### **Introducción**

Este instructivo tiene como objetivo guiar paso a paso en el despliegue de una base de datos y su conexión con una API en Render. A través de este proceso, se utilizará Render para subir el backend y la base de datos, asegurando una correcta configuración y conexión entre ambas, para la parte frontend realizaremos la guia paso a paso del despliegue en el proveedor Firebase.

### **Objetivos**

* Desplegar una base de datos PostgreSQL en Render.
* Configurar un servicio web para la API y conectarlo con la base de datos desplegada.
* Realizar el despliegue automático en Render cada vez que se realicen cambios y commits en la rama principal del repositorio.
* Realizar el despliegue de la parte frontend.

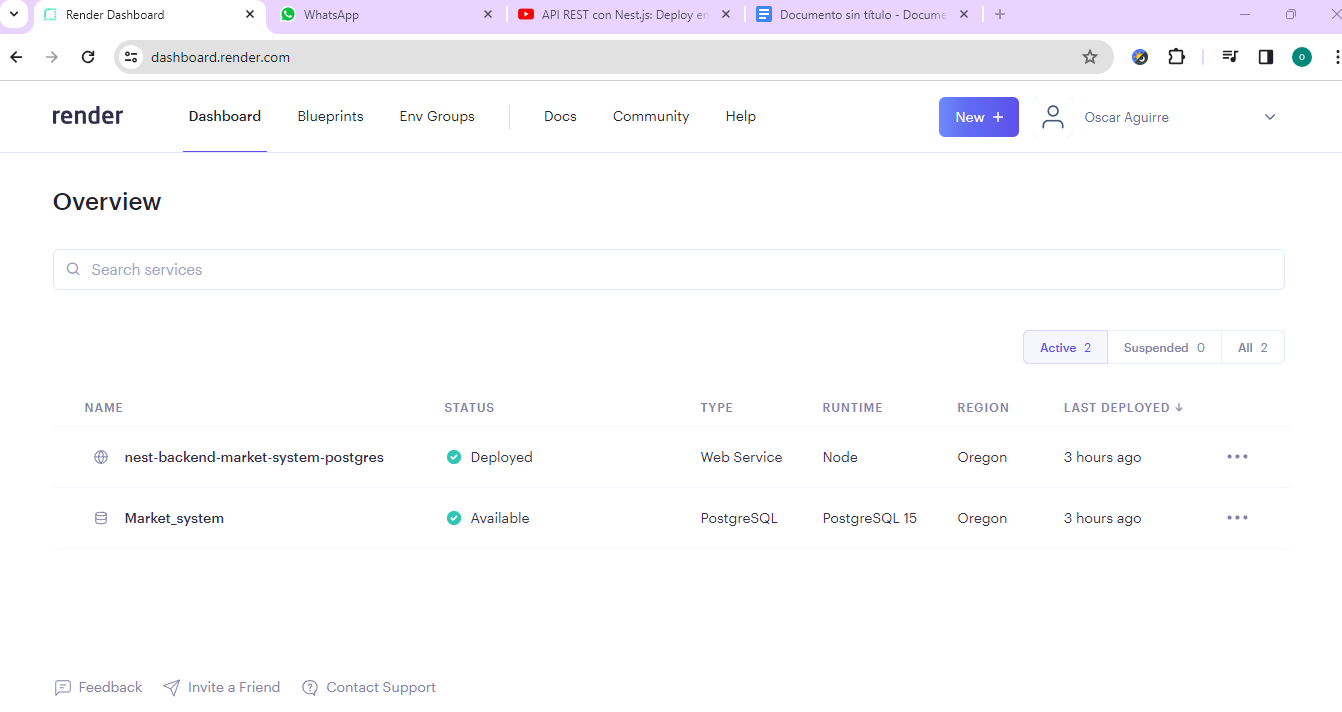
### **Tabla de Contenido**

* Creación del Servicio de Base de Datos en Render
  + Generación del servicio PostgreSQL
  + Configuración de parámetros y selección de opciones
  + Obtención de datos para la conexión con la API
* Creación del Servicio Web en Render
  + Configuración del despliegue desde un repositorio Git
  + Definición de parámetros del servicio web
  + Configuración de variables de entorno
  + Despliegue del servicio web
* Verificación y Gestión
  + Verificación del enlace generado
  + Revisión de logs y solución de errores durante la ejecución automática
* Despliegue del frontend
  + Configurando Firebase.
  + Configurando el proyecto para el despliegue.
  + Desplegando el frontend.

