

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

Martínez García Yahir Omar

Pérez de Jesús Edith

López García Lourdes Gloria

PRESENTA:

MANUAL DE APIS

VISIÓN GENERAL DE LA ARQUITECTURA API

Frontend: Aplicación React desplegada en Vercel, utiliza variables de entorno para consumir servicios del backend y APIs externas (YouTube, Google Maps).

Backend: Node.js/Express alojado en Render. Expone endpoints RESTful para autenticación, usuarios, videos y ubicaciones.

APIs Externas: Integración de YouTube Data API (consulta información de videos) y Google Maps Platform (visualización y geolocalización en el frontend).

ENDPOINTS DEL BACKEND EXPRESS (RENDER)

- Autenticación

POST `/api/auth/login` : Valida credenciales de usuario y devuelve token JWT.

POST `/api/auth/register` : Registra nuevo usuario.

GET `/api/auth/me` : Obtiene info básica del usuario autenticado (requiere JWT).

- Usuarios

GET `/api/users` : Lista todos los usuarios (requiere permisos).

GET `/api/users/:id` : Consulta por usuario específico.

PUT `/api/users/:id` : Actualiza información de usuario (restringido).

DELETE `/api/users/:id` : Elimina un usuario (restringido).

- Videos

GET `/api/videos` : Lista los videos disponibles en la base de datos.

GET `/api/videos/:id` : Detalles de un video por su ID.

Implementa consumo indirecto de YouTube API mediante lógica personalizada en el backend

- Ubicaciones (Locations)

GET `/api/locations` : Devuelve lista de ubicaciones almacenadas.

GET `/api/locations/:id` : Información detallada de una ubicación.

- YouTube API (Data API v3)

GET `/api/youtube/search?q=PALABRA` : Busca videos en YouTube a partir de un término y devuelve datos seleccionados. El backend procesa la petición, consulta YouTube y filtra la respuesta.

GET `/api/youtube/video/:id` : Detalles extendidos de un video de YouTube por su ID.

Ambos requieren haber configurado la API Key de YouTube como variable de entorno en Render*

- Geocodificación

GET `/api/geocode?address=DIRECCION` : Traduce una dirección en coordenadas usando la Geocoding API de Google Maps.

Requiere API Key de Google Maps configurada en Render

CONSUMO DE APIS EN EL FRONTEND REACT (VERCEL)

El frontend está diseñado con componentes que interactúan con el backend y con APIs externas:

- Configuración:

- Variables de entorno en Vercel como `REACT_APP_API_URL` (URL de Render) y `REACT_APP_GOOGLE_MAPS_API_KEY`.

Usuarios, Autenticación: Formularios que envían datos mediante `fetch`/`Axios` al backend, gestionan tokens y sesiones.

Videos: El componente de biblioteca carga datos de `/api/videos`, e implementa funcionalidades como búsqueda que pueden consumir video info de YouTube usando los endpoints backend que acceden a la YouTube API.

Google Maps: Uso de la biblioteca `@vis.gl/react-google-maps`, integrando mapas interactivos con marcadores de ubicaciones y rutas.

Geocodificación: Formularios en el frontend pueden consumir `/api/geocode` para transformar direcciones en coordenadas, usando la respuesta para posicionar marcadores en el mapa.

CONFIGURACIÓN DE VARIABLES DE ENTORNO

Backend (Render):

- `YOUTUBE_API_KEY`: Clave para consumir YouTube Data API desde el servidor.
- `GOOGLE_MAPS_API_KEY`: Clave para consultas de Geocoding y otros servicios Maps.

Frontend (Vercel):

- `REACT_APP_API_URL`: URL pública del backend en Render.
- `REACT_APP_GOOGLE_MAPS_API_KEY`: Clave para visualizar Google Maps en el frontend.

INTEGRACIÓN Y CONSUMO: EJEMPLO GENERAL

Buscar videos en YouTube desde el frontend: Formulario en React envía la búsqueda al backend `/api/youtube/search?q=palabra`; el backend consulta la YouTube API y devuelve resultados filtrados que la interfaz muestra.

Visualizar ubicaciones: Componente de mapa en React obtiene coordenadas desde `/api/locations` (backend), las coloca como marcadores usando Google Maps API. Si el usuario ingresa una dirección, se consume `/api/geocode` y se añade un nuevo marcador en el mapa.

Autenticación y sesiones: El frontend utiliza endpoints de login y registro, almacena el token JWT, y muestra contenido protegido según el estado de la sesión.

DESPLIEGUE

Render (backend):

- Proyecto conectado a GitHub, configurado como Web Service. Claves de APIs y otras variables de entorno gestionadas en el panel de Render.
- Prueba de endpoints mediante Postman o llamados desde el frontend.

