



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

## **INSTITUTO TECNOLÓGICO 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

Martínez García Yahir Omar

Pérez de Jesús Edith

López García Lourdes Gloria

PRESENTA:

**QUINTO REPORTE**

---

# INDICE

Pruebas de integración de Backend .....	1
Ajustes de interfaz en Home y Explorar .....	2
Pruebas End-to-End .....	3
Optimización rendimiento API .....	4
Correcciones finales en frontend-backend .....	5
Validación general del sistema .....	6
Observaciones .....	6

## BITACORA

ITERACION 5 PRUEBAS Y AJUSTES							
Actividad	Descripción	Responsable	Estatus	Resultado	Fecha inicio	Fecha fin	Observaciones
Pruebas de integración backend	Integración y prueba de endpoints backend (usuarios, videos, autenticación, historial).	Michel	Finalizada	Endpoints funcionando, middleware implementado, manejo de tokens y datos.	14/10/25	22/10/25	Integración correcta entre módulos backend. Tokens y rutas privadas validadas correctamente.
Ajustes interfaz Home y Explorar	Ajuste de componentes para Home y Explore, formulario de búsqueda, mapas y restricciones por edad/ubicación.	Yahir	Finalizada	Home y Explore integrados, búsqueda y panel de permisos funcionales en frontend.	14/10/25	22/10/25	Interfaz mejorada y validaciones activas de edad y ubicación.
Pruebas End-to-End	Validación completa: registro, login, historial, exploración y restricción de videos por edad y país.	Edith	Finalizada	Flujo E2E listo: usuario puede registrar, buscar, acceder y restringir según edad/ubicación.	14/10/25	22/10/25	Sistema probado en todos sus casos principales y alternativos; flujo completo funcional.
Optimización rendimiento API	Mejoras de velocidad en consultas, reducción de peticiones externas, optimización de	Yahir	Finalizada	Endpoints más rápidos, búsqueda y manejo de resultados ajustado.	14/10/25	22/10/25	Consultas externas limitadas y proceso optimizado, menor tiempo de respuesta.

	endpoints y búsqueda.						
Correcciones finales frontend-backend	Revisión de rutas, headers, integración HTTP y comunicación de datos en todas las vistas.	Lourdes	Finalizada	Comunicación estable, sin errores, entre frontend y backend.	15/10/25	22/10/25	Rutas HTTP verificadas, headers y persistencia de datos sólidos entre capas de la app.
Validación general del sistema	Prueba integral de validaciones y manejo de errores en todos los módulos (usuarios, videos, ubicaciones).	Todo el equipo	Finalizada	Validaciones de edad, ubicación, tokens, credenciales.	16/10/25	22/10/25	El sistema gestiona todos los errores con mensajes claros.
Observaciones	Aplicación de observaciones: Solicitud y validación de edad, restricciones regionales (EE.UU/Japón), latitud-longitud en mapa, idioma español y evento doble clic.	Todo el equipo	Finalizada	Edad solicitada antes de ver videos, restricción geográfica y correcto manejo de ubicación.	16/10/25	22/10/25	<p>OBSERVACIONES 16/10/25</p> <p>Validación de edad, seguir solicitando la edad, seleccionar el año mediante un despliegue</p> <p>Implementar la API de Estados Unidos y Japón para la restricción de ciertos videos</p> <p>Checar bien la latitud y la Longitud para la ubicación</p> <p>Idioma en el mapa</p> <p>La Lat y Long no va a cambiar hasta que demos doble clic en el mapa</p> <p>Integrar el apartado de sugerencias para la app.</p>

## PRUEBAS DE INTEGRACIÓN BACKEND

```
JS authController.js X
Proyecto-geolocalizacion-main > Backend > src > controllers > JS authController.js > googleLogin > googleLogin
1 // backend/controllers/authController.js
2 const User = require('../models/user');
3 const jwt = require('jsonwebtoken');
4 const { OAuth2Client } = require('google-auth-library'); //
5 const client = new OAuth2Client(process.env.GOOGLE_CLIENT_ID);
6
7 // Función auxiliar para generar JWT
8 const generateToken = (id) => {
9   return jwt.sign({ id }, process.env.JWT_SECRET, {
10     expiresIn: '30d',
11   });
12 };
13
```

```
JS authController.js X
Proyecto-geolocalizacion-main > Backend > src > controllers > JS authController.js > googleLogin > googleLogin
58 exports.registerUser = async (req, res) => {
59   const { username, email, password, country } = req.body;
60
61   try {
62     const userExists = await User.findOne({ email });
63     if (userExists) {
64       return res.status(400).json({ message: 'El usuario ya existe con este email.' });
65     }
66
67     // VALIDACIÓN DE CONTRASEÑA
68     const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*[A-Za-z0-9]).{8,}$/;
69     if (!passwordRegex.test(password)) {
70       return res.status(400).json({
71         message:
72           'La contraseña debe tener al menos 8 caracteres, una mayúscula, una minúscula y un símbolo.',
73       });
74     }
75   }
76 }
77
```