**1. Creación del Servicio de Base de Datos en Render**

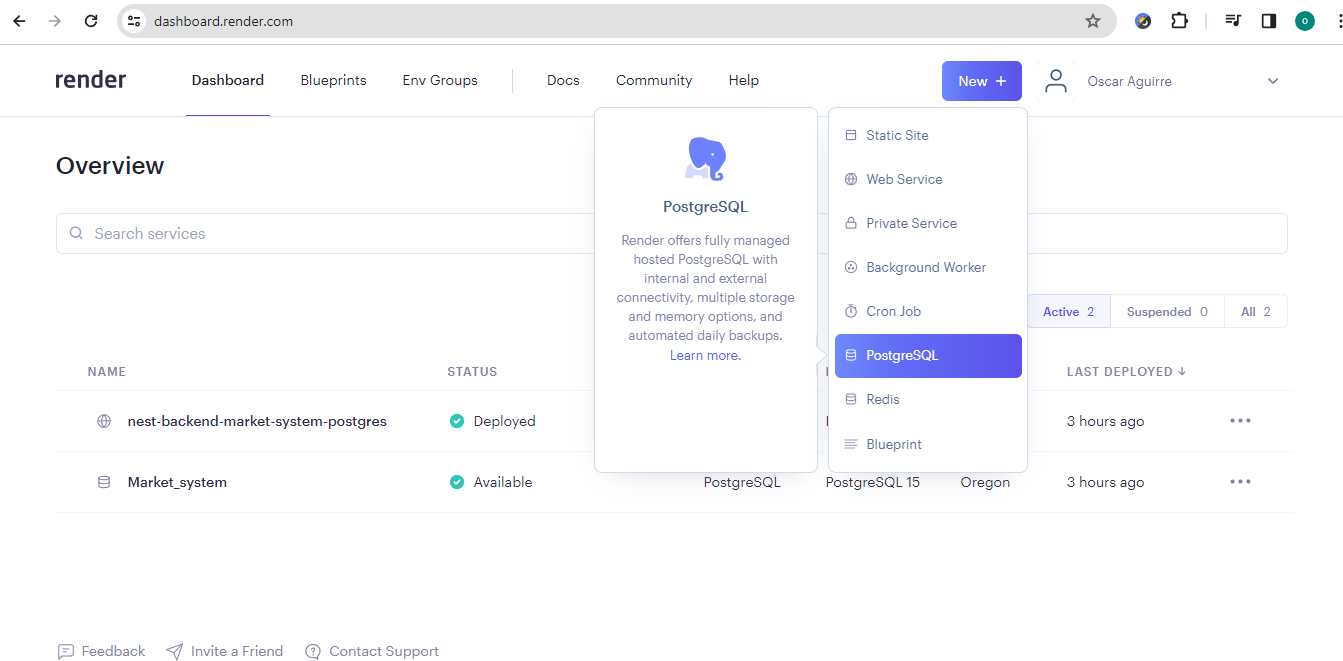
#### Paso 1: Generación del Servicio PostgreSQL

Para iniciar, accede a Render después de crear tu cuenta. Se recomienda vincular la cuenta de Render con GitHub para una integración más sencilla. Al ingresar, verás la interfaz de Render, excluyendo las áreas marcadas en amarillo, que corresponden a servicios previamente creados.