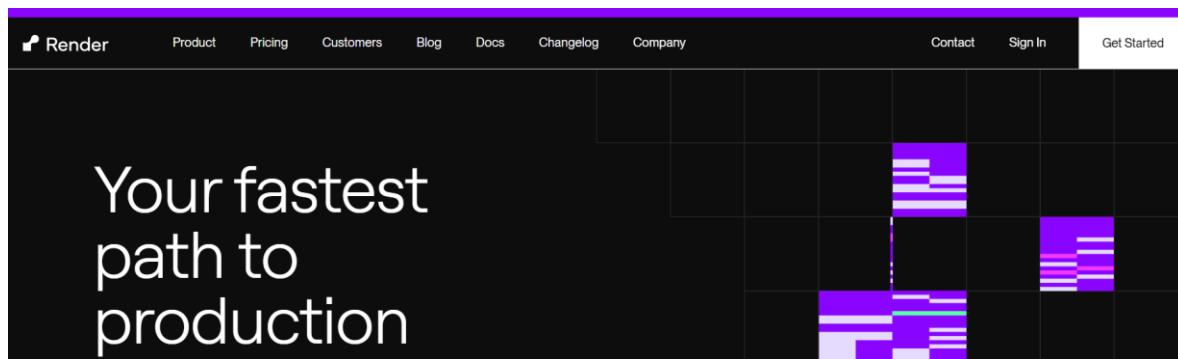
Vercel (frontend):

- Proyecto React vinculado a GitHub, variables de entorno configuradas en el dashboard.
- Pruebas directas en navegador y mediante herramientas de desarrollo de React.

PROCESO DE DESPLIEGUE

CONFIGURAR EL BACKEND EN RENDER.COM

Ingresá a [render.com](#), haz clic en “Get started for free” o “Sign up”, y regístrate con GitHub.



Creamos un nuevo Web Service haciendo clic en "New" > "Web Service". Conectamos nuestro repositorio de GitHub donde tenemos nuestro backend.

The screenshot shows the Render interface. At the top, there's a navigation bar with a projects icon and the word 'Projects'. Below it is a section titled 'Overview'. Under 'Overview', there's a 'Projects' section. It shows one project named 'My project' with a status message: 'All services are up and running'. To the right of this, there's a dashed box with a '+ Create new project' button. The overall interface is clean and modern, typical of a cloud-based development platform.

Si el backend está dentro de una carpeta específica (por ejemplo, “backend”), lo especificamos en Root Directory.

This screenshot shows the Render logs interface. At the top, there's a message: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' and a link to 'Upgrade now'. Below that, it shows the date 'October 23, 2025 at 2:39 PM' and a 'Live' status indicator. A log entry '858cfa6 Logs en back funcional' is listed. The main area is a log viewer with tabs for 'All logs' and 'Search'. It shows a date range from 'Oct 23, 2:38 PM - 2:42 PM' to 'CST'. A message 'No logs to show' is displayed, along with a note: 'The time range is outside your log retention period. Consider setting up a log stream or increasing your workspace.' In the bottom right corner, there's a link to 'Activar Windows' and a note about configuration for activating Windows.

This screenshot shows the Render deployment configuration interface. It includes sections for 'Ignored Paths', 'Build Command' (set to 'Backend/ \$ npm install'), 'Pre-Deploy Command' (optional), 'Start Command' (set to 'Backend/ \$ node server.js'), 'Auto-Deploy' (disabled), and 'Deploy Hook' (private URL). Each section has an 'Edit' button next to its input field.

Llenamos los datos de configuración: Name, Region, Branch, Runtime (Node), Build Command (npm install), Start Command (node server.js o npm start si se tiene el script en package.json).

Configuramos las Variables de Entorno importantes (PORT, MONGODB_URI, JWT_SECRET, YOUTUBE_API_KEY, GOOGLE_CLIENT_ID).

The screenshot shows the 'Environment' section of the Render interface. At the top right are buttons for '+ Create environment group', 'Export', and 'Edit'. Below this is a table titled 'Environment Variables' with the following data:

KEY	VALUE
GOOGLE_CLIENT_ID
JWT_SECRET
MONGODB_URI
PORT
YOUTUBE_API_KEY

Seleccionamos el plan “Free” en Instance Type para evitar costos.

Hacemos clic en “Create Web Service”. Render instalará dependencias y desplegará la aplicación.

Cuando el servicio sea “Live”, tendremos una URL como <https://geotube-back.onrender.com>. Copiamos esta URL, será nuestra nueva API_BASE_URL.