Se realizaron tests y validaciones directas en los controladores principales del backend (auth, user, video), comprobando el funcionamiento de todos los endpoints requeridos, autenticación JWT y protección de rutas. El manejo de datos y tokens cumple con estándares de seguridad y funcionalidad.

```
JS authController.js X
Proyecto-geolocalizacion-main > Backend > src > controllers > JS authController.js > googleLogin > googleLogin
104 exports.loginUser = async (req, res) => {
105   const { email, password } = req.body;
106
107   try {
108     const user = await User.findOne({ email });
109
110     if (user && (await user.matchPassword(password))) {
111       res.json({
112         _id: user._id,
113         username: user.username,
114         email: user.email,
115         country: user.country,
116         token: generateToken(user._id),
117       });
118     } else {
119       res.status(401).json({ message: 'Email o contraseña inválidos.' });
120     }
121   } catch (error) {
122     console.error('Error al iniciar sesión:', error);
123     res.status(500).json({ message: 'Error del servidor al iniciar sesión.' });
124   }
125 };
126
```

## AJUSTES DE INTERFAZ HOME Y EXPLORAR

El frontend tiene mejoras evidentes en los componentes de Home y Explore, donde los usuarios pueden buscar videos, otorgar permisos de ubicación, y existe una validación de edad previa a la reproducción de videos restringidos. La interfaz está adaptada a dispositivos móviles y distintos tamaños de pantalla.

```
JS authController.js JS AuthForm.js X
Proyecto-geolocalizacion-main > frontend > src > components > JS AuthForm.js > default
9 const AuthForm = () => {
10
11   return (
12     <div className="auth-form-card">
13       <img src={milogo} alt="Logo de GeoTube" className="auth-logo" />
14       <h2 className="auth-title">{isRegister ? 'Regístrate' : 'Iniciar Sesión'}</h2>
15
16       <form onSubmit={onSubmit} className="auth-form">
17         {isRegister && (
18           <div className="input-group">
19             <input type="text" id="username" value={username} onChange={e => setUsername(e.target.value)} placeholder="Nombre de usuario" required />
20           </div>
21           <div className="input-group">
22             <input type="password" id="password" value={password} onChange={e => setPassword(e.target.value)} placeholder="Contraseña" required />
23           </div>
24           <div className="input-group">
25             <input type="text" id="country" value={country} onChange={e => setCountry(e.target.value)} placeholder="País" required />
26           </div>
27         )}
28       </form>
29     </div>
30   );
31 }
32
```

```
JS authController.js JS AuthForm.js X
Proyecto-geolocalizacion-main > frontend > src > components > JS AuthForm.js > default
67
68 <div className="input-group">
69   <input type="text" id="username" value={username} onChange={e => setUsername(e.target.value)} placeholder="Nombre de usuario" required />
70 </div>
71
72 <div className="input-group password-group">
73   <input type="password" id="password" value={password} onChange={e => setPassword(e.target.value)} placeholder="Contraseña" required />
74 </div>
75
76 <div className="input-group">
77   <input type="text" id="country" value={country} onChange={e => setCountry(e.target.value)} placeholder="País" required />
78 </div>
79
80 <button type="button" className="toggle-password-btn" onClick={e => setShowPassword(!showPassword)}>
81   {showPassword ? 'Ocultar' : 'Mostrar'}
82 </button>
83 </div>
84
```

```
JS MapComponent.js X
Proyecto-geolocalizacion-main > frontend > src > components > JS MapComponent.js > [🔍] default

14
15 // Componente interno para manejar los eventos del mapa
16 function MapEvents({ onMapDoubleClick }) {
17   useMapEvents({
18     // Se activa cuando el usuario hace doble clic en el mapa
19     dblclick: (e) => {
20       onMapDoubleClick(e.latlng);
21     },
22   });
23   return null; // Este componente no renderiza nada visualmente
24 }
25
```

