

## **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:

**SEXTO REPORTE**

## INDICE

Bitácora .....	1
Proceso de despliegue .....	2
Configurar el Backend en RENDER.COM .....	3
Desplegar Frontend en VERCCEL .....	4
Pruebas de usabilidad con usuarios .....	5
Comentarios .....	6

## BITACORA

ITERACION 6 DESPLIEGUE							
Actividad	Descripción	Responsable	Estatus	Resultado	Fecha inicio	Fecha fin	Observaciones
Refactorización backend	Limpieza y organización del código backend, mejora de módulos y dependencias.	Michel	Finalizada <span style="color: green;">✓</span>	Backend estable, código modular.	16/10/25	24/10/25	El backend listo y optimizado para despliegue.
Configuración de variables de entorno	Definición en Render de variables (.env): MongoDB, JWT, Google Client, Youtube API, PORT.	Lourdes	Finalizada <span style="color: green;">✓</span>	Variables configuradas, ambiente seguro.	16/10/25	24/10/25	Evidencia Render: Environment Variables configuradas.
Documentación del backend	Generación de guía para endpoints y uso de la API REST.	Michel	Finalizada <span style="color: green;">✓</span>	Documentación añadida al repositorio.	16/10/25	24/10/25	Documentación disponible para consulta del equipo.
Refactorización frontend	Organización y modularización del código React, mejora de componentes.	Edith	Finalizada <span style="color: green;">✓</span>	Frontend optimizado y limpio.	16/10/25	24/10/25	Frontend listo para build de producción.
Build de producción para React	Ejecución de build para React con variables y rutas definidas.	Edith	Finalizada <span style="color: green;">✓</span>	Build generado correctamente.	16/10/25	24/10/25	Configuración de build evidenciada en Vercel.
Despliegue frontend	Uso de Vercel con las configuraciones y despliegue exitoso.	Yahir	Finalizada <span style="color: green;">✓</span>	Frontend publicado y funcional en web.	16/10/25	24/10/25	Link de Vercel: aplicación accesible públicamente.
Pruebas de integración final backend-frontend	Validación completa: registro, autenticación, uso de endpoints,	Yahir	Finalizada <span style="color: green;">✓</span>	Flujo E2E exitoso y validado.	16/10/25	24/10/25	Flujos cubiertos, integración

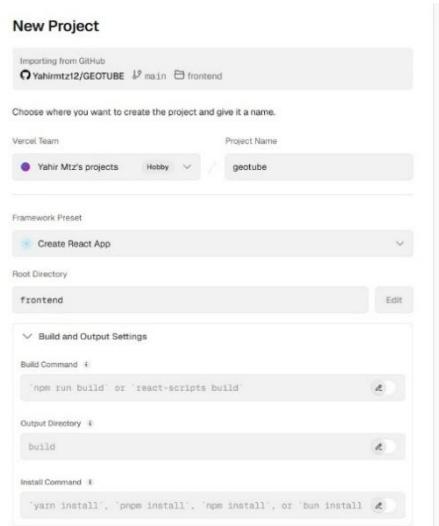
	permisos y restricciones.						validada (ver capturas).
Pruebas de usabilidad con usuarios	Sesiones de prueba con usuarios finales, retroalimentación de UX.	Todos	Pendiente	Mejoras sugeridas implementadas, usabilidad confirmada.	16/10/25	28/10/25	
Validación final y entrega	Verificación del funcionamiento completo de la aplicación para entrega final.	Todos	Finalizada 	Sistema listo para entrega.	16/10/25	29/10/25	Documentación y links de despliegues incluidos.

