Contenido

[**PASO A PASO PARA INSTALAR EL PROYECTO:** 1](#_Toc207967093)

[**PASO A PASO PARA SUBIR EL PROYECTO EN EL GITHUB:** 3](#_Toc207967094)

[1. Subir tu proyecto a GitHub (si aún no lo haces) 3](#_Toc207967095)

[1.1 Si aún no has creado el repositorio, haz esto: 6](#_Toc207967096)

[En mongo atlas 6](#_Toc207967097)

[**PASO A PASO PARA DESPLEGAR EN RENDER** 9](#_Toc207967098)

[Render 10](#_Toc207967099)

[1. Crear una cuenta en Render 10](#_Toc207967100)

[2. Desplegar 10](#_Toc207967101)

[**🔹 Paso 1: Selecciona qué repositorios quieres que Render vea** 12](#_Toc207967102)

[**🔹 Paso 2: Verifica los permisos** 13](#_Toc207967103)

[**🔹 Paso 3: Haz clic en el botón verde que dice:** 13](#_Toc207967104)

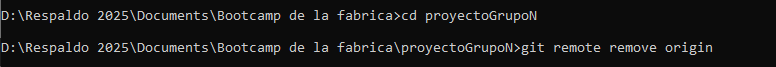
[**🔹 Paso 4: Espera a que Render te regrese** 13](#_Toc207967105)

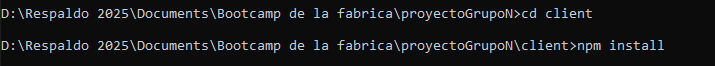
[Resultado esperado: 16](#_Toc207967106)

**PASO A PASO PARA INSTALAR EL PROYECTO:**

En el caso de haber clonado el código en github







Instalación de paquetes en modo local en símbolo del sistema o en terminal de visual studio code:

En el lado del servidor: PS D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN\server>

Ejecutar los siguientes comandos:

1. npm init

Nota: Test comand, git repository y keywords opcional

1. npm i dotenv
2. npm i body-parser
3. npm i express
4. npm i mongoose
5. npm install --save-dev nodemon
6. npm install dotenv jsonwebtoken bcryptjs
7. npm i cors
8. npm run dev
9. npm install multer

En el lado del cliente: PS D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN\client>

Ejecutar los siguientes comandos:

1. npx create-react-app client

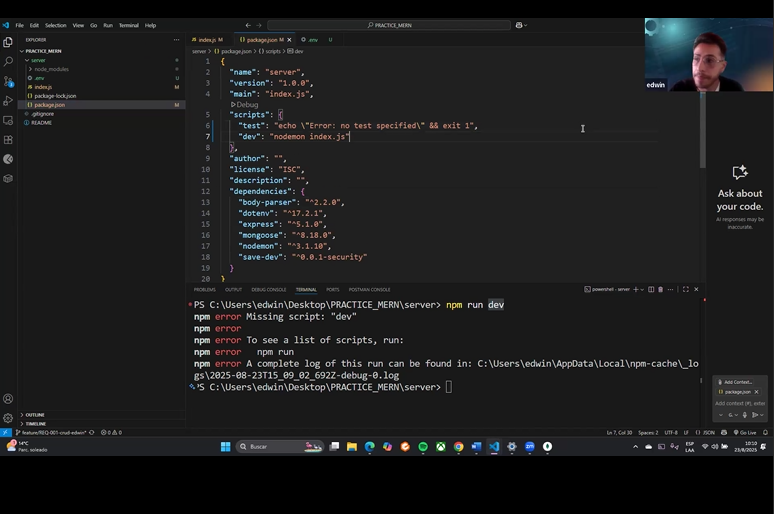
Nota: En el caso de que se haya creado manualmente la carpeta client ser como: npx create-react-app client

1. npm start
2. npm i bootstrap
3. npm i react-router-dom
4. npm i react-hot-toast
5. npm install react-bootstrap bootstrap
6. npm i font-awesome
7. npm i axios
8. npm install chart.js react-chartjs-2
9. npm run build

Para ejecutar en desarrollo local el proyecto:

1. para frontend: D:\Respaldo 2025\Documents\Bootcamp de la fabrica\modulo3\PRACTICE\_MERN\client>npm start
2. para backend: D:\Respaldo 2025\Documents\Bootcamp de la fabrica\modulo3\PRACTICE\_MERN\server> npm run dev

Nota: en el caso de error agregar en el package.json de server agregar “dev”: “nodemon index.js”



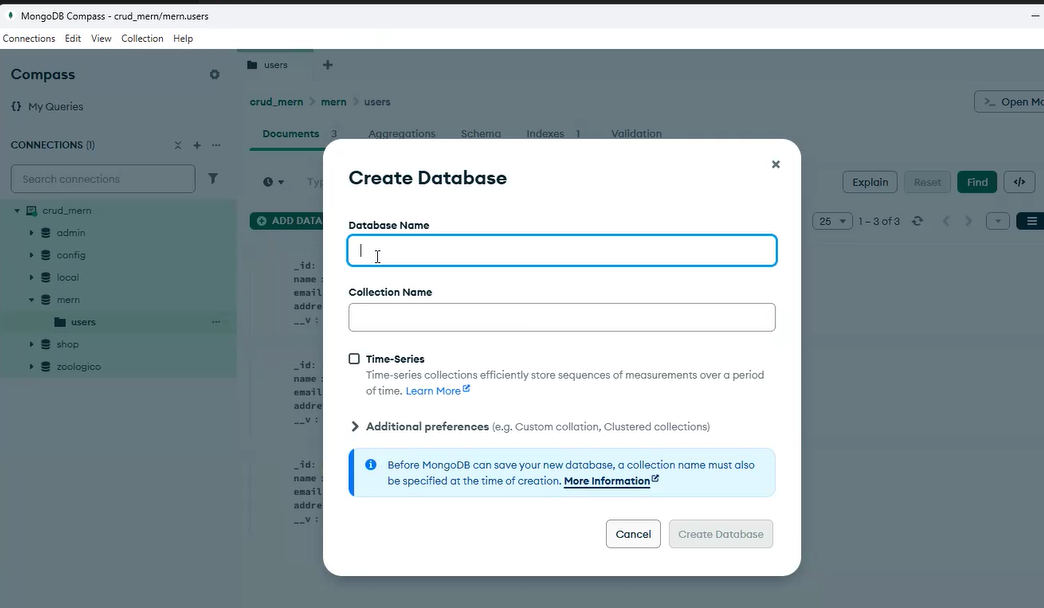
En mongo db compass hacer una nueva conexión:

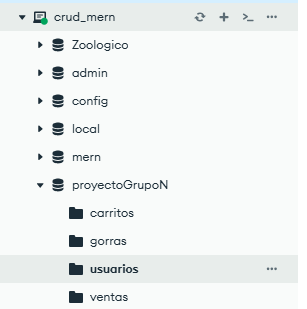
URL: mongodb://localhost:27017

Nombre de conexión: crud\_mern

Nombre de base de datos: proyectoGrupoN

Collection Name: usuarios, carritos, gorras, ventas





OPCIONAL:

Generación de JWT\_SECRET - Paso a paso

1. Crear el archivo generateSecret.js

Crea el archivo generateSecret.js en la **raíz de la carpeta**server (el mismo nivel que server.js):

**Ubicación:**

D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN\server\generateSecret.js

**Contenido del archivo:**

// generateSecret.js

const crypto = require('crypto');

const secret = crypto.randomBytes(64).toString('hex');

console.log('JWT\_SECRET=', secret);

2. Ejecutar el script

Abre tu terminal/consola y navega a la carpeta del servidor:

cd "D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN\server"

Luego ejecuta:

node generateSecret.js

3. Copiar el resultado

Verás una salida similar a esto en tu terminal:

JWT\_SECRET= 4f7d9a2c8b3e6f1a9c5d2e8b7a4f9c1d3e6f8a2b5c9d7e1f4a8b3c6d9e2f5a7b9c4d6e8f1a3b5c7d9e2f4a6b8c1d3e5f7a9b2c4d6e8f1a3b5

4. Actualizar el archivo .env

Abre tu archivo .env y reemplaza la línea del JWT\_SECRET con la clave generada:

**Archivo:** D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN\server\.env

PORT=5001

MONGO\_URI=mongodb://localhost:27017/proyectoGrupoN

JWT\_SECRET=4f7d9a2c8b3e6f1a9c5d2e8b7a4f9c1d3e6f8a2b5c9d7e1f4a8b3c6d9e2f5a7b9c4d6e8f1a3b5c7d9e2f4a6b8c1d3e5f7a9b2c4d6e8f1a3b5

JWT\_EXPIRE=30d

NODE\_ENV=development

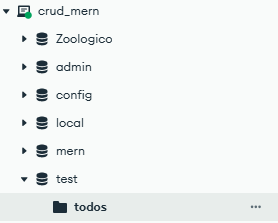
5. (Opcional) Eliminar el archivo generateSecret.js

Una vez que hayas copiado el JWT\_SECRET a tu archivo .env, puedes eliminar el archivo generateSecret.js por seguridad:

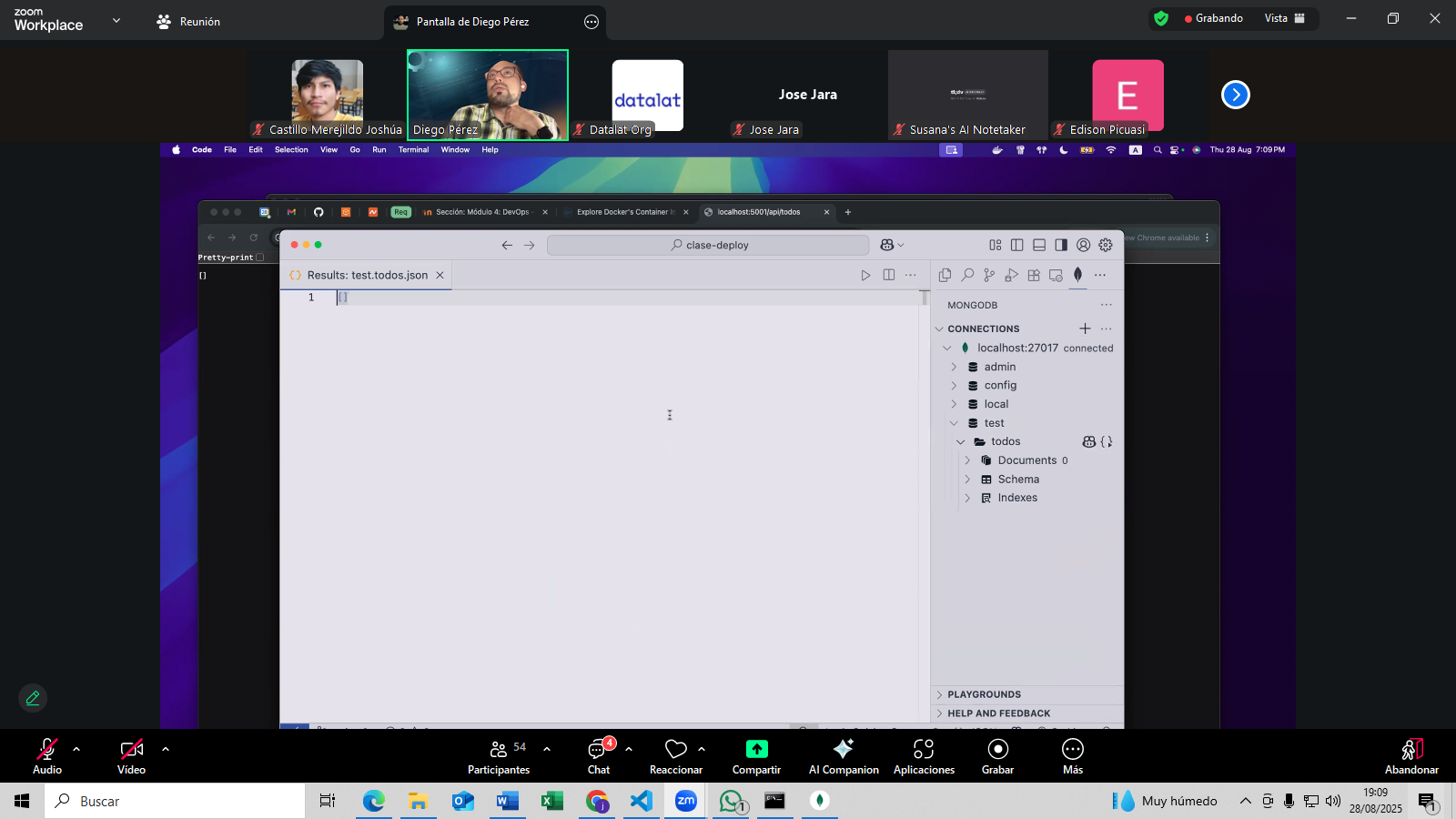
# Desde la carpeta server/

del generateSecret.js

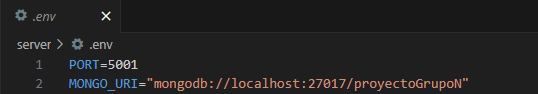
En mongoDB compass crear una nueva conexión como crud\_mern, data\_base\_name: test, collection\_name: todos



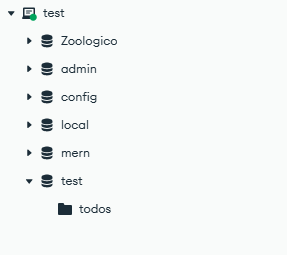
O en la extension de mongodb de visual studio code



O pueden poner:



En dicho caso se creará una nueva conexión en caso de no existir de manera automática al arrancar el programa en local:



**PASO A PASO PARA SUBIR EL PROYECTO EN EL GITHUB:**

# 1. Subir tu proyecto a GitHub (si aún no lo haces)

Desde tu computadora:

cd D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN>

git init

git add .

git commit -m "Proyecto Grupo N inicial"

en caso de que se presente estos mensajes:



Abre tu terminal (CMD o PowerShell) y escribe:

D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN> git config --global user.name "Tu Nombre"

D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN> git config --global user.email "tunombre@ejemplo.com"

e intentar nuevamente:

cd D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN> git commit -m "Proyecto Grupo N inicial"

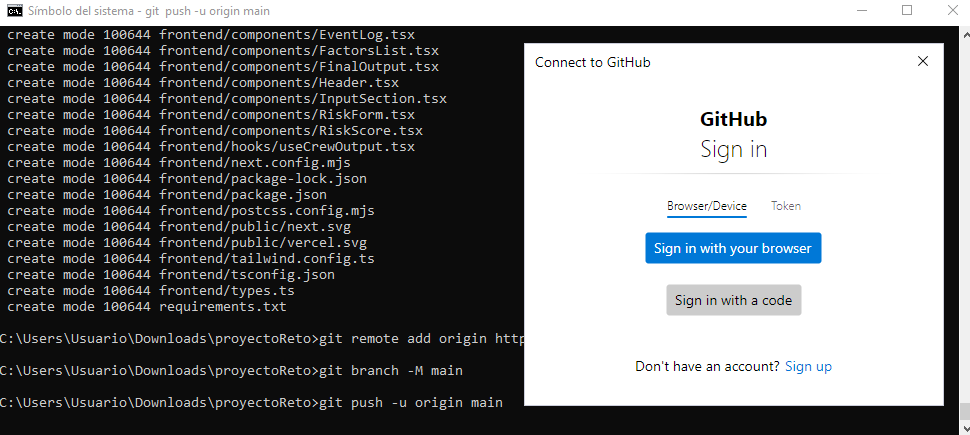
Continuar con el resto de comandos:

cd D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN>git remote add origin https://github.com/TU\_NOMBRE\_DE\_USUARIO\_DE\_GITHUB/proyecto-Grupo-N.git

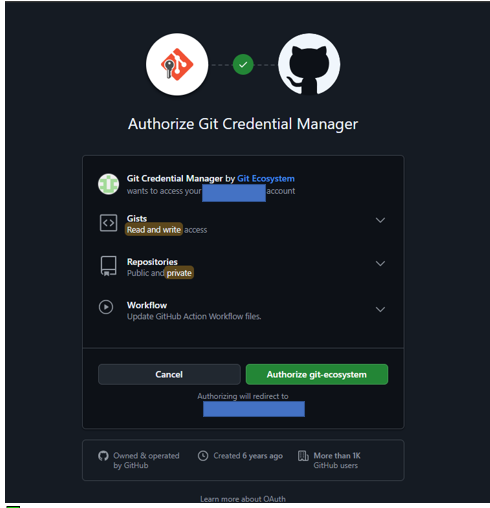
git branch -M main

git push -u origin main

**Paso 1: Clic en Sign in with your browser**



1. Dale clic a “Sign in with your browser”
2. Se abrirá tu navegador web predeterminado
3. GitHub te pedirá iniciar sesión
   * Ingresa tu usuario y contraseña
   * (Si tienes activado el 2FA, te pedirá el código de tu app o celular)
4. Cuando GitHub te pregunte:

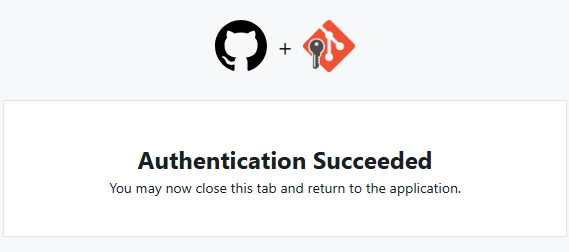


**“Authorize Git Credential Manager”**  
Haz clic en el botón verde: Authorize git-ecosystem

**Paso 2: Git te confirmará**

Volverás al terminal y verás algo como:

Successfully authenticated GitHub CLI.



**🔸 Paso 3: Git push se completa automáticamente**

Luego de la autenticación, Git debería continuar automáticamente con:

git push -u origin main

Y verás algo como:

Enumerating objects: 55, done.

Counting objects: 100% (55/55), done.

Delta compression using up to 4 threads

Compressing objects: 100% (40/40), done.

Writing objects: 100% (55/55), 123.45 KiB | 1.23 MiB/s, done.

Total 55 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/TU\_NOMBRE\_DE\_USUARIO\_EN\_GITHUB/proyecto-Grupo-N.git

\* [new branch] main -> main

Branch 'main' set up to track remote branch 'main' from 'origin'.

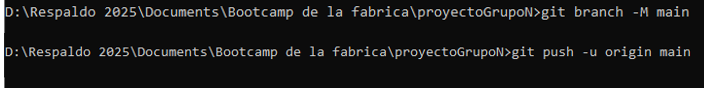
**¿Y ahora?**

¡Ya está! Tu proyecto completo está en GitHub.