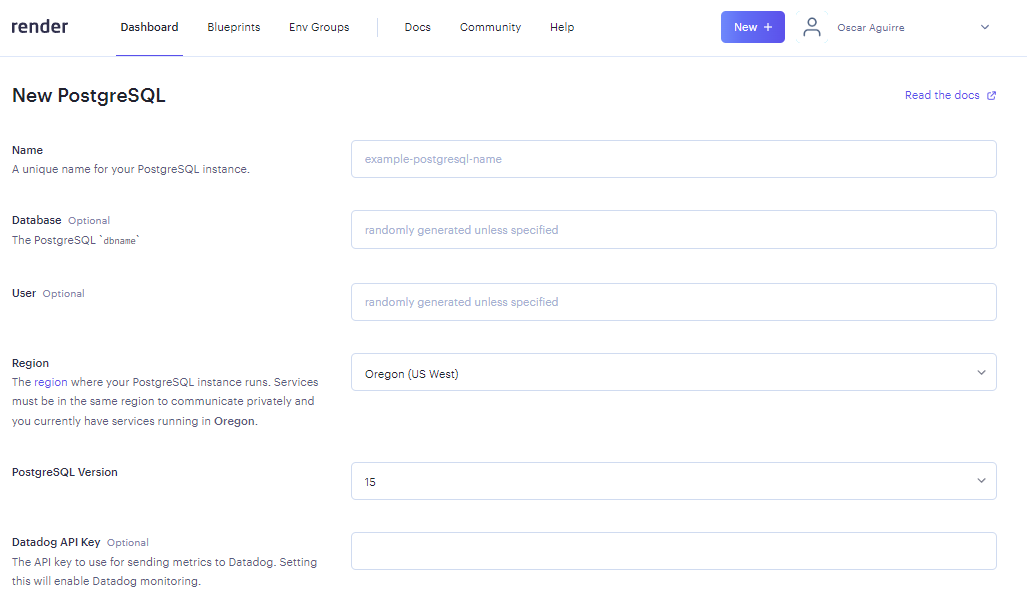
#### Paso 2: Configuración del Servicio

Haz clic en "NEW" y luego selecciona "PostgreSQL".



#### Paso 3: Opciones Adicionales

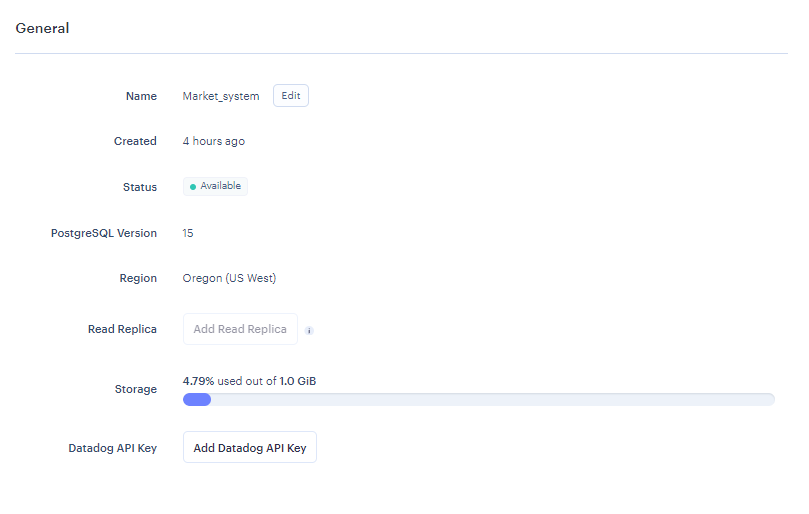
Aquí, proporciona un nombre al servicio (no a la base de datos), elige la región más cercana para reducir la latencia y selecciona la versión de PostgreSQL (como la versión 15 o la más reciente). Los campos "Database" y "User" son opcionales; Render generará valores predeterminados si no se completan. Selecciona el plan gratuito y crea la base de datos.

