

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLOGICO DE OAXACA

INTEGRACION DE PROCESOS DE DESARROLLO DE SOFTWARE

GRUPO: 9SA

HORARIO: 07:00-08:00

DOCENTE: Espinosa Pérez Jacob

ALUMNOS:

Méndez Mendoza Luisa Michel

Pérez de Jesús Edith

Martínez García Yahir Omar

López García Lourdes Gloria

PRESENTA:

CUARTO REPORTE

INDICE

Bitácora	3
Optimization endpoint de búsqueda	5
Implementación de historial de visualización	5
Control de edad en reproducción	6
Exploración de videos por ubicación e historial	6
Vista “Historial” y “Exploración”	7
Refinamiento de UX/UI y refinamiento en vista de reproducción	8
Integración de backend y frontend (historial y exploración)	9
Pruebas funcionales	10

BITACORA

ITERACION 3 REPRODUCCION, HISTORIAL Y EXPLORACION DE VIDEOS							
Actividad	Descripción	Responsable	Estatus	Resultado	Fecha inicio	Fecha fin	Observaciones
Optimization endpoint de búsqueda	Se mejoró el endpoint /api/videos/search para aceptar parámetros searchTerm, location y maxResults, optimizando el filtrado de resultados relevantes.	Michel	✓ Finalizada	Endpoint optimizado y probado con distintos parámetros desde el frontend.	06/10/25	07/10/25	Según la búsqueda responde correctamente según el término y ubicación.
Implementación de historial de visualización	Se añadió el registro automático de videos vistos mediante el endpoint /api/users/history, almacenando título, miniatura, canal y fecha.	Yahir	✓ Finalizada	Los videos reproducidos se guardan en MongoDB y se muestran en la vista Library.	06/10/25	07/10/25	El historial se actualiza cada vez que se abre un video
Control de edad en reproducción	Se implementó el componente AgePrompt.js que solicita verificación de edad antes de abrir videos marcados como +18.	Michel	✓ Finalizada	Función de validación de edad funcionando correctamente.	06/10/25	07/10/25	Se muestra la alerta si el usuario no cumple con la edad requerida
Exploración de videos por ubicación e historial	Se desarrolló la vista ExplorePage.js que muestra videos según la ubicación del usuario usando	Lourdes	✓ Finalizada	Mapa y videos geolocalizados integrados en el frontend.	06/10/25	07/10/25	Funciona correctamente incluso si el usuario deniega permisos

	navigator.geolocation y /api/location/geocode.						
Vista "Historial"	Se implementó la sección de videos sugeridos según historial y ubicación actual del usuario.	Edith	<input checked="" type="checkbox"/> Finalizada	ExplorePage muestra resultados basados en preferencias previas.	06/10/25	07/10/25	Los usuarios pueden visualizar la vista, historial y explorar
Refinamiento de UX/UI	Se mejoró el diseño de las tarjetas de video en Home y Explore con un formato responsive y distintivo +18.	Yahir	<input checked="" type="checkbox"/> Finalizada	. Interfaz refinada, fluida y compatible con móviles.	06/10/25	07/10/25	No se detectaron errores visuales tras las pruebas en distintos dispositivos
Refinamiento en vista de reproducción							
Integración de backend y frontend (historial y exploración)	Se integraron las funciones de búsqueda, historial y geolocalización con el frontend mediante Axios.	Michel	<input checked="" type="checkbox"/> Finalizada	Flujo completo de búsqueda → reproducción → historial funcional.	06/10/25	07/10/25	Todo el flujo fue probado con diferentes usuarios
Pruebas funcionales	Se validó el flujo completo entre módulos: búsqueda, reproducción, historial y exploración.	Todos	<input checked="" type="checkbox"/> Finalizada	Todos los módulos operan correctamente de manera integrada.	06/10/25	07/10/25	Pruebas exitosas
Observaciones		Todos	<input checked="" type="checkbox"/> Finalizada		06/10/25	07/10/25	Observaciones 06/10/25 Al momento de reproducir un video no se debe de salir de la aplicación, se debe reproducir en la aplicación