Puedes confirmarlo entrando a:

<https://github.com/TU_NOMBRE_DE_USUARIO_EN_GITHUB/proyecto-Grupo-N.git>

En el caso de error:



## 1.1 Si aún no has creado el repositorio, haz esto:

1. Entra a <https://github.com/new>
2. Llena los campos:
   * **Repository name:** proyectoGrupoN
   * ✅ *Público* o *privado* (como prefieras)
   * (No marques ninguna opción de README o .gitignore)
3. Haz clic en **“Create repository”**

Luego, **vuelve a tu terminal** y ejecuta:

D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN> git remote set-url origin

D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN>https://github.com/TU\_NOMBRE\_DE\_USUARIO\_EN\_GITHUB/proyecto-Grupo-N.git

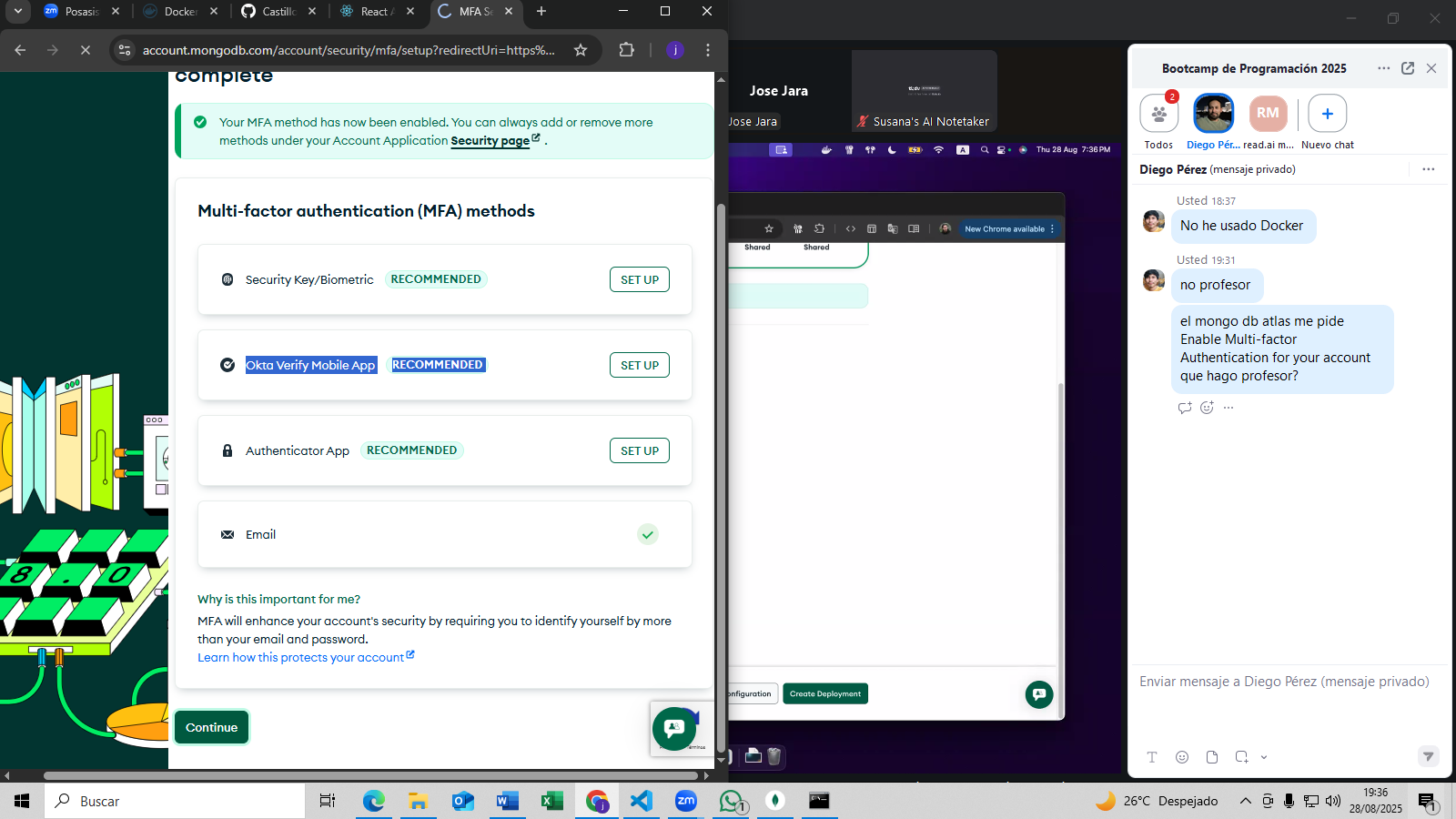
D:\Respaldo 2025\Documents\Bootcamp de la fabrica\proyectoGrupoN> git push -u origin main

# En mongo atlas

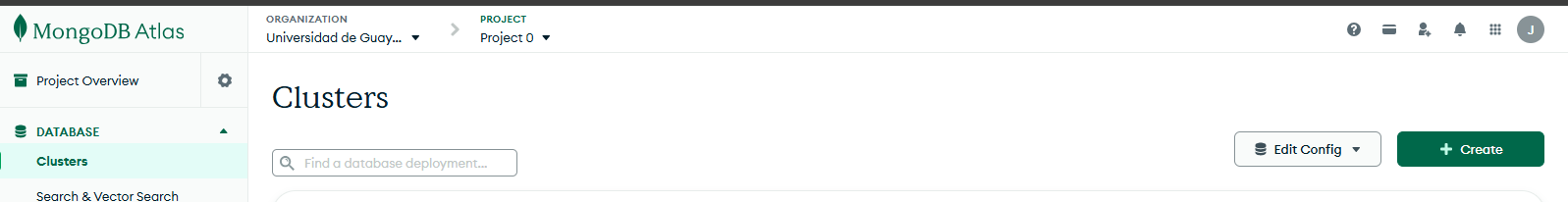
Crear una cuenta en MongoDB Atlas e iniciar sesión en Mongo atlas:

<https://account.mongodb.com/account/login>

De preferencia de email verificación.