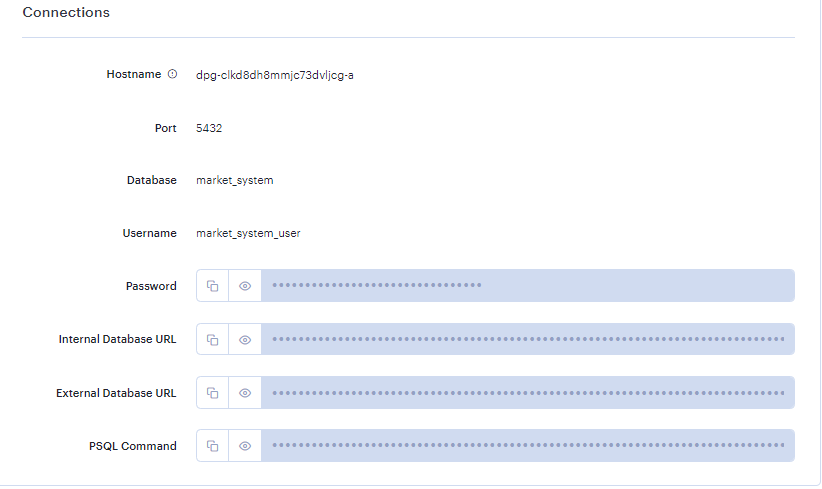




#### Paso 4: Obtención de Datos

Espera a que se complete la creación de la base de datos. Una vez finalizado, desplázate hacia abajo para obtener información crucial para la conexión con la API: anota el puerto, la base de datos, el nombre de usuario, la contraseña y la URL externa.