OPTIMIZATION ENDPOINT DE BUSQUEDA

```
videoRoutes.js ×
ck-nodejs > src > routes > videoRoutes.js > ...
1 // backend/src/routes/videoRoutes.js
2 const express = require('express');
3 const { searchYouTubeVideos } = require('../utils/youtube');
4
5 const router = express.Router();
6
7 router.get('/search', async (req, res) => {
8   // ... CAMBIO AQUI ...
9   // Recibimos searchTerm y location por separado
10  const { searchTerm, location, maxResults } = req.query;
11
12  if (!searchTerm) {
13    return res.status(400).json({ message: 'Se requiere un término de búsqueda.' });
14  }
15
16  try {
17    const videos = await searchYouTubeVideos(searchTerm, location, maxResults ? parseInt(maxResults) : 10);
18    res.json(videos);
19  } catch (error) {
20    console.error(`Error en el endpoint /videos/search`, error.message);
21    res.status(500).json({ message: 'Error del servidor al buscar videos.' });
22  }
23});
24
25
26 module.exports = router;
```

Endpoint que recibe parametros de busqueda (searchTerm, location, maxResults) y devuelve resultados optimizados desde YouTube.

IMPLEMENTACIÓN DE HISTORIAL DE VISUALIZACIÓN

```
userController.js × JS history.js
ck-nodejs > src > controllers > userController.js > ...
1 // backend/src/controllers/userController.js
2 const User = require('../models/user');
3
4 // @desc Añadir un video al historial del usuario
5 // @route POST /api/users/history
6 // @access Private
7 exports.addVideoToHistory = async (req, res) => {
8   const { videoId, title, thumbnail, channelTitle } = req.body;
9
10  if (!videoId || !title || !thumbnail || !channelTitle) {
11    return res.status(400).json({ message: 'Faltan datos del video para añadir al historial.' });
12  }
13
14  try {
15    const user = req.user; // req.user viene del middleware de autenticación
16
17    // Verificar si el video ya está en el historial para evitar duplicados recientes
18    const existingEntryIndex = user.history.findIndex(
19      (item) => item.videoId === videoId
20    );
21
22    if (existingEntryIndex !== -1) {
23      // Si el video ya está, lo quitamos para moverlo al principio (más reciente)
24      user.history.splice(existingEntryIndex, 1);
25    }
26
27    // Añadir el nuevo video al principio del historial
28    user.history.unshift({
29      videoId,
30      title,
31      thumbnail,
32      channelTitle,
33      watchedAt: new Date(),
34    });
35
36    // Limitar el historial a un número razonable de videos (ej. los últimos 50)
37    if (user.history.length > 50) {
38
```

```
userController.js × JS history.js
ck-nodejs > src > utils > history.js > ...
1 // frontend/src/utils/history.js
2 import axios from 'axios';
3
4 const API_BASE_URL = 'http://localhost:5000/api';
5
6 export const addVideoToUserHistory = async (video) => {
7   const token = localStorage.getItem('token');
8   if (!token) {
9     console.warn('No hay token de autenticación. No se puede añadir video al historial.');
10    return;
11  }
12
13  try {
14    await axios.post(`${API_BASE_URL}/users/history`,
15      {
16        videoId: video.id,
17        title: video.title,
18        thumbnail: video.thumbnail,
19        channelTitle: video.channelTitle
20      },
21      {
22        headers: { Authorization: `Bearer ${token}` }
23      }
24    );
25  } catch (error) {
26    console.error(`Error al añadir video al historial:`, error.response ? error.response.data : error.message);
27  }
28}
29
30 export const getHistory = async () => {
31   const token = localStorage.getItem('token');
32   if (!token) return [];
33   try {
34     const res = await axios.get(`${API_BASE_URL}/users/history`,
35      {
36        headers: { Authorization: `Bearer ${token}` }
37      }
38    );
39  } catch (error) {
40    console.error(`Error al obtener el historial:`, error.message);
41  }
42}
```

Como se reduce un video, se guarda automáticamente en MongoDB y se muestra en la vista LibraryPage

