

«Talento Tech»

# Node JS

Clase 16



# Clase N° 16 - Despliegue a producción

## Temario:

1. Integración con Vercel
  2. Despliegue del proyecto
- 

## Objetivos de la Clase

En esta clase, los estudiantes aprenderán a desplegar una aplicación web tipo API Rest en la nube utilizando Vercel. Para ello, configurarán correctamente el entorno de despliegue, incluyendo la creación del archivo de configuración de Vercel, la vinculación del repositorio con el servicio de hosting y la gestión de variables de entorno. Al finalizar, habrán realizado el despliegue exitoso de su proyecto y probado su funcionamiento en un entorno productivo.

# Integración con Vercel



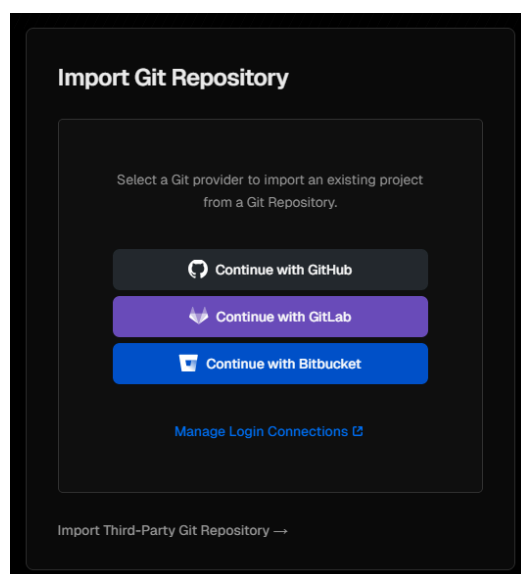
Vercel es una plataforma de hosting en la nube diseñada para facilitar el despliegue y la escalabilidad de aplicaciones web. Su propósito principal es permitir a los desarrolladores implementar proyectos de manera rápida y eficiente, sin necesidad de gestionar servidores o configuraciones complejas.

Ofrece integración nativa con repositorios de GitHub, GitLab y Bitbucket, lo que permite el despliegue continuo con cada actualización en el código. Además, soporta tecnologías modernas como Next.js, React, Vue y otros frameworks de desarrollo web. Su infraestructura optimizada garantiza tiempos de carga rápidos, distribución global de contenido mediante CDN y un entorno de trabajo ideal para aplicaciones frontend y API serverless.

Vercel ofrece un servicio de hosting gratuito mediante el plan Hobby pensado para estudiantes, aficionados y proyectos personales pequeños.

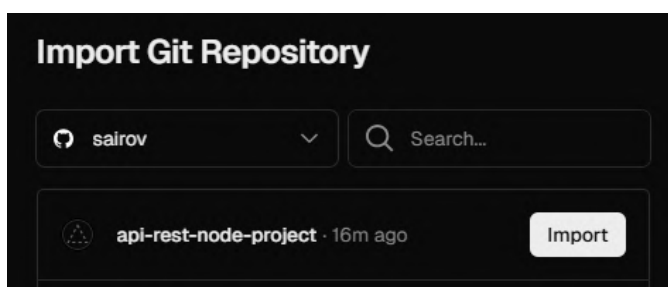


Luego de crear una cuenta en <https://vercel.com/> veremos nuestro panel de control donde en el futuro se mostrarán nuestros proyectos. En esta ocasión, damos click en Add New...



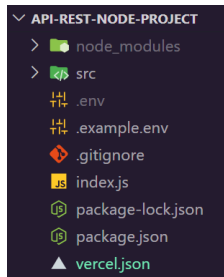
Lo siguiente es vincular Vercel con nuestro proveedor de repositorios en la nube, en este caso Github.

Nos solicitará acceder con nuestra cuenta y una vez realizado podremos ver la lista de repositorios:



Ya realizamos la integración con Vercel, el último paso es importar nuestro repositorio y proceder con el despliegue a producción.

