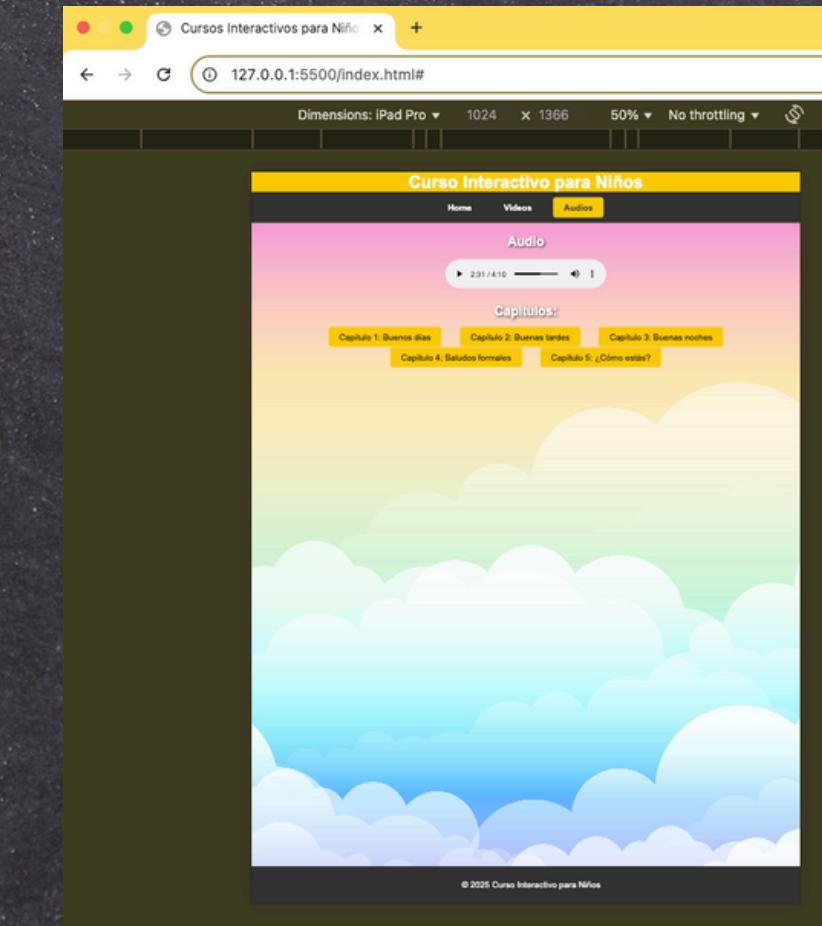
Modo iPad: para verificar la adaptación en tablets:



Home



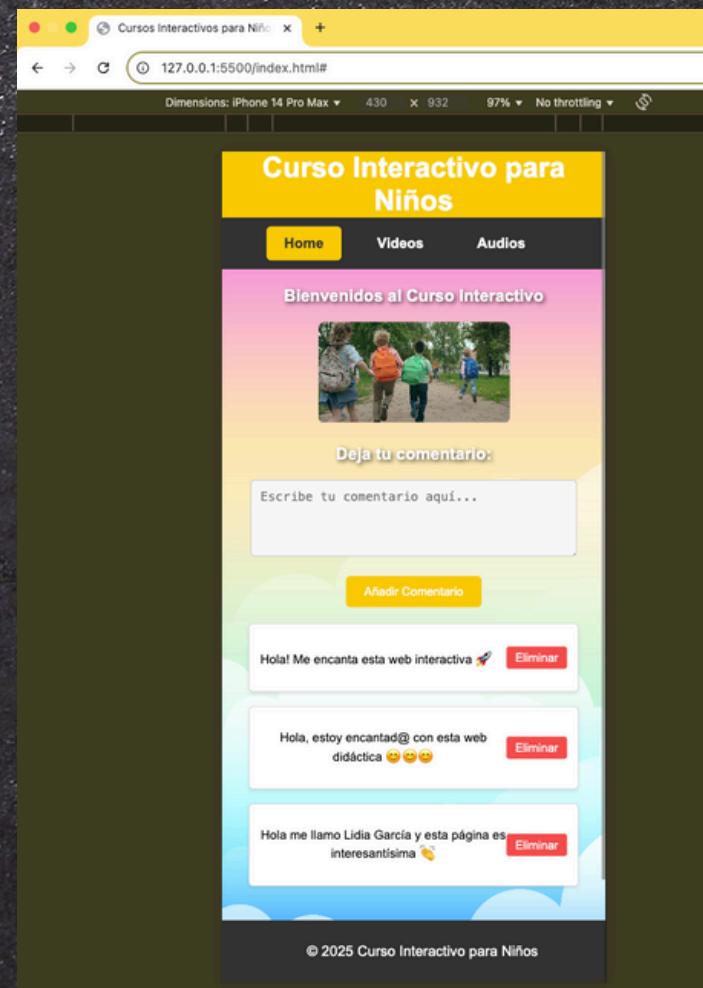
Videos



Audios

5. PRUEBAS DE COMPATIBILIDAD Y RESPONSIVIDAD

Modo iPhone: para evaluar la experiencia en dispositivos móviles.



Home



Videos



Audios

6. CONCLUSIÓN

En resumen, este proyecto ha sido desarrollado con HTML, CSS y JavaScript para crear una web educativa interactiva. A lo largo de la presentación, hemos visto cómo la combinación de estos elementos multimedia permite mejorar la experiencia de los usuarios, haciendo que se involucren de forma más dinámica y entretenida.

La interactividad es clave en este tipo de plataformas, ya que facilita la participación y el aprendizaje activo a través de videos, audios, comentarios y actividades interactivas.

FUENTES

→ Temario Cesur . Diseño de Interfaces Web - UD4 "Integración de contenido interactivo".

→ Presentación Lidia García Muñoz - "Creación de una página web interactiva: HTML, CSS. y JavaScript en acción"

RECURSOS

✖ IDE Visual Studio Code

Chrome Navegador Chrome

¡GRACIAS!



LIDIA GARCIA MUÑOZ

FEB 2025