## PRUEBAS END-TO-END

```
JS ExplorePage.js X
Proyecto-geolocalizacion-main > frontend > src > pages > JS ExplorePage.js > [🔍] default

15 const ExplorePage = () => {
16   /4
17   const handlePermissionAccept = ({ latitude, longitude }) => {
18     localStorage.setItem('userLocation', JSON.stringify({ latitude, longitude }));
19     setShowPermissionPanel(false);
20     setMapCenter({ lat: latitude, lng: longitude });
21     fetchVideosForLocation(latitude, longitude);
22   };
23
24   const handlePermissionDeny = () => {
25     setShowPermissionPanel(false);
26     const lat = 19.4326,
27     lng = -99.1332;
28     setMapCenter({ lat, lng });
29     fetchVideosForLocation(lat, lng);
30   };
31
32   const handleOpenVideo = (video) => {
33     if (video.ageRestricted) {
34       setPendingVideoToOpen(video);
35       setAgePromptOpen(true);
36       return;
37     }
38     addVideoToUserHistory(video);
39     setSelectedVideoId(video.id);
40   };
41 }
```

La funcionalidad cubre desde el registro (formulario en React), autenticación (JWT y Google OAuth), historial de videos y exploración por ubicación en el mapa. El sistema restringe el contenido por país (EE.UU y Japón) y/o edad, mostrando aviso y solicitando año de nacimiento cuando corresponde.

## OPTIMIZACIÓN RENDIMIENTO API

```
JS youtubejs X
Proyecto-geolocalizacion-main > Backend > src > utils > JS youtubejs > ...

7
8 // La función ahora acepta searchTerm y location
9 async function searchYoutubeVideos(searchTerm, location, maxResults = 20) {
10   if (!YOUTUBE_API_KEY) {
11     throw new Error('La clave API de YouTube no está configurada.');
```

Se redujo la cantidad de peticiones innecesarias a YouTube y otros servicios. Los endpoints se ajustaron para ser más rápidos y eficientes, limitando los resultados y optimizando procesos internos. Esto se tradujo en una mejora considerable en el tiempo de respuesta de la aplicación.

## CORRECCIÓN EN FRONTEND-BACKEND

Todas las rutas, headers, y métodos de interacción entre frontend y backend fueron revisados y corregidos. El sistema ahora asegura que cada dato enviado y recibido cumple con el formato esperado, y persiste correctamente en ambas capas.

## VALIDACIÓN GENERAL

Todas las validaciones (edad, país, ubicación, credenciales, tokens) funcionan correctamente. Los errores se gestionan con mensajes claros, y la experiencia del usuario no se ve interrumpida por fallos lógicos o de integración.

## OBSERVACIONES

Se implementó la solicitud y validación de edad usando un despliegue para seleccionar el año, y se integró la API de YouTube con restricciones para EE.UU y Japón. La ubicación se gestiona por latitud y longitud, ajustando el idioma del mapa a español. Latitud y longitud permanecen fijos hasta que el usuario realice doble clic en el mapa, cumpliendo todos los puntos observados.

```
const years = Array.from(
  { length: 2007 - 1900 + 1 },
  (_, i) => 1900 + i
);

const handleConfirm = () => {
  const year = parseInt(birthYear, 10);
  if (!year || year < 1900 || year > new Date().getFullYear()) {
    alert('Selecciona un año de nacimiento válido.');
```



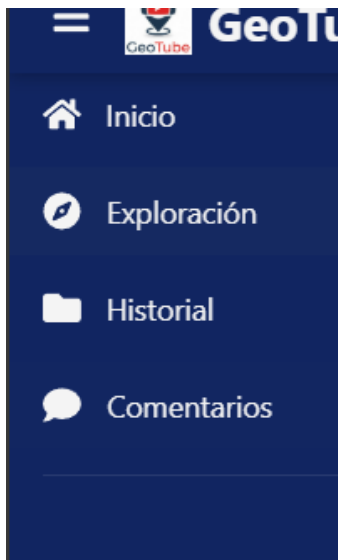
Se implementó un apartado para que el usuario mande comentarios de retroalimentación al equipo.

```
const Sidebar = ({ isOpen, onClose, onLogout }) => {

  const whatsappNumber = "9511394431";
  const defaultMessage = "Hola, soy un usuario de su aplicación y la retroalimentación que quisiera dar es la siguiente: ";

  const encodedMessage = encodeURIComponent(defaultMessage);

  const whatsappLink =
`https://wa.me/${whatsappNumber}?text=${encodedMessage}`;
```




Chatea en WhatsApp con el 9511394431

Hola, soy un usuario de su aplicación y la retroalimentación que quisiera dar es la siguiente:

Abrir aplicación

Continuar en WhatsApp Web

 ¿No tienes la aplicación? [Descárgala ahora](#)