# PROCESO DE DESPLIEGUE

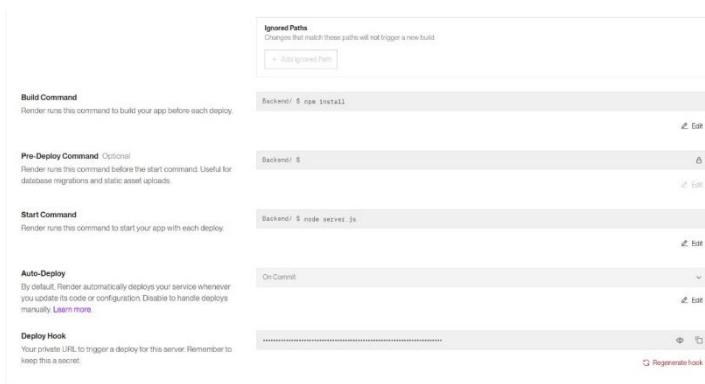
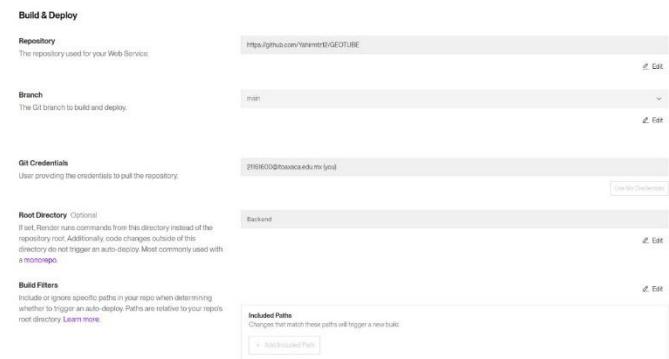
## CONFIGURAR EL BACKEND EN RENDER.COM

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

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



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



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).

Seleccionamos el plan “Free” en Instance Type para evitar costos.  
Hacemos clic en “Create Web Service”. Render instalará dependencias y desplegará la aplicación.

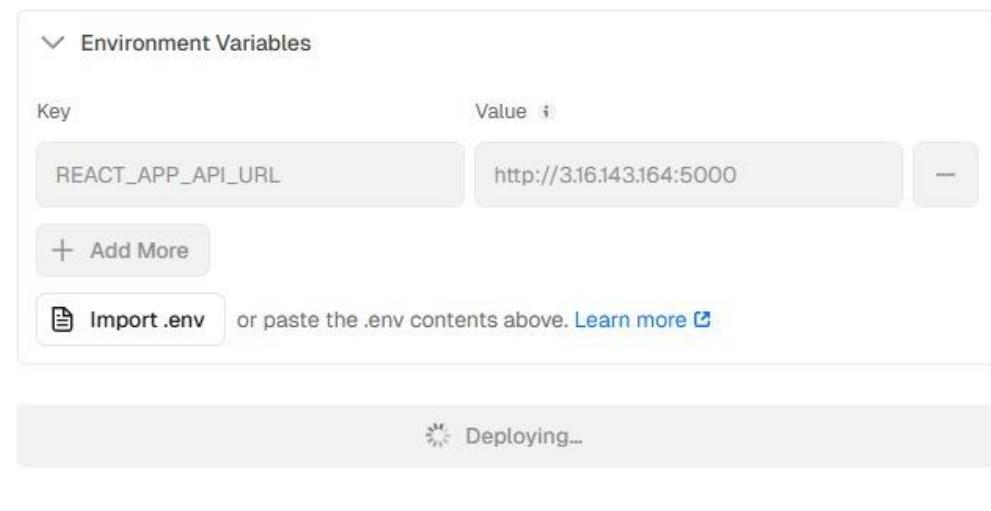
Cuando el servicio sea “Live”, tendremos una URL como <https://geotube-back.onrender.com>. Copiamos esta URL, será nuestra nueva API\_BASE\_URL.

## DESPLEGAR EL FRONTEND EN VERCEL

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

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/>

Evidencia adicional del proceso (comandos y pantallas de la VM)

Mostramos comandos ejecutados en la terminal (como git clone, apt install, npm install, actualización de paquetes, uso de PM2).