CONTROL DE EDAD EN REPRODUCCIÓN

```
AgePrompt.js X
onEdit > src > components > JS AgePrompt.js > ...
1 // frontend/src/components/AgePrompt.js
2 import React, { useState } from 'react';
3 import './styles/AgePrompt.css'; // (opcional) estilos
4
5 const AgePrompt = ({ isOpen, onConfirm, onCancel, videoTitle }) => {
6   const [birthYear, setBirthYear] = useState('');
7
8   if (!isOpen) return null;
9
10  const handleConfirm = () => {
11    const year = parseInt(birthYear, 10);
12    if (!year || year < 1900 || year > new Date().getFullYear()) {
13      alert('Introduce un año de nacimiento válido.');
14      return;
15    }
16    const age = new Date().getFullYear() - year;
17    onConfirm(age >= 18);
18  };
19
20  return (
21    <div className="age-overlay">
22      <div className="age-panel">
23        <h3>Verificación de edad:</h3>
24        <p>El video "{videoTitle}" está marcado como no apto para menores.</p>
25        <p>Por favor, ingresa tu año de nacimiento para continuar.</p>
26        <input
27          type="number"
28          placeholder="Ej. 1995"
29          value={birthYear}
30          onChange={(e) => setBirthYear(e.target.value)}
31        />
32        <div className="age-actions">
33          <button className="age-btn" onClick={handleConfirm}>Confirmar</button>
34          <button className="age-btn cancel" onClick={onCancel}>Cancelar</button>
35        </div>
36      </div>
37    </div>
38  );
39
```

Ventana modal que solicita el año de nacimiento antes de reproducir videos restringidos (+18). Lo utilizamos en HomePage y ExplorePage

```
AgePrompt.js | JS ExplorePage.js X
onEdit > src > pages > JS ExplorePage.js > ...
1 // frontend/src/pages/ExplorePage.js
2 import React, { useState, useEffect, useCallback } from 'react';
3 import axios from 'axios';
4 import MapComponent from '../components/MapComponent';
5 import './styles/ExplorePage.css';
6 import { addVideoToUserHistory } from '../utils/history';
7 import LocationPermissionPanel from '../components/LocationPermissionPanel';
8 import AgePrompt from '../components/AgePrompt';
9 import VideoPlayer from '../components/VideoPlayer';
10
11 const API_URL = 'http://localhost:5000/api';
12
13 const ExplorePage = () => {
14   const [videos, setVideos] = useState([]);
15   const [loading, setLoading] = useState(true);
16   const [error, setError] = useState('');
17   const [locationName, setLocationName] = useState('');
18   const [mapCenter, setMapCenter] = useState(null);
19   const [showPermissionPanel, setShowPermissionPanel] = useState(null);
20   const [selectedVideoId, setSelectedVideoId] = useState(null);
21   const [agePromptOpen, setAgePromptOpen] = useState(false);
22   const [pendingVideoToOpen, setPendingVideoToOpen] = useState(null);
23
24   const fetchVideosForLocation = useCallback(async (lat, lng) => {
25     setLoading(true);
26     setError('');
27     try {
28       const geoRes = await axios.get(`${API_URL}/location/geocode`, { params: { lat, lon: lng } });
29       const fetchedLocationName = geoRes.data.locationName;
30       setLocationName(fetchedLocationName);
31
32       const videoRes = await axios.get(`${API_URL}/videos/search`, {
33         params: { searchTerm: 'videos', location: fetchedLocationName, maxResults: 15 },
34       });
35       setVideos(videoRes.data || []);
36     } catch (err) {
37       console.error('Error al cargar videos:', err);
38     }
39   }, []);
40
```

EXPLORACIÓN DE VIDEOS POR UBICACIÓN E HISTORIAL

Usamos navigator.geolocation y el endpoint /api/location/geocode para mostrar videos según la ubicación del usuario en un mapa