## Despliegue del proyecto



Antes de importar el proyecto en vercel, vamos a crear un archivo de configuración llamado `vercel.json` en la carpeta principal:

En este archivo colocaremos la configuración base de vercel para el despliegue de nuestro código:

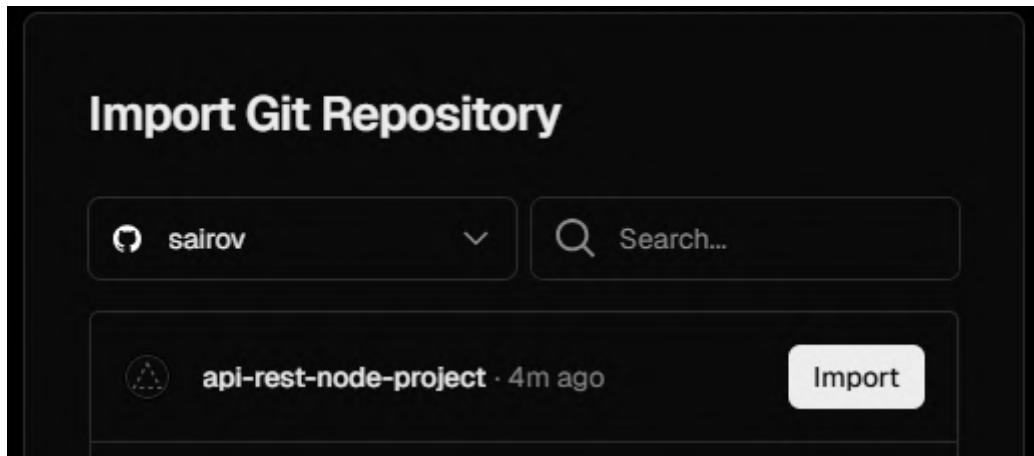
```
{
  "builds": [
    {
      "src": "./index.js",
      "use": "@vercel/node"
    }
  ],
  "routes": [
    {
      "src": "/(.*)",
      "dest": "/"
    }
  ]
}
```

En este caso le estamos indicando a Vercel el nombre del archivo principal del proyecto, el “template” de @vercel que debe utilizar (node) y las rutas de origen y destino del código desplegado en el servidor web que Vercel pone a disposición.

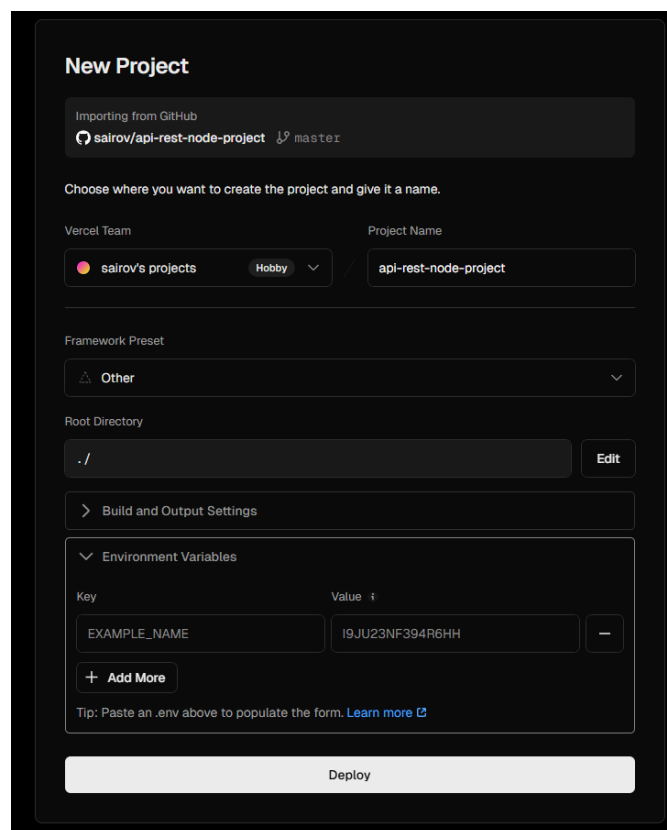
Una vez creado el archivo `vercel.json` establecido la configuración base, es momento de registrar estos cambios en GIT mediante un commit y actualizar el repositorio remoto. Para eso corremos los comandos:

- `git add .`
- `git commit -m "[CONFIG] configuración para el despliegue"`
- `git push origin master`

Ahora que ya tenemos nuestro archivo de configuración listo y el repositorio remoto actualizado, es momento de volver a Vercel e importar nuestro proyecto:



Luego de darle al botón de **Import** es importante completar las variables de entorno de nuestro proyecto:



**New Project**

Importing from GitHub  
saurov/api-rest-node-project master

Choose where you want to create the project and give it a name.