Verificamos la instancia corriendo en AWS y la configuración de nuestra cuenta.

```
C:\Users\Pollos Junior\Downloads>ssh -i "geotube-key.pem" ubuntu@3.16.143.164
The authenticity of host '3.16.143.164 (3.16.143.164)' can't be established.
ED25519 key fingerprint is SHA256:BLUHLWicOkhb+XEsbXMYGVKiEXdLV3NGBD95e3ZpjSI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
```

```
ubuntu@ip-172-31-29-219:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1535 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [292 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1496 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2164 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [488 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7136 B]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1249 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [207 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1904 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2041 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [462 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:27 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
9% [9 Packages stored 0 B]
```

```
ubuntu@ip-172-31-29-219:~$ sudo apt install -y nodejs npm git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nodejs is already the newest version (18.19.1+dfsg-6ubuntu5).
npm is already the newest version (9.2.0~ds1-2).
git is already the newest version (1:2.43.0-1ubuntu7.3).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
```

```
ubuntu@ip-172-31-29-219:~$ git clone https://github.com/Miche127/Proyecto-geolocalizacion.git
Cloning into 'Proyecto-geolocalizacion'...
remote: Enumerating objects: 40441, done.
remote: Counting objects: 100% (148/148), done.
remote: Compressing objects: 100% (119/119), done.
remote: Total 40441 (delta 53), reused 97 (delta 28), pack-reused 40293 (from 1)
Receiving objects: 100% (40441/40441), 138.84 MiB | 23.79 MiB/s, done.
Resolving deltas: 100% (9223/9223). done.
ubuntu@ip-172-31-29-219:~$ cd Proyecto-geolocalizacion/
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion$ ls
Backend README.md frontend
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion$ cd Backend/
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion/Backend$ npm install

up to date, audited 144 packages in 732ms

25 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion/Backend$ sudo npm install -g pm2
```

```
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion/Backend$ pm2 start server.js --name geotube-api
```



Runtime Edition

PM2 is a Production Process Manager for Node.js applications  
with a built-in Load Balancer.

Start and Daemonize any application:  
\$ pm2 start app.js

Load Balance 4 instances of api.js:  
\$ pm2 start api.js -i 4

Monitor in production:  
\$ pm2 monitor

Make pm2 auto-boot at server restart:  
\$ pm2 startup

To go further checkout:  
<http://pm2.io/>

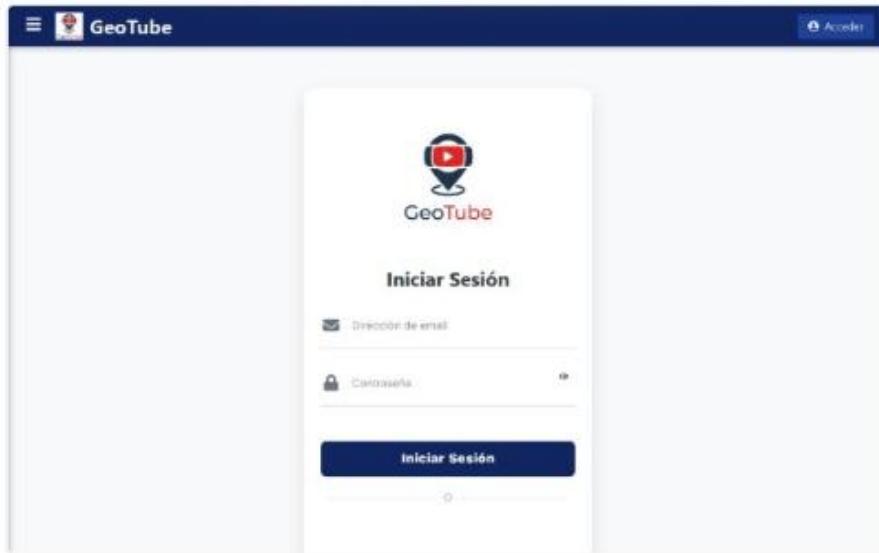
```
[PM2] Spawning PM2 daemon with pm2_home=/home/ubuntu/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/ubuntu/Proyecto-geolocalizacion/Backend/server.js in fork_mode (1 instance)
[PM2] Done.
```