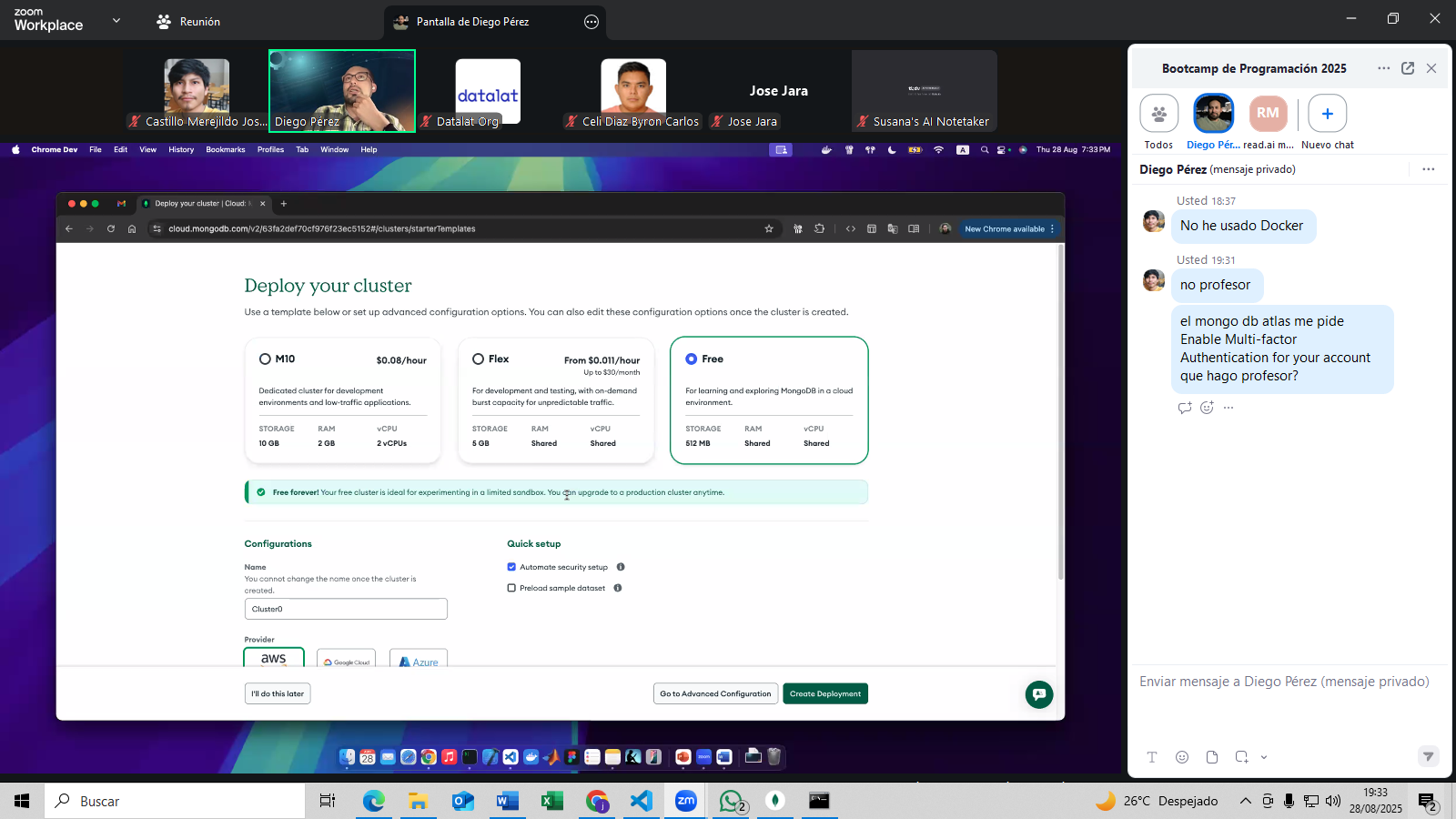


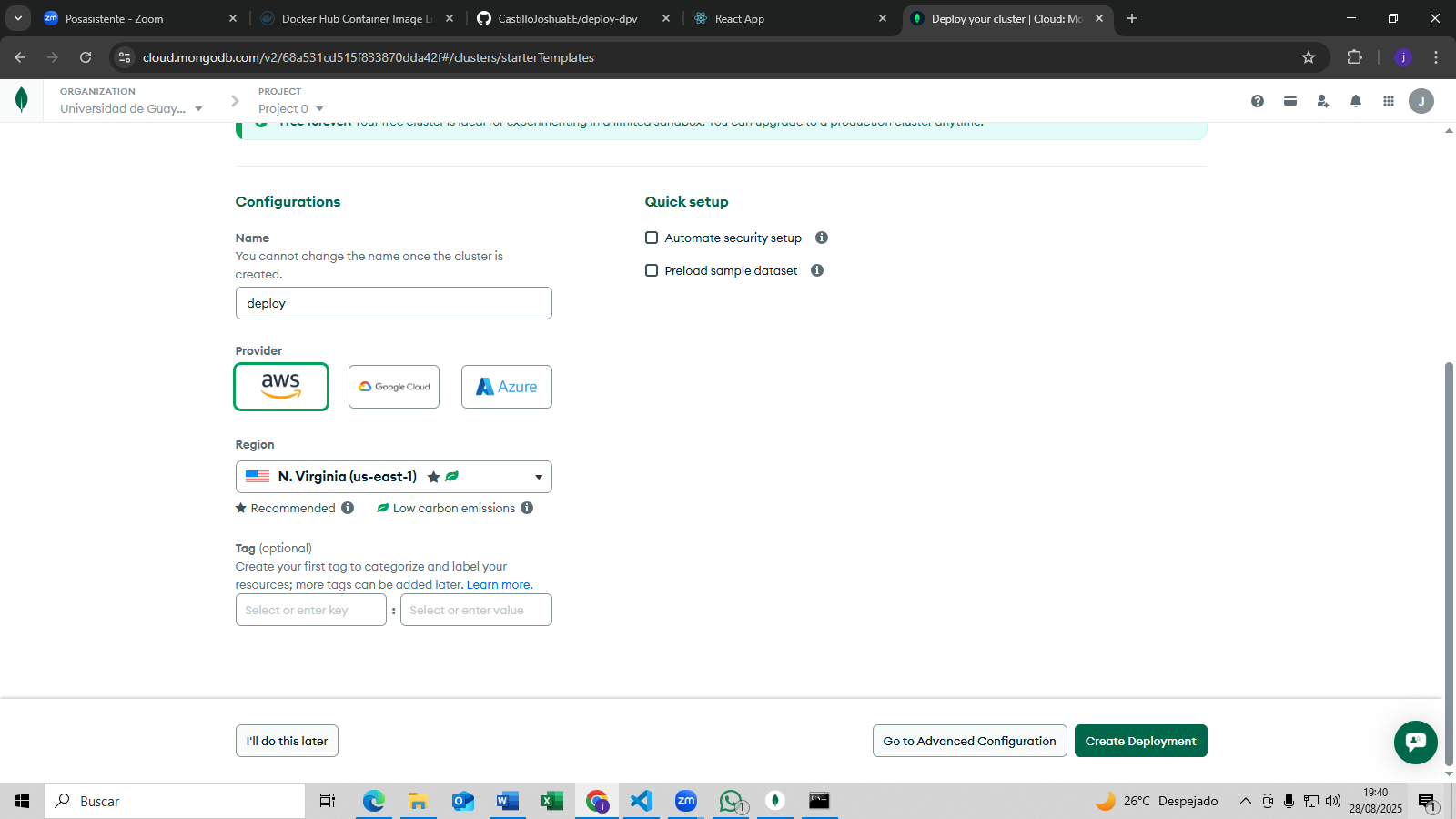
Crear un nuevo cluster



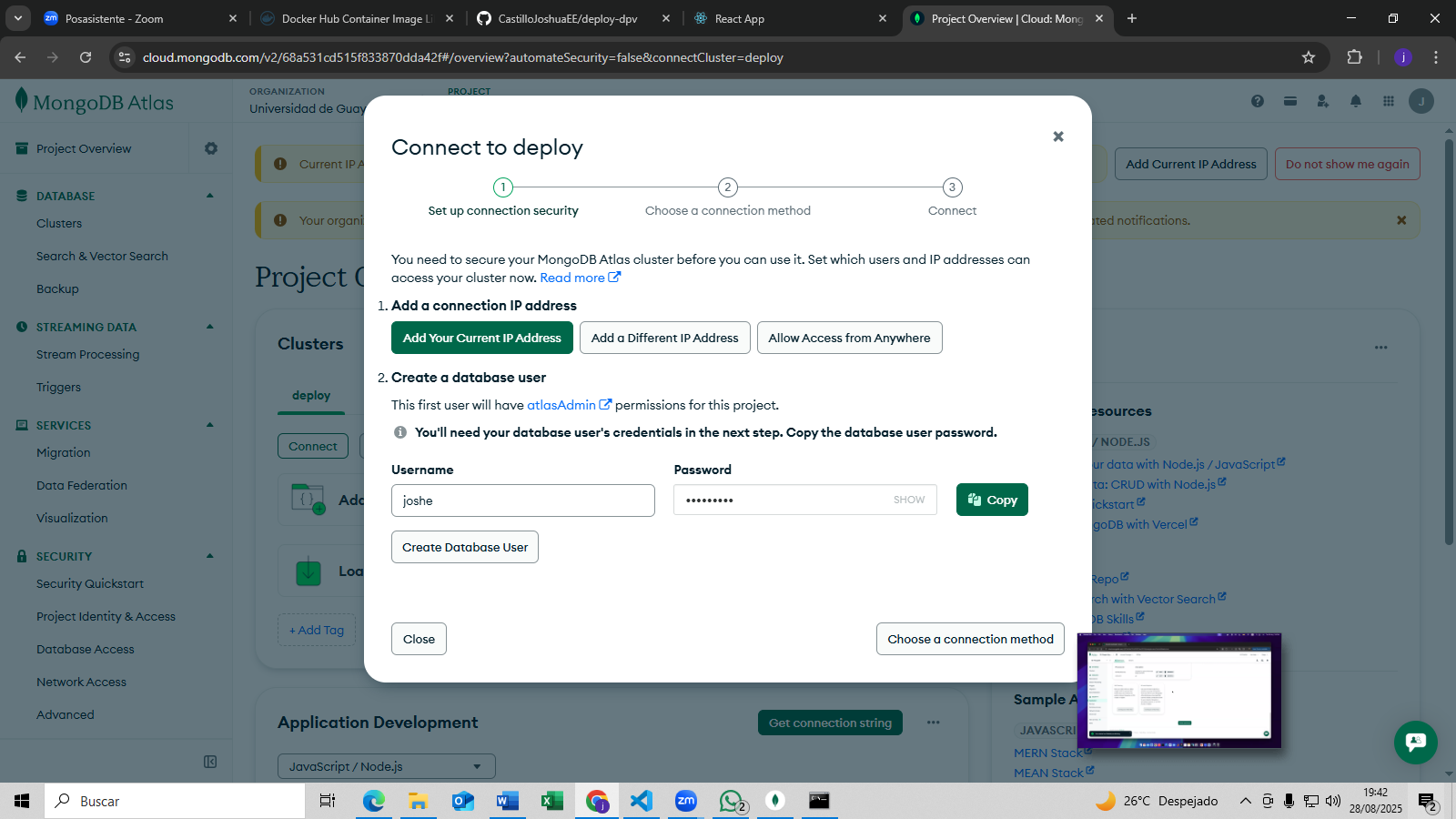
Escoger un plan: Por ser la primera vez le aparecerá el plan free.