VISTA “HISTORIAL” Y “EXPLORAR”

```
JS AgePrompt.js | JS ExplorePage.js ×
frontend > src > pages > JS ExplorePage.js > [x] ExplorePage
1 // frontend/src/pages/ExplorePage.js
2 import React, { useState, useEffect, useCallback } from 'react';
3 import axios from 'axios';
4 import MapComponent from '../components/MapComponent';
5 import '../styles/ExplorePage.css';
6 import { addVideoToUserHistory } from '../utils/history';
7 import LocationPermissionPanel from '../components/LocationPermissionPanel';
8 import AgePrompt from '../components/AgePrompt';
9 import VideoPlayer from '../components/VideoPlayer';
10
11 const API_URL = 'http://localhost:5000/api';
12
13 const ExplorePage = () => [
14   const [videos, setVideos] = useState([]);
15   const [loading, setLoading] = useState(true);
16   const [error, setError] = useState('');
17   const [locationName, setLocationName] = useState('');
18   const [mapCenter, setMapCenter] = useState(null);
19   const [showPermissionPanel, setShowPermissionPanel] = useState(null);
20   const [selectedVideoId, setSelectedVideoId] = useState(null);
21   const [agePromptOpen, setAgePromptOpen] = useState(false);
22   const [pendingVideoToOpen, setPendingVideoToOpen] = useState(null);
23
24   const fetchVideosForLocation = useCallback(async (lat, lng) => {
25     setLoading(true);
26     setError('');
27     try {
28       const geoRes = await axios.get(`${API_URL}/location/geocode`, { params: { lat, lon: lng } });
29       const fetchedLocationName = geoRes.data.locationName;
30       setLocationName(fetchedLocationName);
31
32       const videoRes = await axios.get(`${API_URL}/videos/search`, {
33         params: { searchTerm: 'videos', location: fetchedLocationName, maxResults: 15 },
34       });
35       setVideos(videoRes.data || []);
36     } catch (err) {
37       console.error('Error al cargar videos:', err);
    
```

```
JS LibraryPage.js ×
frontend > src > pages > JS LibraryPage.js > [x] LibraryPage > ↗ useEffect() callback > [x] fetchHistory
1 // frontend/src/pages/LibraryPage.js
2
3 import React, { useState, useEffect } from 'react';
4 import axios from 'axios';
5 import { addVideoToUserHistory } from '../utils/history';
6 import '../styles/LibraryPage.css';
7 import VideoPlayer from '../components/VideoPlayer';
8
9 const API_URL = 'http://localhost:5000/api';
10
11 const LibraryPage = () => {
12   const [history, setHistory] = useState([]);
13   const [loading, setLoading] = useState(true);
14   const [error, setError] = useState('');
15   const [selectedVideoId, setSelectedVideoId] = useState(null);
16
17   useEffect(() => {
18     const fetchHistory = async () => {
19       setLoading(true);
20       setError('');
21       const token = localStorage.getItem('token');
22       if (!token) {
23         setError('Debes iniciar sesión para ver tu historial.');
24         setLoading(false);
25         return;
26       }
27
28       try {
29         const res = await axios.get(`${API_URL}/users/history`, {
30           headers: {
31             Authorization: `Bearer ${token}`,
32           },
33         });
34         setHistory(res.data);
35       } catch (err) {
36         console.error(
            'Error al cargar el historial:',
```

Se muestra la vista historial y explorar, calcula automáticamente palabras clave frecuentes del historial para generar sugerencias personalizadas en la vista explorador.

REFINAMIENTO DE UX/UI Y REFINAMIENTO EN VISTA DE REPRODUCCIÓN

```

JS ExplorePage.js JS HomePage.js X JS LibraryPage.js
frontend > src > pages > JS HomePage.js > [o] HomePage
1 // frontend/src/pages/HomePage.js
2 import React, { useState, useEffect } from 'react';
3 import axios from 'axios';
4 import { FaSearch, FaMapMarkedAlt } from 'react-icons/fa';
5 import '../styles/HomePage.css';
6 import { addVideoToUserHistory } from '../utils/history';
7 import AgePrompt from '../components/AgePrompt';
8 import LocationPermissionPanel from '../components/LocationPermissionPanel';
9
10 import VideoPlayer from '../components/VideoPlayer';
11
12 const API_URL = 'http://localhost:5000/api';
13
14 const HomePage = ({ user }) => [
15   const [videos, setVideos] = useState([]);
16   const [loading, setLoading] = useState(false);
17   const [error, setError] = useState('');
18   const [locationName, setLocationName] = useState('');
19   const [searchQuery, setSearchQuery] = useState('');
20   const [agePromptOpen, setAgePromptOpen] = useState(false);
21   const [pendingVideoToOpen, setPendingVideoToOpen] = useState(null);
22   const [showPermissionPanel, setShowPermissionPanel] = useState(true);
23
24   const [selectedVideoId, setSelectedVideoId] = useState(null);
25
26
27   useEffect(() => {
28     const savedLocation = localStorage.getItem('userLocation');
29     if (savedLocation) {
30       const { latitude, longitude } = JSON.parse(savedLocation);
31       setShowPermissionPanel(false);
32       fetchVideosByLocation(latitude, longitude);
33     } else {
34       setShowPermissionPanel(true);
35     }
36   }, []);
37