<code>id</code>	<code>name</code>	<code>mode</code>	<code>�</code>	<code>status</code>	<code>cpu</code>	<code>memory</code>
0	geotube-api	Fork	0	online	0%	30.9mb

```
ubuntu@ip-172-31-29-219:~/Proyecto-geolocalizacion/Backend$ |
```

## 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](#)

# PRUEBAS DE USUABILIDAD CON USUARIOS

The collage consists of five images illustrating the user testing process for GeoTube:

- Top Left:** A login screen titled "Iniciar Sesión" (Log In) featuring a "GeoTube" logo with a play button icon inside a location pin. It includes fields for "Dirección de email" (Email address) and "Contraseña" (Password), a "Iniciar Sesión" (Log In) button, and a "Iniciar sesión con Google" (Log in with Google) button.
- Top Right:** A screenshot of a Google Chrome window titled "Iniciar sesión con Google" (Log in with Google). It shows a list of accounts under "Selecciona una cuenta" (Select a account): "michel mendez" (luisamichel271202@gmail.com), "LUISA MICHEL MENDEZ MENDOZA" (20161271@itoaxaca.edu.mx), "Michel" (mech73m@gmail.com), and "Usar otra cuenta" (Use another account).
- Middle Left:** A modal dialog titled "Información sobre tu ubicación" (Information about your location). It explains that GeoTube will request location to provide local content and emphasizes privacy. It includes "Permitir" (Allow) and "Ahora no" (Not now) buttons.
- Middle Right:** A map-based interface titled "GeoTube". It shows a map of a region with several locations labeled, including "Villa de Etla". Below the map is a search bar with the placeholder "Buscar videos en esta área del mapa..." (Search for videos in this area of the map) and a "Feedback" button.
- Bottom:** A grid of video thumbnail cards titled "Recomendaciones según tu ubicación: #Oaxaca" (Recommendations based on your location: #Oaxaca). Each card includes a thumbnail image, a caption, and hashtags. Examples include "Arriba sola de Vega Oaxaca #oaxaca #baile #huipangos #chilenas #música MUCHACHO DE CAMPO OFICIAL", "Así baila la gente de Oaxaca en los bailes #música #baile #partytime #reading #book Posts Marido", and "CHULADA DE SOLTECA ►►► #baile #oaxacayguerrero #bailesmexicanos... El Oaxaca De Oregon".

**Mi Historial de Videos**

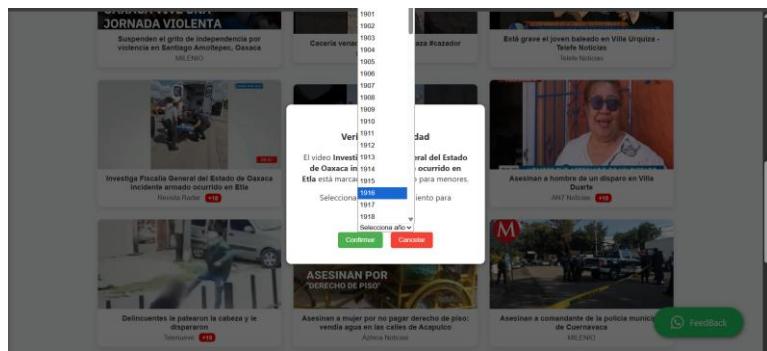
Los bailes más hermosos de Oaxaca | Flor de Piña  
♥ #oaxaca #mexico #oaxacacity #oaxacaturismo

Callejando Mx

Visto el: 16 de octubre de 2025, 06:57

Reina la violencia en el estado de Oaxaca en México

Univision Noticias +18



## COMENTARIOS

(LA APLICACION AUN NO AH SIDO PROVADA POR LOS COMPAÑEROS, HASTA DONDE NOSOTROS LA PROBAMOS, FUNCIONA CORRECTAMENTE)