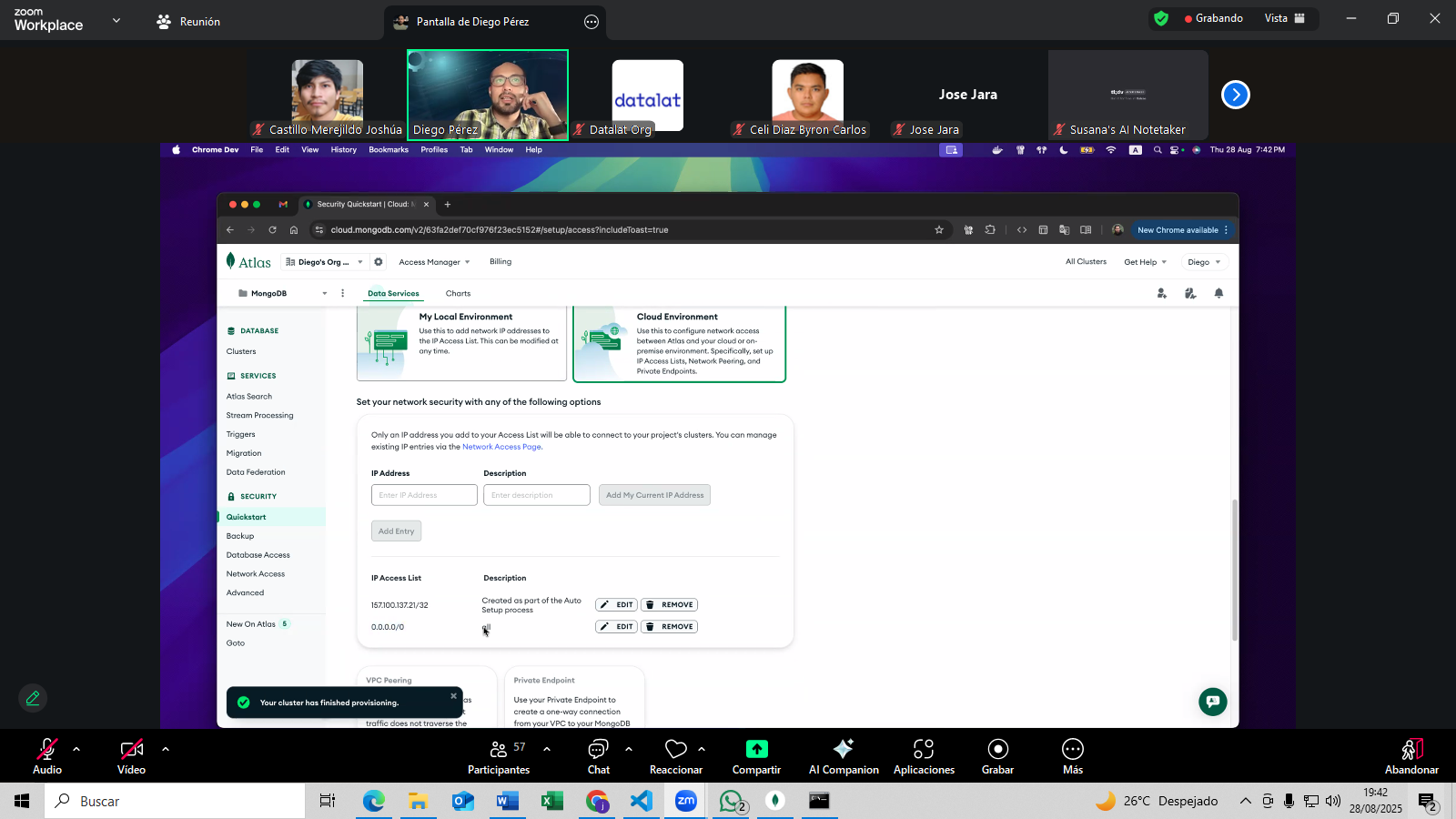
* Name: Cluster()
* Provider: AWS
* Region: N. Virginia (us-east-1)
* Tag (optional):



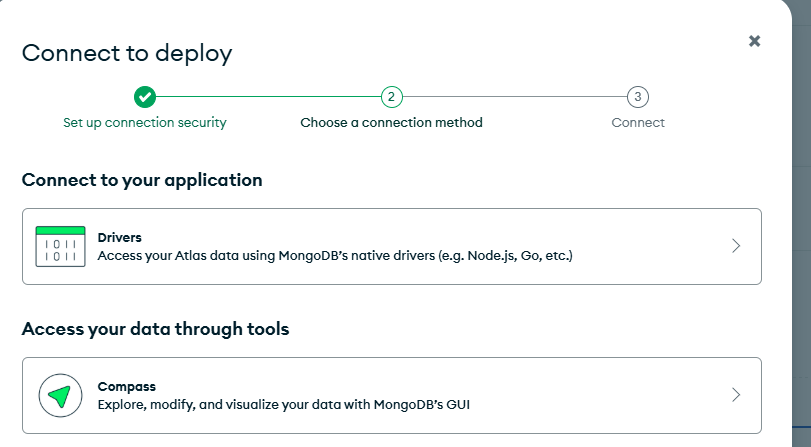


Crear un usuario de base de datos: usuario y contraseña

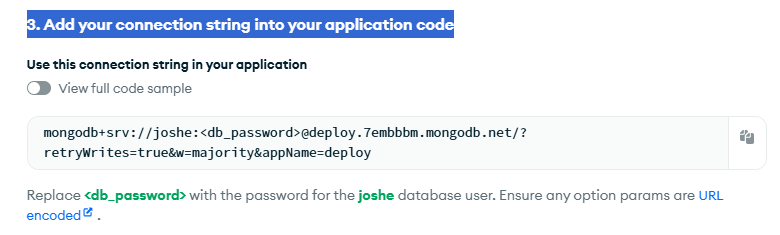


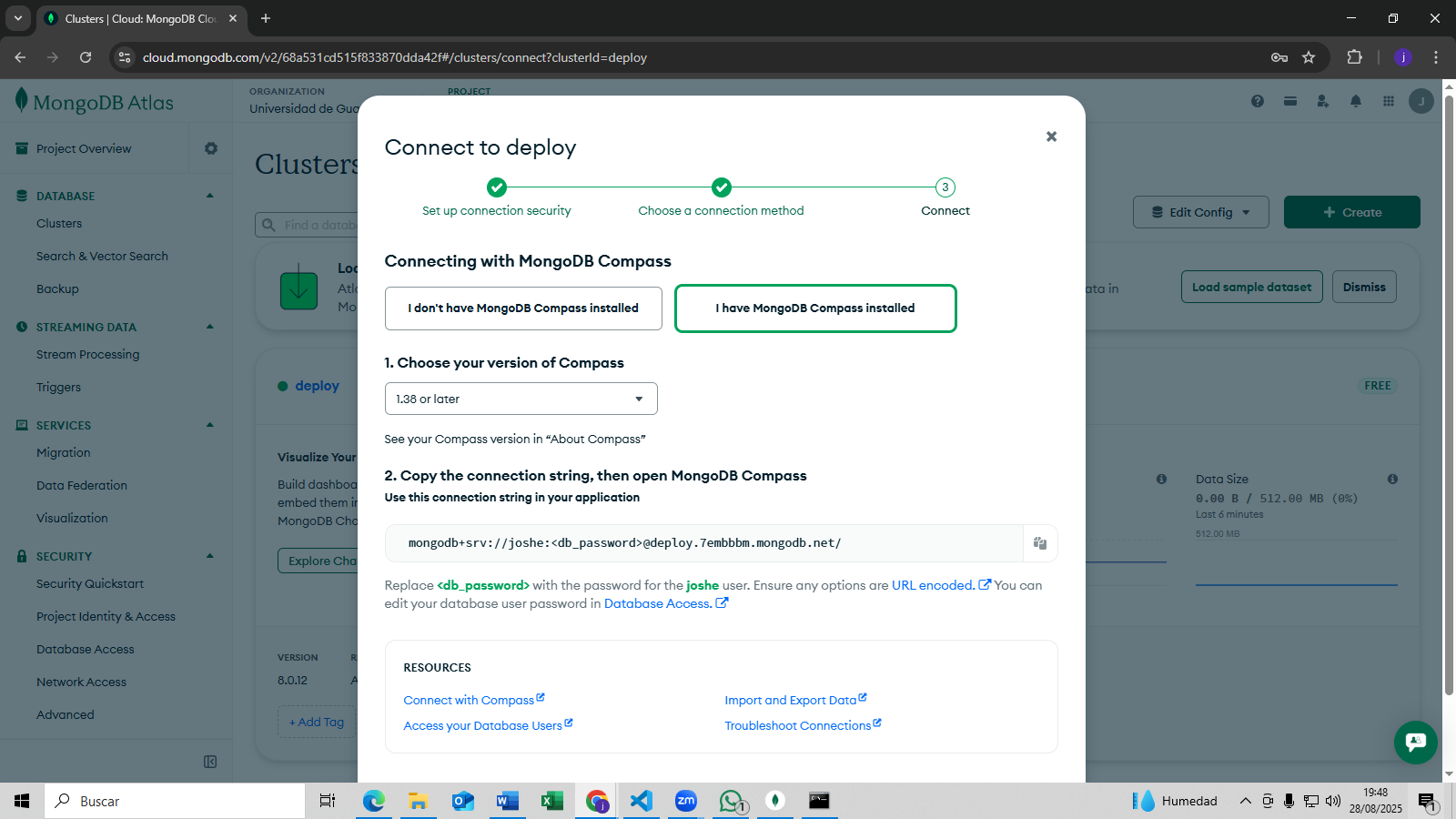
Crear un add current IP ADDRESS: una IP Address y description.

