Claro, con gusto te explico el proceso general para poner tu proyecto en internet. Tu aplicación se compone de dos partes principales que deben manejarse de forma separada:

1. **server (Backend):** Es una aplicación de Node.js que se conecta a una base de datos y ejecuta la lógica de negocio. Necesita estar corriendo constantemente en un servidor.
2. **client (Frontend):** Es una aplicación de React (hecha con Vite). Esta no se "ejecuta" de la misma forma que el backend. En su lugar, se "compila" a archivos estáticos (HTML, CSS, JavaScript) que un servidor web simplemente entrega al navegador del usuario.

Aquí te presento el paso a paso para desplegar ambas partes.

**Paso 1: Preparar el Backend (server) para Producción**

Antes de subir tu código, debes asegurarte de que esté listo para un entorno de producción.

1. **Variables de Entorno:** No debes tener datos sensibles (como contraseñas de base de datos, secretos para tokens JWT, etc.) directamente en el código. Usa variables de entorno.
   * Crea un archivo .env en tu carpeta server.
   * Añade las variables que necesites, por ejemplo:
   * DATABASE\_URL="la\_url\_de\_tu\_base\_de\_datos\_en\_la\_nube"
   * PORT=3000
   * JWT\_SECRET="un\_secreto\_muy\_largo\_y\_dificil"
   * Asegúrate de que tu código ([database.js](code-assist-path:d:\Secretar%C3%ADaDeSeguridad\2025\05.%20Sistemas\08082025%20SIHUR%20Sistema%20de%20Informaci%C3%B3n%20para%20la%20gesti%C3%B3n%20del%20hurto%20en%20Medell%C3%ADn\server\database.js), [index.js](code-assist-path:d:\Secretar%C3%ADaDeSeguridad\2025\05.%20Sistemas\08082025%20SIHUR%20Sistema%20de%20Informaci%C3%B3n%20para%20la%20gesti%C3%B3n%20del%20hurto%20en%20Medell%C3%ADn\server\index.js)) lea estas variables usando process.env.NOMBRE\_VARIABLE. Necesitarás instalar el paquete dotenv (npm install dotenv) y cargarlo al inicio de tu [index.js](code-assist-path:d:\Secretar%C3%ADaDeSeguridad\2025\05.%20Sistemas\08082025%20SIHUR%20Sistema%20de%20Informaci%C3%B3n%20para%20la%20gesti%C3%B3n%20del%20hurto%20en%20Medell%C3%ADn\server\index.js) con require('dotenv').config();.
2. **Configurar CORS (Cross-Origin Resource Sharing):** Tu frontend estará en un dominio diferente al de tu backend. El backend debe permitir explícitamente las solicitudes que vienen del frontend.
   * Instala el paquete cors: npm install cors.
   * En tu [index.js](code-assist-path:d:\Secretar%C3%ADaDeSeguridad\2025\05.%20Sistemas\08082025%20SIHUR%20Sistema%20de%20Informaci%C3%B3n%20para%20la%20gesti%C3%B3n%20del%20hurto%20en%20Medell%C3%ADn\server\index.js), configúralo para que acepte peticiones desde el futuro dominio de tu cliente:
   * const cors = require('cors');
   * *// ... más código de tu app*
   * app.use(cors({
   * origin: 'https://la-url-de-tu-frontend.com' *// ¡Importante! Esta será la URL de tu app cliente desplegada*
   * }));

**Paso 2: Desplegar el Backend (server)**

Necesitas un servicio de "Platform as a Service" (PaaS) que pueda ejecutar aplicaciones de Node.js.

* **Opciones Populares (con capas gratuitas):**
  + **Render:** Muy fácil de usar y popular para Node.js.
  + **Vercel:** Aunque es más conocido por frontends, también tiene un buen soporte para funciones de servidor Node.js.
  + **Heroku:** Un clásico, aunque su capa gratuita ha cambiado.
* **Proceso General en Render (ejemplo):**
  + Sube tu proyecto a un repositorio de GitHub o GitLab.
  + Crea una cuenta en Render y conéctala a tu cuenta de GitHub/GitLab.
  + Crea un nuevo "Web Service".
  + Selecciona tu repositorio. Render detectará que es una aplicación Node.js.
  + **Comando de Build:** npm install
  + **Comando de Inicio:** node index.js (o lo que uses para iniciar tu servidor).
  + **Variables de Entorno:** Añade las mismas variables que pusiste en tu archivo .env en la sección de "Environment" de Render.
  + Haz clic en "Deploy".

Al finalizar, Render te dará una URL pública para tu backend, por ejemplo: https://mi-backend-sihur.onrender.com. **Guárdala, la necesitarás.**

**Paso 3: Preparar el Frontend (client) para Producción**

1. **Configurar la URL del API:** Tu código de React ([ConsultaCaso.jsx](code-assist-path:d:\Secretar%C3%ADaDeSeguridad\2025\05.%20Sistemas\08082025%20SIHUR%20Sistema%20de%20Informaci%C3%B3n%20para%20la%20gesti%C3%B3n%20del%20hurto%20en%20Medell%C3%ADn\client\src\components\ConsultaCaso.jsx), etc.) probablemente hace peticiones a http://localhost:3000/api/.... Debes cambiar esto para que apunte a la URL de tu backend desplegado. La mejor manera es usar variables de entorno de Vite.
   * Crea un archivo llamado .env.production en la raíz de tu carpeta client.
   * Dentro, añade la URL de tu backend:
   * VITE\_API\_URL="https://mi-backend-sihur.onrender.com"
   * En tu código, en lugar de usar http://localhost:3000, usa import.meta.env.VITE\_API\_URL.
2. **Construir la Aplicación:** Ejecuta el comando de build en tu terminal, dentro de la carpeta client:
3. npm run build

Esto creará una nueva carpeta dist con los archivos HTML, CSS y JS optimizados para producción.

**Paso 4: Desplegar el Frontend (client)**

Necesitas un servicio de hosting para sitios estáticos.

* **Opciones Populares (muy rápidas y con capas gratuitas generosas):**
  + **Netlify:** Excelente para sitios estáticos y muy fácil de usar.
  + **Vercel:** Probablemente la opción más popular para proyectos de React/Vite.
  + **GitHub Pages:** Una opción gratuita directamente desde tu repositorio.
* **Proceso General en Netlify (ejemplo):**
  + Crea una cuenta en Netlify y conéctala a tu cuenta de GitHub/GitLab.
  + Selecciona "Add new site" -> "Import an existing project".
  + Elige tu repositorio.
  + Netlify detectará que usas Vite. Revisa que la configuración sea correcta:
    - **Build Command:** npm run build o vite build
    - **Publish directory:** dist
  + Haz clic en "Deploy site".

Netlify construirá tu proyecto y lo desplegará. Te dará una URL pública, como https://mi-sihur-app.netlify.app.

**Resumen y Pasos Finales**

1. **Despliega el backend** en un servicio como Render y obtén su URL.
2. **Configura esa URL** en el código del frontend usando variables de entorno.
3. **Despliega el frontend** en un servicio como Netlify.
4. **Actualiza CORS:** Vuelve a la configuración de tu backend en Render y asegúrate de que la variable de entorno para el origin de CORS sea la URL final de tu frontend de Netlify.

Con esto, tu aplicación estará completamente en línea y funcionando. Si después quieres usar un dominio personalizado (ej. www.sihurmedellin.com), puedes comprarlo y configurarlo en los ajustes de Netlify y Render.

Principio del formulario

Final del formulario

1 context item