```



```

JS ExplorePage.js X JS HomePage.js JS LibraryPage.js
frontend > src > pages > JS ExplorePage.js ...
1 // frontend/src/pages/ExplorePage.js
2 import React, { useState, useEffect, useCallback } from 'react';
3 import axios from 'axios';
4 import MapComponent from '../components/MapComponent';
5 import '../styles/ExplorePage.css';
6 import { addVideoToUserHistory } from '../utils/history';
7 import LocationPermissionPanel from '../components/LocationPermissionPanel';
8 import AgePrompt from '../components/AgePrompt';
9 import VideoPlayer from '../components/VideoPlayer';
10
11 const API_URL = 'http://localhost:5000/api';
12
13 const ExplorePage = () => {
14   const [videos, setVideos] = useState([]);
15   const [loading, setLoading] = useState(true);
16   const [error, setError] = useState('');
17   const [locationName, setLocationName] = useState('');
18   const [mapCenter, setMapCenter] = useState(null);
19   const [showPermissionPanel, setShowPermissionPanel] = useState(null);
20   const [selectedVideoId, setSelectedVideoId] = useState(null);
21   const [agePromptOpen, setAgePromptOpen] = useState(false);
22   const [pendingVideoToOpen, setPendingVideoToOpen] = useState(null);
23
24   const fetchVideosForLocation = useCallback(async (lat, lng) => {
25     setLoading(true);
26     setError('');
27     try {
28       const geoRes = await axios.get(`${API_URL}/location/geocode`, { params: { lat, lon: lng } });
29       const fetchedLocationName = geoRes.data.locationName;
30       setLocationName(fetchedLocationName);
31
32       const videoRes = await axios.get(`${API_URL}/videos/search`, {
33         params: { searchTerm: 'videos', location: fetchedLocationName, maxResults: 15 },
34       });
35       setVideos(videoRes.data || []);
36     } catch (err) {
37       console.error('Error al cargar videos:', err);
38     }
39   }, []);
40
41   return (
42     <div>
43       <h1>Explorar Videos</h1>
44       <div>
45         <input type="text" value={searchQuery} onChange={<input>=> setSearchQuery} />
46         <button onClick={<button>=> fetchVideosForLocation(locationName)}>Buscar</button>
47       </div>
48       <div>
49         <h2>Resultados de búsqueda</h2>
50         <ul>
51           {videos.map((video, index) => (
52             <li key={index}>
53               <img alt={video.title} />
54               <div>{video.title}</div>
55               <div>{video.duration}</div>
56             </li>
57           ))}
58         </ul>
59       </div>
60     </div>
61   );
62 }

```

Se mejoró la presentación de las tarjetas de video con estilos responsivos y distintivo +18. Estilos definidos en HomePage.css y ExplorePage.css

INTEGRACIÓN DE BACKEND Y FRONTEND (HISTORIAL Y EXPLORACIÓN)