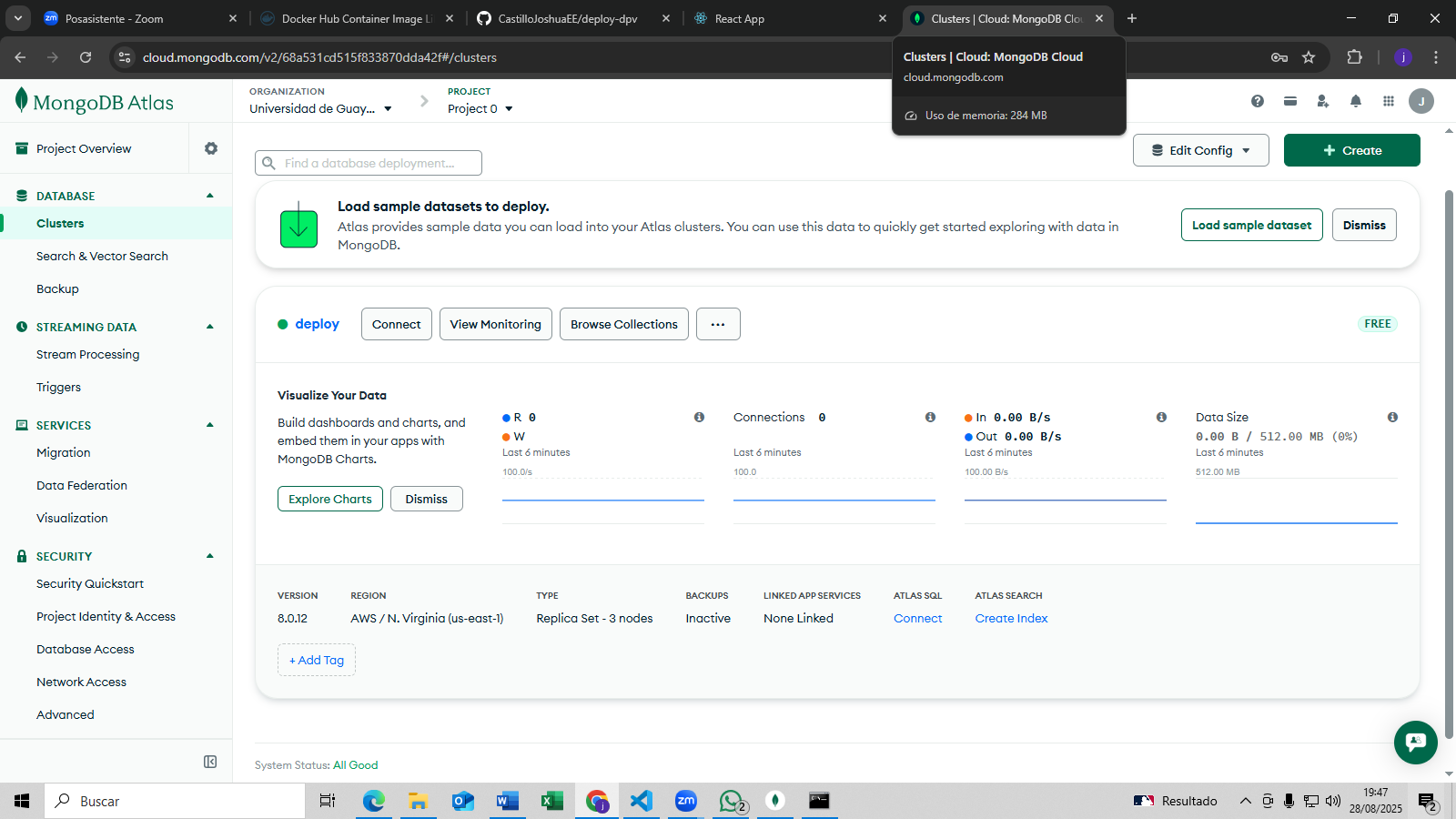
En connect to deploy, nombre escogido previamente en el cluster, hacer el connect to your application con drivers:



Tener en cuenta este link:



Si escoge compass le aparecerá esta opción: 



**PASO A PASO PARA DESPLEGAR EN RENDER**

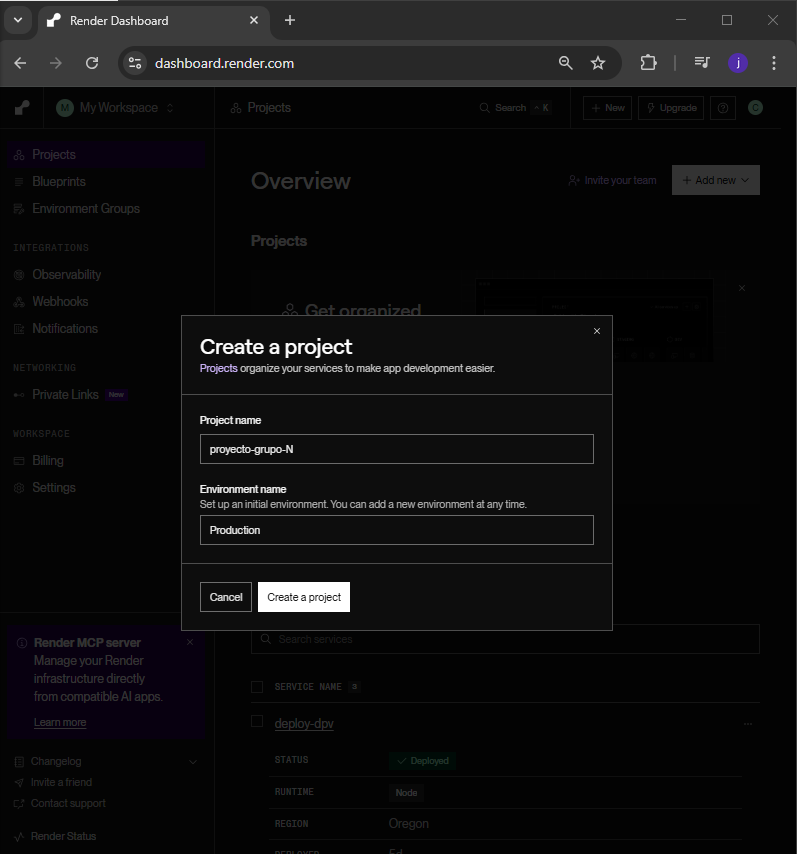
# Render

## 1. Crear una cuenta en Render

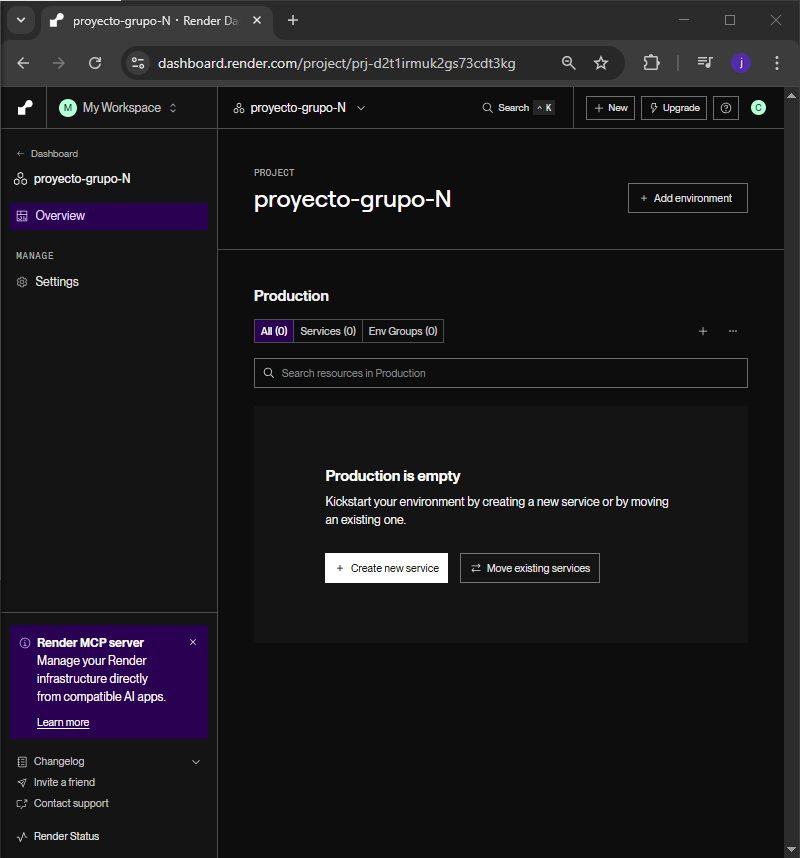
1. Ve a 👉 <https://render.com>
2. Haz clic en **Sign Up**.
3. Regístrate usando tu cuenta de GitHub.
4. Autorizar a Render.

## 2. Desplegar

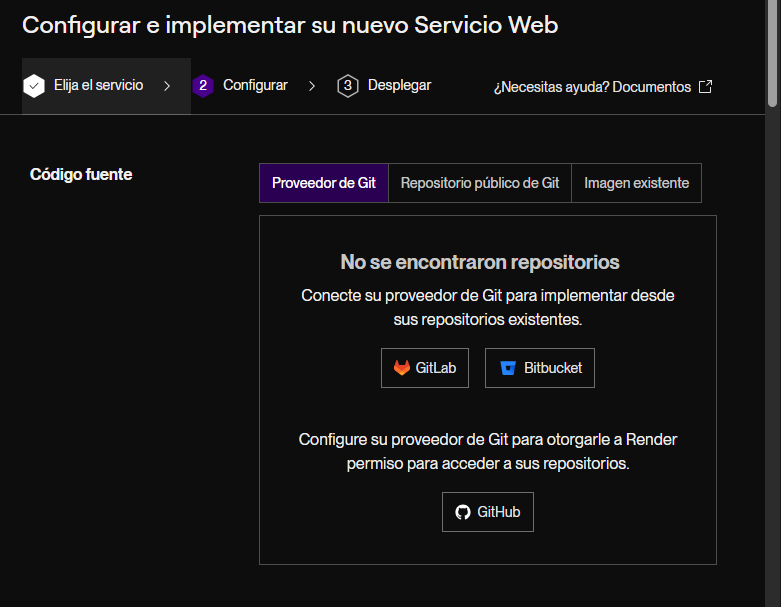
1. En Render inicia con tu cuenta de github.
2. Haz clic en create a Project



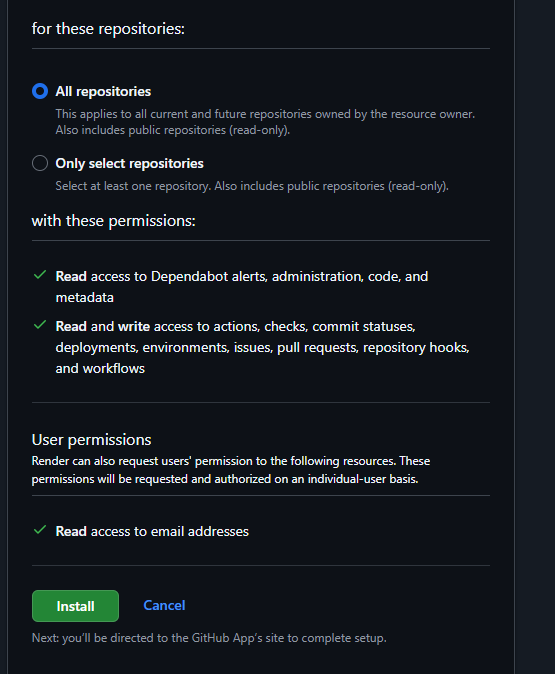
1. Haz clic en **New → Web services→ New Web Services**



