

# Guía Completa: Backend Express.js y MongoDB Atlas en Vercel



Este proceso se divide en la **Configuración de la Base de Datos (MongoDB Atlas)** y el **Despliegue del Backend (Express.js en Vercel)**.

## I. Configuración de la Base de Datos (MongoDB Atlas)

Aquí configurarás tu clúster de MongoDB Atlas y obtendrás la URL de conexión.

### 1. Creación del Clúster

1. Accede a la web de **MongoDB** y ve a "Get Started" para **registrarse o iniciar sesión**.
2. Completa los pasos iniciales y, si no se te crea uno automáticamente, ve a "**New Project**".

**Welcome to Atlas. Let's build something great.**

Help us tailor your experience by taking a minute to answer the questions below.

**GETTING TO KNOW YOU**

**What is your primary goal?**

Learn MongoDB

**How long have you been developing software with MongoDB?**

More than a year

**GETTING TO KNOW YOUR PROJECT**

**What programming language are you primarily building on MongoDB with?**

JavaScript / Node.js

**What type(s) of data will your project use?**

You can choose as many as you want

Customer... X Catalog... X Sales / t... X

**Will your application include any of the following architectural models?**

You can choose as many as you want

Not sure... X

[Skip personalization](#)

**Finish**

3. Dentro del proyecto, haz clic en "**Create**" o "**Build a Database**".
4. En "**Deploy your cluster**", selecciona la opción "**M0 FREE**" (la versión gratuita).

5. En "Cloud Provider & Region", selecciona una región cercana, por ejemplo, "**Sao Paulo**" (o la que mejor te convenga).
6. Acepta y haz clic en "**Create Deployment**".

## 2. Configuración de Seguridad y Conexión

Una vez que el clúster esté listo:

1. **Crea el Usuario de la Base de Datos:**
  - En el panel de la izquierda, en "**Security**", ve a "**Database Access**".
  - Haz clic en "**Add New Database User**" y crea un **nombre de usuario** y una **contraseña** seguros. **Guárdalos**; los necesitarás para la URL de conexión.
2. **Permite el Acceso desde Vercel (y Cualquier Lugar):**
  - En "**Security**", ve a "**Network Access**".
  - Haz clic en "**Add IP Address**" y selecciona "**Allow Access from Anywhere**". Esto establece el rango IP a `0.0.0.0/0`, permitiendo las conexiones de Vercel y tu máquina de desarrollo.
3. **Obtén el String de Conexión (URI):**
  - En la vista general del clúster ("**Database Deployments**"), haz clic en "**Connect**".
  - Elige la opción "**Connect your application**" (controladores/drivers).
  - **Copia la cadena de conexión y reemplaza manualmente** `<password>` con la contraseña que creaste para el usuario de la base de datos.
    - **Consejo:** Si quieras apuntar a una base de datos específica, edita la URL aquí:  
`mongodb+srv://...mongodb.net/<NOMBRE-BASE-DE-DATOS>?...`

## 3. Crear Registros Iniciales (Opcional)

Para crear colecciones y documentos de prueba:

- En tu clúster, ve a "**Browse Collections**".
- Si no tienes bases de datos o colecciones, haz clic en "**Add My Own Data**" para crear una nueva.
- Dentro de una colección, usa "**Insert Document**" para crear registros de prueba.

---

## II. Despliegue del Backend Express.js en Vercel

Vercel requiere una estructura de archivos específica para funcionar como una API *serverless*.

### 1. Preparación del Código

1. **Mueve el archivo principal:** En la raíz de tu proyecto, crea una carpeta llamada `api`. Mueve tu archivo de inicio (ej. `index.js` o `server.js`) dentro de esta carpeta. Acepta cualquier corrección de rutas de importación.

2. **Modifica package.json:** Ajusta las rutas para apuntar a la nueva ubicación del archivo de inicio.

3. JSON

```
{  
  "main": "./api/index.js", // O ./api/server.js, según el nombre de tu archivo  
  "scripts": {  
    "start": "node ./api/index.js",  
    // ... otros scripts  
  }  
}
```

- 4.

- 5.

6. **Crea vercel.json:** Este archivo le indica a Vercel que todas las peticiones deben ser manejadas por el archivo dentro de la carpeta api. Crea este archivo en la **raíz de tu proyecto** y añade:

7. JSON

```
{  
  "version": 2,  
  "rewrites": [  
    {  
      "source": "/(.*)",  
      "destination": "/api/index.js" // Ajusta a tu nombre de archivo si es diferente  
    }  
  ]  
}
```

- 8.

- 9.

## 2. Despliegue a Tráves de Vercel ☁

1. **Sube a Git:** Asegúrate de que todos los cambios estén en un repositorio de **GitHub** (o GitLab/Bitbucket).
2. **Importa en Vercel:**
  - Inicia sesión en **Vercel** y haz clic en "**Add New**" > "**Project**".
  - **Importa el repositorio** que contiene tu backend.
3. **Configura Variables de Entorno (¡Clave!):**
  - Durante el proceso de configuración del proyecto en Vercel, busca la sección "**Environment Variables**".
  - Añade las variables que usa tu proyecto de Express, incluyendo:
    - **Name:** MONGO\_URI
    - **Value:** Pega aquí el string de conexión completo que obtuviste de MongoDB Atlas.
    - Agrega también SECRET y cualquier otra variable que uses (como PORT, aunque Vercel lo maneja).

4. **Despliega:** Haz clic en "**Deploy**". Vercel construirá y subirá tu backend.

### 3. Prueba y Continúa

- Una vez que el despliegue esté completo, usa el enlace que Vercel te proporciona en "**Domains**" para probar tu API.
- **Prueba Local vs. Desplegado:** Es crucial probar tu frontend *local* con tu API *desplegada* antes de subir el frontend.

---

## III. Despliegue del Frontend

Si tienes un frontend (React, Vue, etc.) separado:

1. **Prepara y prueba:** Asegúrate de que el frontend ya apunte a la **URL desplegada de tu backend en Vercel** (no a `localhost`).
2. **Sube a Vercel:** Importa el repositorio de tu frontend desde GitHub a Vercel.
3. **Configuraciones:** Vercel suele detectar automáticamente la configuración de frameworks como Next.js o Create React App. Asegúrate de que la configuración, como la **carpeta raíz** y el **script de instalación** (`npm install`), sea correcta.
4. **Despliega** y accede al enlace de "**Domains**" para probar la aplicación completa.