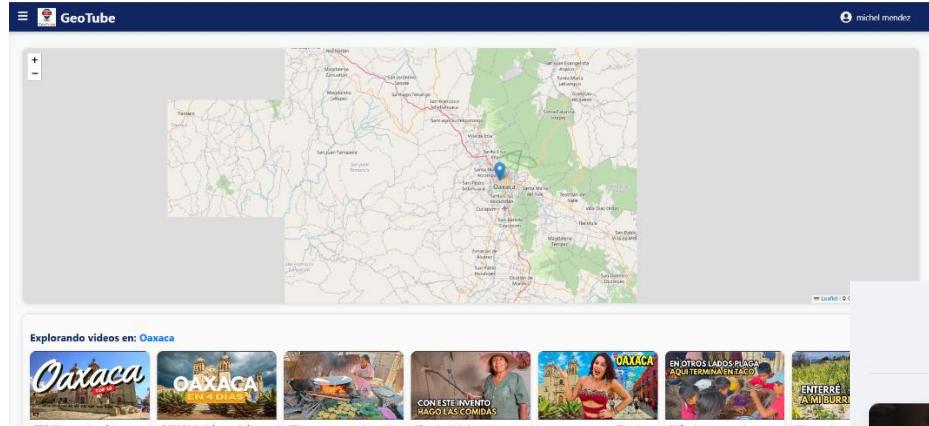
```
JS HomePage.js X JS ExplorePage.js JS LibraryPage.js
frontend > src > pages > JS HomePage.js > [edit] HomePage.js
  1 // frontend/src/pages/HomePage.js
  2 import React, { useState, useEffect } from 'react';
  3 import axios from 'axios';
  4 import { FaSearch, FaMapMarkedAlt } from 'react-icons/fa';
  5 import './styles/HomePage.css';
  6 import { addVideoToUserHistory } from '../utils/history';
  7 import AgePrompt from '../components/AgePrompt';
  8 import LocationPermissionPanel from '../components/LocationPermissionPanel';
  9
10 import VideoPlayer from '../components/VideoPlayer';
11
12 const API_URL = 'http://localhost:5000/api';
13
14 const HomePage = ({ user }) => [
15   const [videos, setVideos] = useState([]);
16   const [loading, setLoading] = useState(false);
17   const [error, setError] = useState('');
18   const [locationName, setLocationName] = useState('');
19   const [searchQuery, setSearchQuery] = useState('');
20   const [agePromptOpen, setAgePromptOpen] = useState(false);
21   const [pendingVideoOpen, setPendingVideoOpen] = useState(null);
22   const [showPermissionPanel, setShowPermissionPanel] = useState(true);
23
24   const [selectedVideoId, setSelectedVideoId] = useState(null);
25
26
27   useEffect(() => {
28     const savedLocation = localStorage.getItem('userLocation');
29     if (savedLocation) {
30       const { latitude, longitude } = JSON.parse(savedLocation);
31       setShowPermissionPanel(false);
32       fetchVideosByLocation(latitude, longitude);
33     } else {
34       setShowPermissionPanel(true);
35     }
36   }, [ ]);
37 ];
```

```
JS HomePage.js JS ExplorePage.js JS LibraryPage.js
frontend > src > pages > JS ExplorePage.js ...
1 // frontend/src/pages/ExplorePage.js
2 import React, { useState, useEffect, useCallback } from 'react';
3 import axios from 'axios';
4 import MapComponent from '../components/MapComponent';
5 import '../styles/ExplorePage.css';
6 import { addVideoToUserHistory } from '../utils/history';
7 import LocationPermissionPanel from '../components/locationPermissionPanel';
8 import AgePrompt from '../components/AgePrompt';
9 import VideoPlayer from '../components/VideoPlayer';
10
11 const API_URL = 'http://localhost:5000/api';
12
13 const ExplorePage = () => {
14   const [videos, setVideos] = useState([]);
15   const [loading, setLoading] = useState(true);
16   const [error, setError] = useState('');
17   const [locationName, setLocationName] = useState('');
18   const [mapCenter, setMapCenter] = useState(null);
19   const [showPermissionPanel, setShowPermissionPanel] = useState(null);
20   const [selectedVideoId, setSelectedVideoId] = useState(null);
21   const [agePromptOpen, setAgePromptOpen] = useState(false);
22   const [pendingVideoToOpen, setPendingVideoToOpen] = useState(null);
23
24   const fetchVideosForLocation = useCallback(async (lat, lng) => {
25     setLoading(true);
26     setError('');
27     try {
28       const geoRes = await axios.get(`${API_URL}/location/geocode`, { params: { lat, lon: lng } });
29       const fetchedLocationName = geoRes.data.locationName;
30       setLocationName(fetchedLocationName);
31
32       const videoRes = await axios.get(`${API_URL}/videos/search`, {
33         params: { searchTerm: 'videos', location: fetchedLocationName, maxResults: 15 },
34       });
35       setVideos(videoRes.data || []);
36     } catch (err) {
37       console.error('Error al cargar videos:', err);
38     }
39   }, []);
40
41   useEffect(() => {
42     fetchVideosForLocation();
43   }, [fetchVideosForLocation]);
44
45   return (
46     <div>
47       <h1>Explore Page</h1>
48       <p>Location: {locationName}</p>
49       <ul>
50         {videos.map((video, index) => (
51           <li key={index}>
52             {video.title} - {video.duration}
53           <button onClick={() => handleVideoClick(video)}>View</button>
54           <button onClick={() => handleAgePrompt(video)}>Age</button>
55           <button onClick={() => handleShareVideo(video)}>Share</button>
56           <button onClick={() => handleDeleteVideo(video)}>Delete</button>
57         </li>
58       )}
59     </ul>
60   );
61 }
62
63 export default ExplorePage;
```

```
JS HomePage.js JS ExplorePage.js JS LibraryPage.js ×
frontend > src > pages > JS LibraryPage.js [e] LibraryPage
  1 // frontend/src/pages/LibraryPage.js
  2
  3 import React, { useState, useEffect } from 'react';
  4 import axios from 'axios';
  5 import { addVideoToUserHistory } from '../utils/history';
  6 import './styles/LibraryPage.css';
  7 import VideoPlayer from '../components/VideoPlayer';
  8
  9 const API_URL = 'http://localhost:5000/api';
 10
 11 const LibraryPage = () => [
 12   const [history, setHistory] = useState([]);
 13   const [loading, setLoading] = useState(true);
 14   const [error, setError] = useState('');
 15   const [selectedVideoId, setSelectedVideoId] = useState(null);
 16
 17   useEffect(() => {
 18     const fetchHistory = async () => {
 19       setLoading(true);
 20       setError('');
 21       const token = localStorage.getItem('token');
 22       if (!token) {
 23         setError('Debes iniciar sesión para ver tu historial.');
 24         setLoading(false);
 25         return;
 26       }
 27
 28       try {
 29         const res = await axios.get(`${API_URL}/users/history`, {
 30           headers: {
 31             Authorization: `Bearer ${token}`,
 32           },
 33         });
 34         setHistory(res.data);
 35       } catch (err) {
 36         console.error(
 37           'Error al cargar el historial:',
```