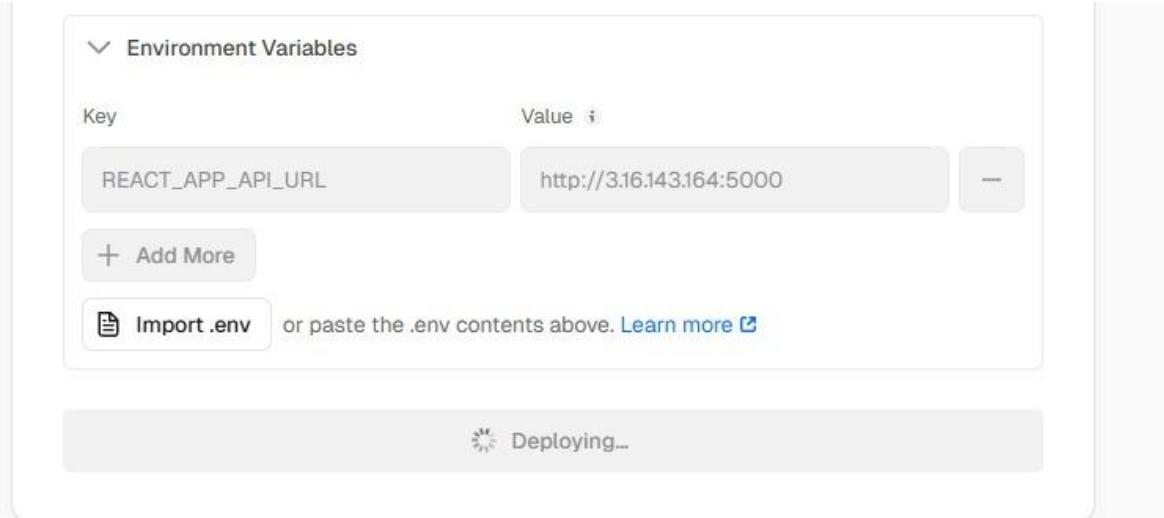
DESPLEGAR EL FRONTEND EN VERCEL

Inicia sesión en Vercel, haz clic en "Add New..." > "Project", buscamos y seleccionamos el repositorio de GitHub.

The screenshot shows the 'New Project' setup in Vercel. It includes fields for 'Importing from GitHub' (set to 'Yahirmtz12/GEOTUBE'), 'Project Name' ('geotube'), 'Vercel Team' ('Yahir Mtz's projects'), 'Framework Preset' ('Create React App'), 'Root Directory' ('frontend'), and build/install commands. The 'Build Command' is set to 'npm run build' or 'react-scripts build', 'Output Directory' is 'build', and 'Install Command' is 'yarn install', 'pnpm install', 'npm install', or 'bun install'.

Vercel detecta automáticamente que es una app de React (create-react-app). Normalmente no necesitamos modificar la configuración de build.

Vamos a “Environment Variables” y definimos la variable REACT_APP_API_URL para indicar la URL de tu backend (puede ser la IP de nuestra instancia o el enlace HTTPS de Render). Hacemos clic en “Deploy”.



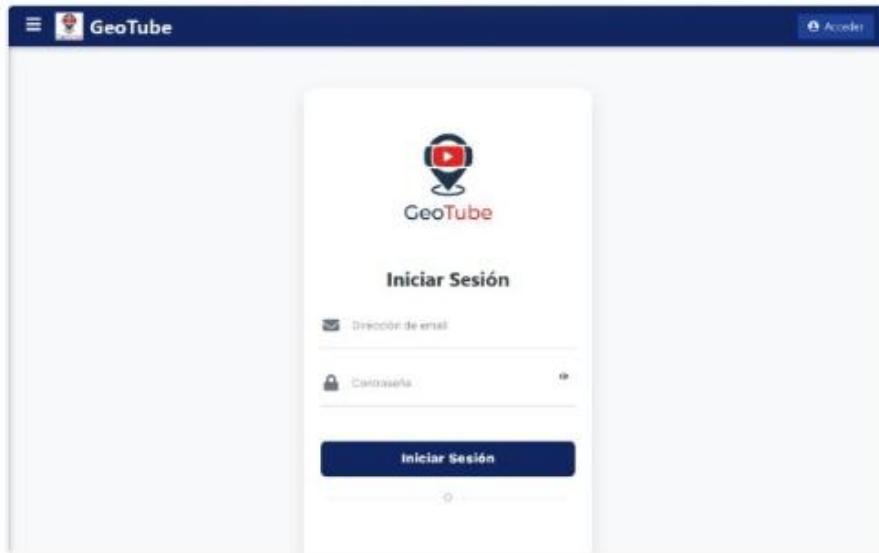
Actualizamos nuestro código local para que use la variable de entorno en vez de <http://localhost:5000>.

Hacemos git add ., git commit y git push para subir este cambio. Vercel detecta el push y despliega automáticamente tu frontend.

Una vez finalizado el despliegue, obtenemos el enlace público de tu frontend como <https://geotube-delta.vercel.app/>

Congratulations!

You just deployed a new project to  Yahir Mtz's projects.



Next Steps

Instant Previews

Push a new branch to preview changes instantly

Add Domain

Add a custom domain to your project



Enable Speed Insights

Track how users experience your site over time



[Continue to Dashboard](#)