Vercel Team: saurov's projects Hobby  
Project Name: api-rest-node-project

Framework Preset: Other

Root Directory: ./ Edit

> Build and Output Settings

Environment Variables

Key	Value
EXAMPLE_NAME	I9JU23NF394R6HH

+ Add More

Tip: Paste an .env above to populate the form. [Learn more](#)

Deploy

Para eso iremos agregando una por una con los valores reales de nuestras variables:



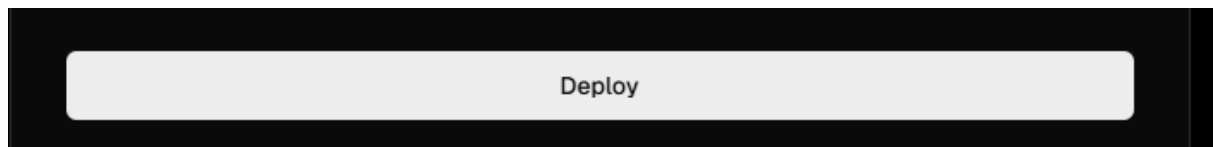
Environment Variables

Key	Value	
FIREBASE_API_KEY	KAJsneuYe_prX8Str-aeSDjHDJS5aleS9012	—
FIREBASE_AUTH_DOMAIN	your-project-name-data.firebaseio.com	—
FIREBASE_STORAGE_BUCKET	your-project-name-data.firebaseio.com	—
FIREBASE_APP_ID	9:12345678907:web:h5x76swsajh3476sd	—
JWT_SECRET_KEY	your_jwt_secret_key	—

+ Add More

Tip: Paste an .env above to populate the form. [Learn more](#)

Finalmente le damos al botón de deploy:



El proyecto comenzará a desplegarse:

### Deployment

Deployment started 7s ago...

>

Build Logs Installing dependencies ...

6s

>

Deployment Summary

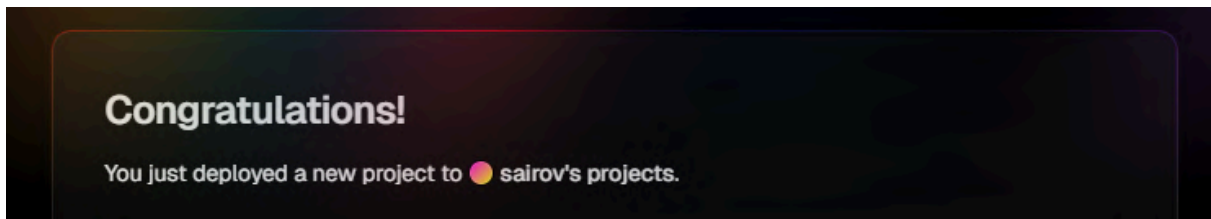
>

Assigning Custom Domains

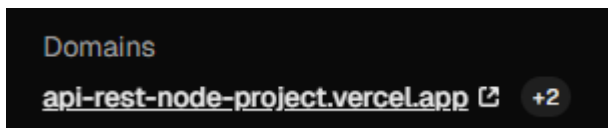
[CONFIG] configuración para el despliegue - 9337ab

Cancel Deployment

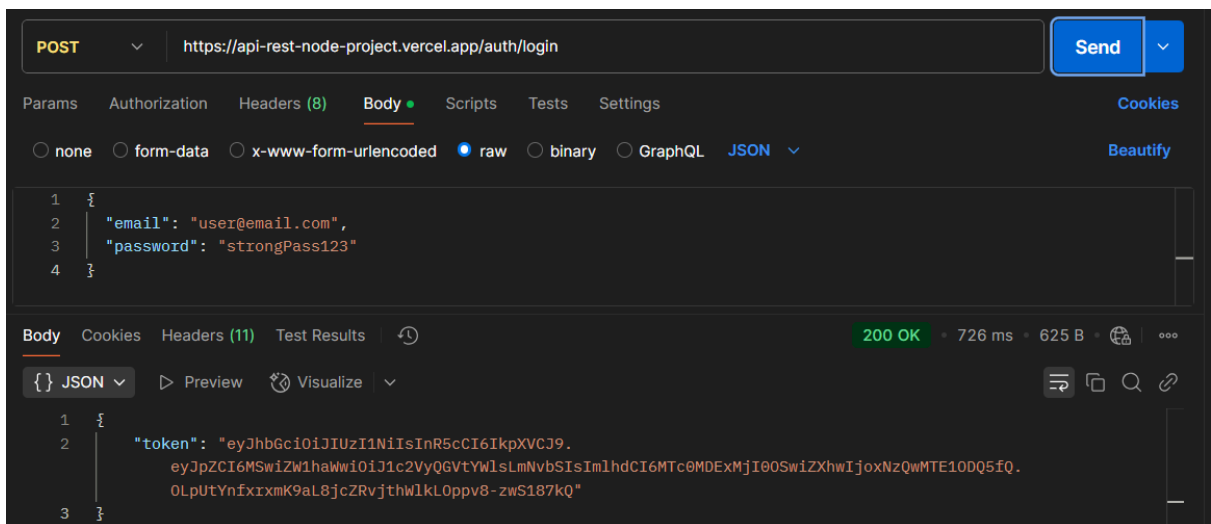
Y una vez finalizado veremos el siguiente mensaje:



Ahora desde el panel de control del proyecto podremos obtener la ruta para probarlo:



Con este domain nos dirigimos a POSTMAN:



podemos solicitar el Token para acceder a nuestra API y luego utilizarlo para consultar los productos:

***¡Felicitaciones! Tu proyecto ya se encuentra productivo. 🚀***

Ahora cualquier persona con credenciales válidas podrá autenticarse y acceder a los recursos de tu API.

Si bien este ha sido un camino extenso, aún no termina. Siempre hay nuevas herramientas y mejores prácticas que pueden ayudarte a perfeccionar tus habilidades en el desarrollo de APIs con Node.js y Express. Para continuar con tu aprendizaje, te recomendamos explorar los siguientes temas complementarios:

- **Documentación de API Rest:** Es fundamental que una API bien diseñada cuente con documentación clara y accesible. Puedes explorar herramientas como Swagger, Postman o Redocly, que permiten generar documentación interactiva y facilitar el consumo de tu API por parte de otros desarrolladores.
- **Testing:** Realizar pruebas en tu código te permitirá detectar errores antes de que lleguen a producción y asegurar el correcto funcionamiento de tu API. Aprende sobre pruebas unitarias, de integración y funcionales con herramientas como Jest, Mocha, Chai y Supertest.
- **TypeScript:** Al ser un superset de JavaScript, TypeScript agrega tipado estático a tu código, lo que facilita el mantenimiento y reduce errores. Es una excelente opción para proyectos escalables y robustos en Node.js.
- **WebSockets:** Si necesitas comunicación en tiempo real en tu aplicación, WebSockets es la solución ideal. Conoce cómo implementarlos en Node.js usando la biblioteca **Socket.io** para mejorar la experiencia del usuario en aplicaciones de chat, notificaciones en vivo y más.
- **Bases de Datos Avanzadas:** Si bien en este curso trabajamos con bases de datos en la nube, profundizar en ORMs como Sequelize o Prisma, o aprender sobre bases de datos NoSQL como MongoDB y bases de datos relacionales como MySQL puede abrirte nuevas posibilidades en el desarrollo de APIs escalables.
- **Despliegue y Escalabilidad:** Llevar tu API a producción de manera eficiente es clave. Aprende sobre contenedores con Docker, orquestación con Kubernetes, servidores en la nube como AWS, Google Cloud o DigitalOcean, y estrategias de escalabilidad como balanceo de carga y cache con Redis.

---

## Materiales y Recursos Adicionales:

- Documentación Oficial de Vercel: <https://vercel.com/docs>
  - Sitio Web oficial de GIT: <https://git-scm.com/>
  - Sitio Web oficial de GITHUB: <http://github.com/>
-



## Preguntas para Reflexionar:

- ¿Por qué podría ser necesario documentar mi API Rest? ¿Qué beneficios aportaría?
- ¿Es GITHUB el único proveedor de repositorios remotos? ¿Qué diferencias tiene con otros como Gitlab o Bitbucket?
- ¿Qué otros proveedores de hosting para Node JS existen?
- ¿Qué es docker y porque puede ser de utilidad en mis proyectos?



**Buenos Aires**  
*aprende*  
Agencia de Políticas para el Futuro

**BA** Buenos  
Aires  
Ciudad