1. Haz click en GitHub.



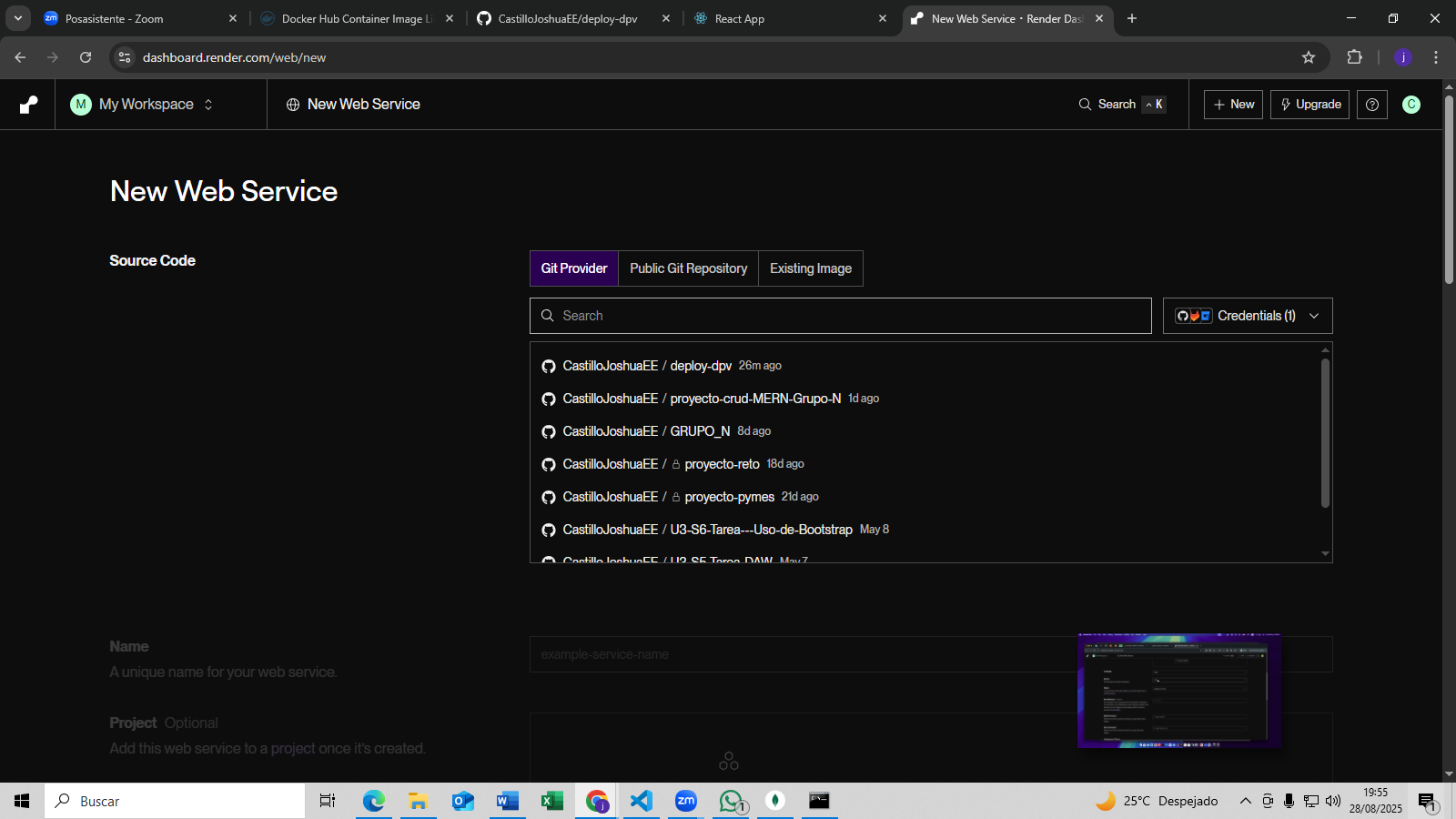
Ahora en esto:



**🔹 Paso 1: Selecciona qué repositorios quieres que Render vea**

Tienes dos opciones:

* ✅ **Opción recomendada (más fácil):**
  + Marca: All repositories



* + Esto permitirá que Render acceda automáticamente a todos tus repos públicos y privados, incluido proyecto-pymes.
* **Opción más específica (más controlada):**
  + Marca: Only select repositories
  + Luego, selecciona manualmente tu repo proyecto-pymes

**🔹 Paso 2: Verifica los permisos**

Debajo verás una lista de permisos como:

* Lectura y escritura en acciones, commits, despliegues, etc.
* Lectura de alertas de seguridad

Esto es normal y necesario para que Render pueda leer el código y desplegar tu proyecto.

**🔹 Paso 3: Haz clic en el botón verde que dice:**

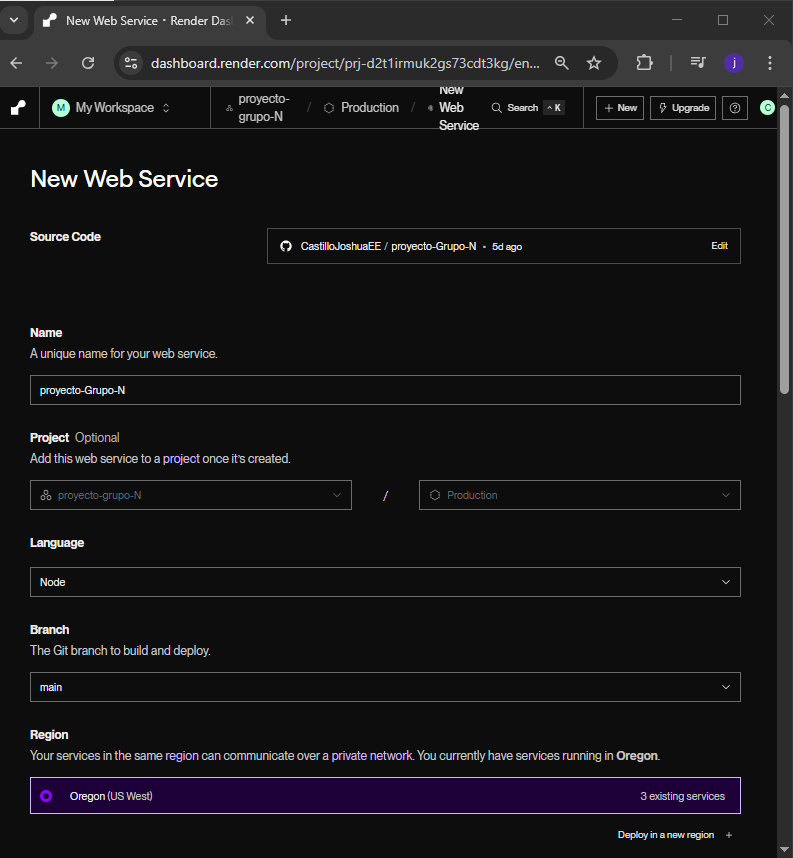
Install & Authorize o Install

(o algo como “REACT” dependiendo del idioma de tu cuenta)

**🔹 Paso 4: Espera a que Render te regrese**

Render volverá a su sitio y ahora sí verás tu repositorio proyecto-pymes disponible para seleccionar.