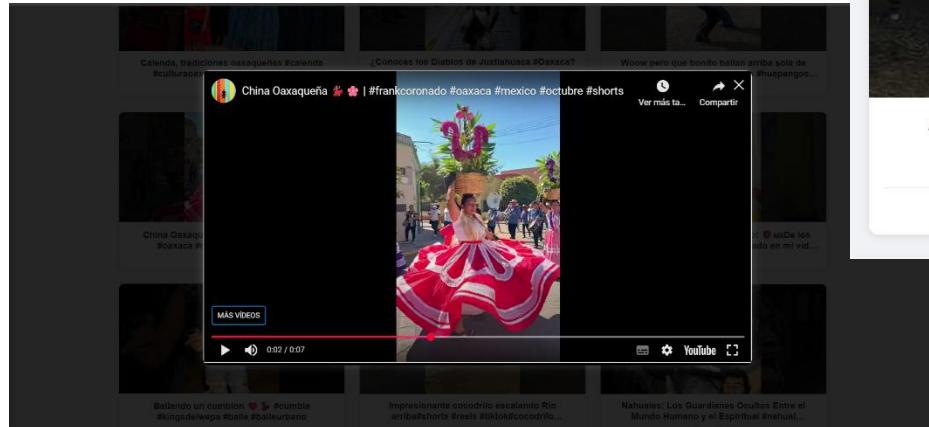
Conexión directa mediante Axios entre backend y frontend para cargar búsqueda, historial y exploración

PRUEBAS FUNCIONALES



Explorando videos en: Oaxaca

Oaxaca
OAXACA EN 4 DÍAS
CON ESTE INVENTO HAGO LAS COMIDAS
EN OTROS LADOS PUEBLO AQUÍ TERMINA EN TACOS
ENTERRE LA MIERDURA



China Oaxaqueña | #frankcoronado #oaxaca #mexico #octubre #shorts

MÁS VÍDEOS

002 / 007

Impresionante cocodrilo escalando Río arriba shorts #reels #tiktok#cocodrilo...

Nahuals: Los Guardianes Ocultos Entre el Mundo Humano y el Espiritual #manuel...

GeoTube

michel mendez

Geotube

Leather

Mi Historial de Videos



Arriba sola de Vega Oaxaca #oaxaca #baile #huapangos #chilenas #musica

MUCHACHO DE CAMPO OFICIAL

Visto el: 9 de octubre de 2025, 07:19