1. Selecciona tu repositorio proyecto-Grupo-N



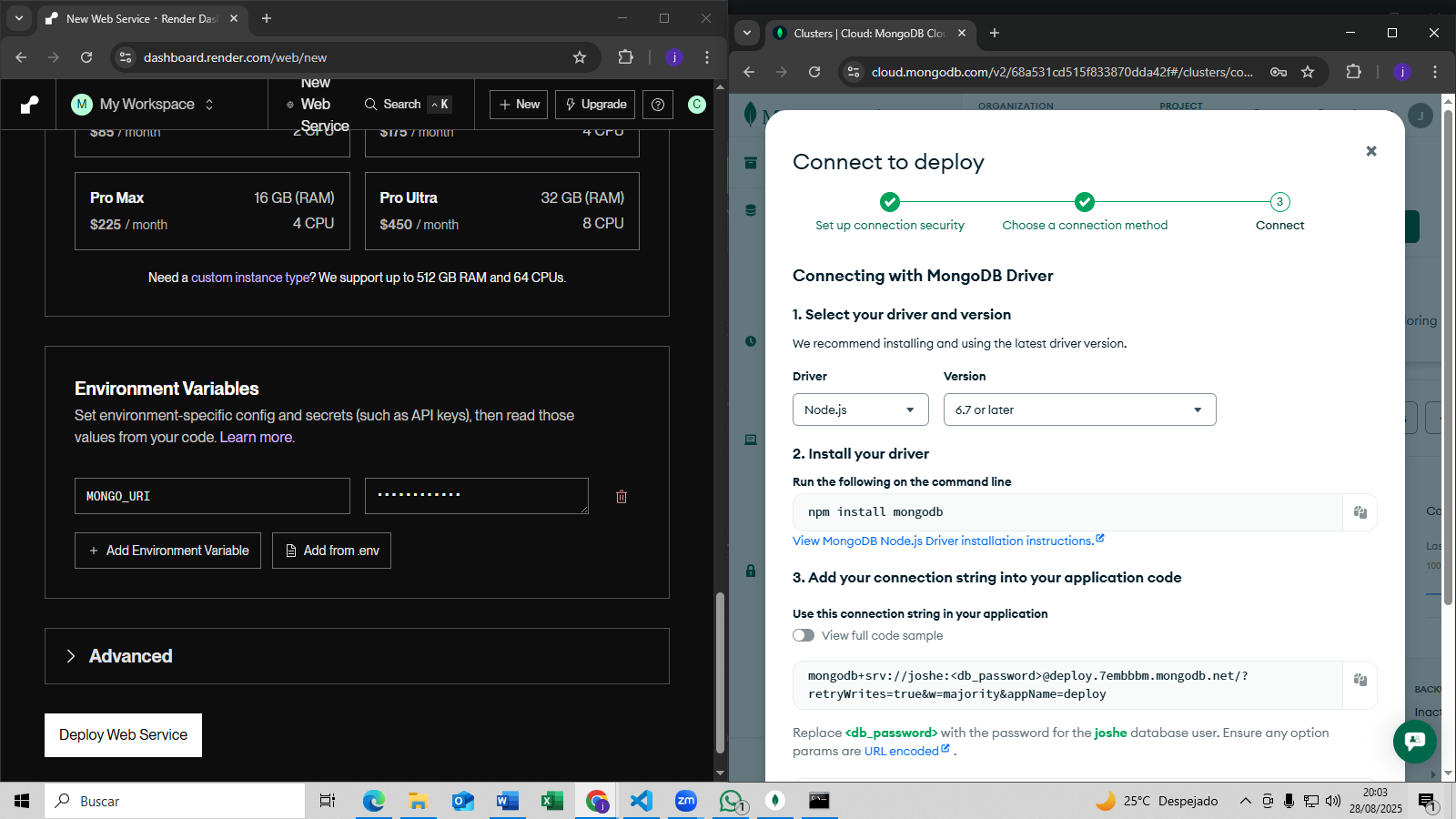
1. Configuración:

| **Campo** | **Valor** |
| --- | --- |
| **Name** | proyecto-grupo-n-unificado |
| **Language** | Node |
| **Branch** | Main |
| **Region** | Oregon (US West) |
| **Root Directory** |  |
| **Git Credentials** | Tu correo electrónico |
| **Build Command** | cd client && npm install && npm run build && cd ../server && npm install |
| **Pre-Deploy Command** |  |
| **Start Command** | cd server && node index.js |
| **Auto-Deploy** | On Commit |
| **Deploy Hook** | Dejarla por defecto |
| **Custom Domains** | Enabled |
| **PR Previews** | Off |
| **Edge caching** | Edge caching is only available for paid instances |
| **Service notifications** | Use workspace default (Only failure notifications) |
| **Preview environment notifications** | Use account default (Disabled) |
| **Health Check path** | /healthz |
| **Maintenance Mode** | Maintenance mode is only available for paid instances |
| **Instance Type** | For hobby projects: Free plan $0/month 512 (RAM) 0.1 CPU |

En “Environment variables” agrega tus variables de entorno:

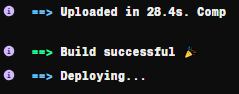
* + **PORT=5001**
  + **JWT\_SECRET=b965b924d9d7bcb2f0d417776918990ff51960a9c4a2e4b4422c900099f38b6c7376e2b083d7a9108e149688bc9d182e974b25567eab6ba892ae91b00834997d**
  + **JWT\_EXPIRE=30d**
  + **NODE\_ENV=production**
  + **REACT\_APP\_API\_URL=https://proyecto-grupo-n-unificado.onrender.com**
  + **MONGO\_URI="mongodb://localhost:27017/proyectoGrupoN"(Para local)**
  + **MONGO\_URI=tu\_link\_de\_mongo\_atlas**

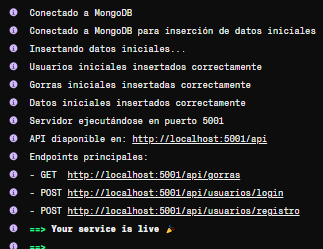
**No** pongas la contraseña con < > en la variable environment de MONGO\_URI



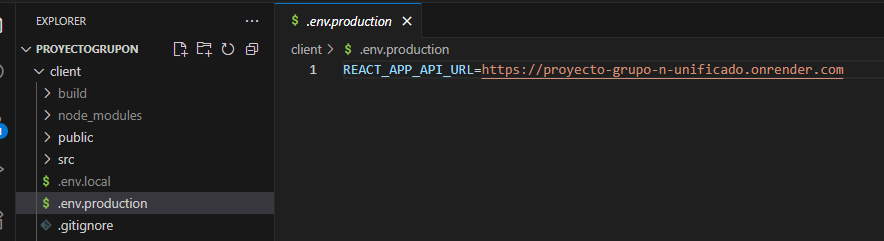
5. Clic en **Create Web Service**

# Resultado esperado:

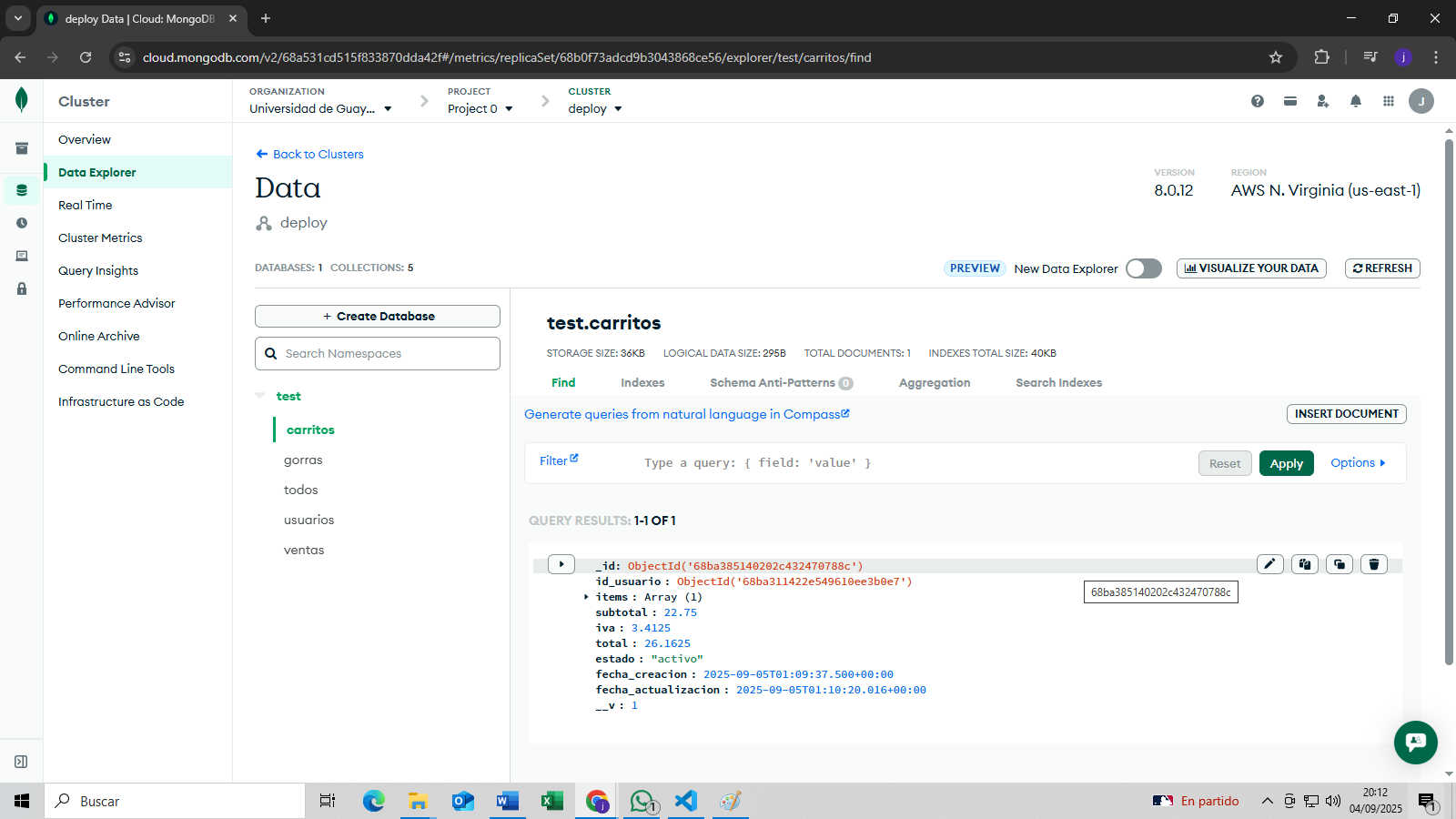




Actualizar con el link que lanza de: <https://proyecto-grupo-n-unificado.onrender.com> en .env.production:



Se puede revisar los datos que se han ingresado en mongo db Atlas:



Nota:

Render, en su plan gratuito, suspende automáticamente las aplicaciones que no reciben tráfico durante cierto tiempo. Cuando alguien vuelve a acceder, Render reactiva la aplicación, pero eso puede tardar entre 30 segundos y 1 minuto, lo que puede parecer que el sitio está caído.

✅ ¿Qué puedes hacer?

* Paciencia al primer acceso: Espera unos segundos si el enlace no responde de inmediato.
* Puedes abrir la app desde el panel de Render para "despertarla" antes de compartir el link.
* Si el presupuesto lo permite, actualiza a un plan de pago, así el servicio permanece activo 24/7 sin suspensión.