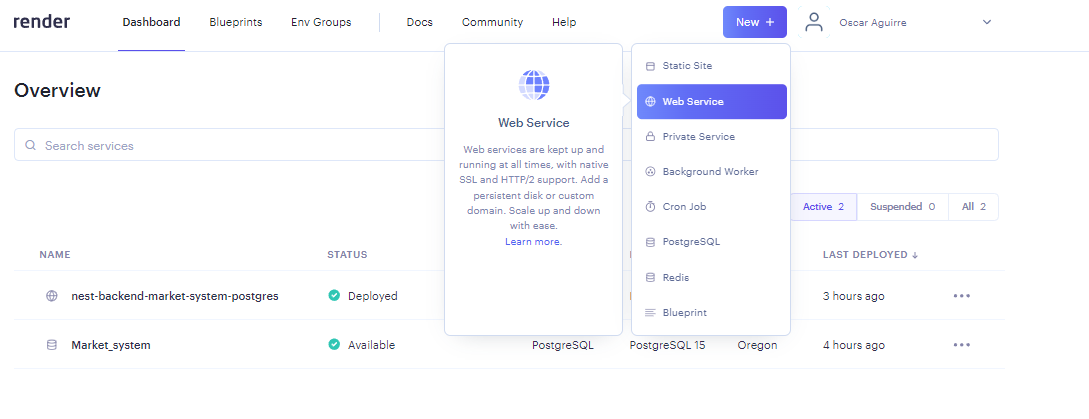


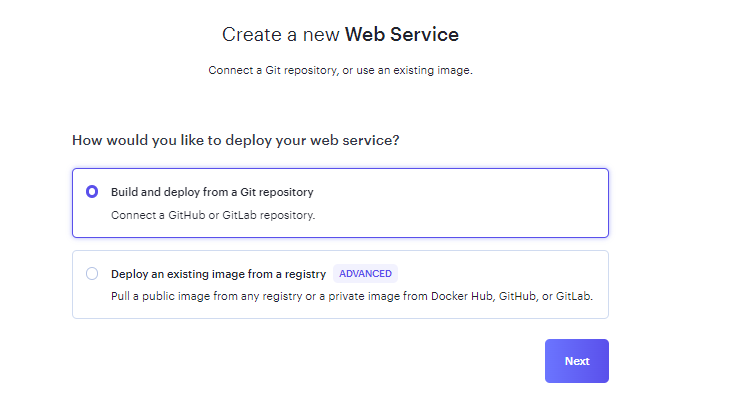


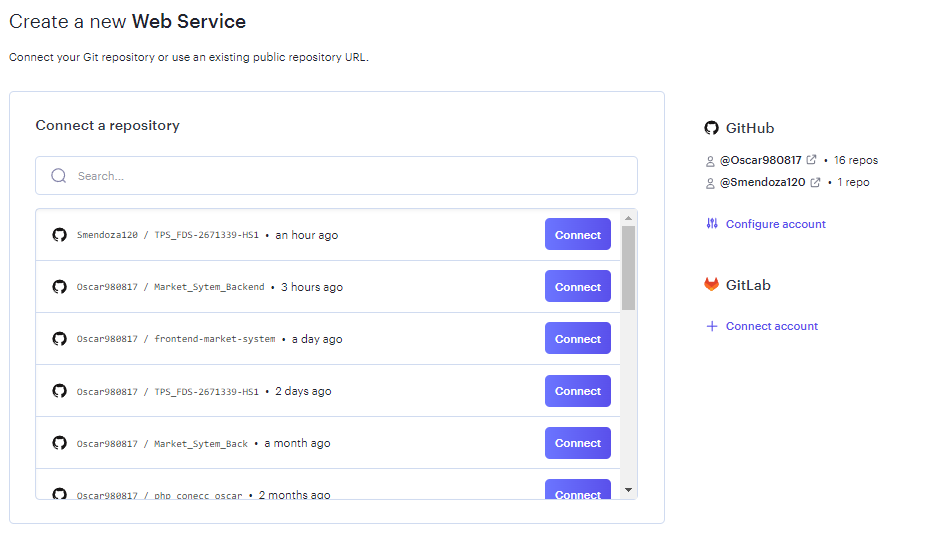
### **2. Creación del Servicio Web en Render**

#### Paso 1: Configuración del Despliegue desde un Repositorio Git

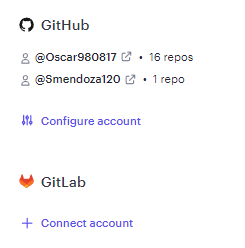
Selecciona "NEW" y luego "Web Service". Elige la opción para obtener el código desde un repositorio Git. Selecciona el repositorio correspondiente o configura la cuenta de GitHub si no está asociada.







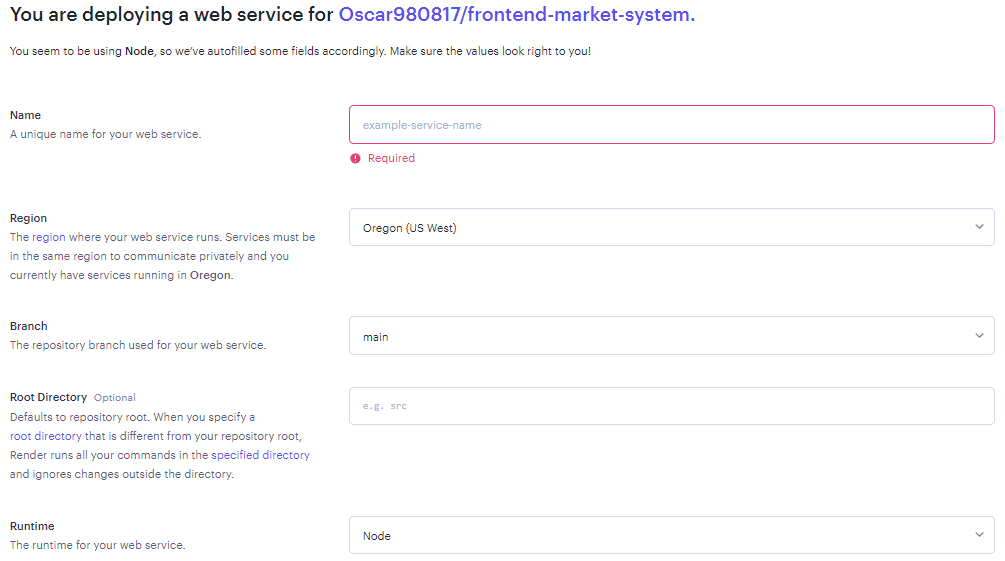
en caso de que el repositorio no esté asociado o la cuenta de github no aparezca seleccionaremos configure account, donde tendremos que ingresar el usuario y contraseña de github para que este lo reconozca



#### Paso 2: Configuración del Servicio Web

Rellena los detalles siguientes:

* Nombre del servicio web: Un nombre descriptivo para identificar el servicio en Render.
* Región para desplegar la aplicación o API: Selecciona la región más cercana al público objetivo para minimizar la latencia.
* Rama principal del repositorio: Por lo general, es "main" o "master", dependiendo de la estructura del repositorio.
* Directorio raíz: Puede dejarse por defecto, a menos que la estructura del repositorio requiera especificar una ruta específica.
* Tipo de lenguaje de programación o contenedor Docker: Selecciona el lenguaje de programación utilizado para la aplicación o especifica si se trata de un contenedor Docker configurado previamente para Render.

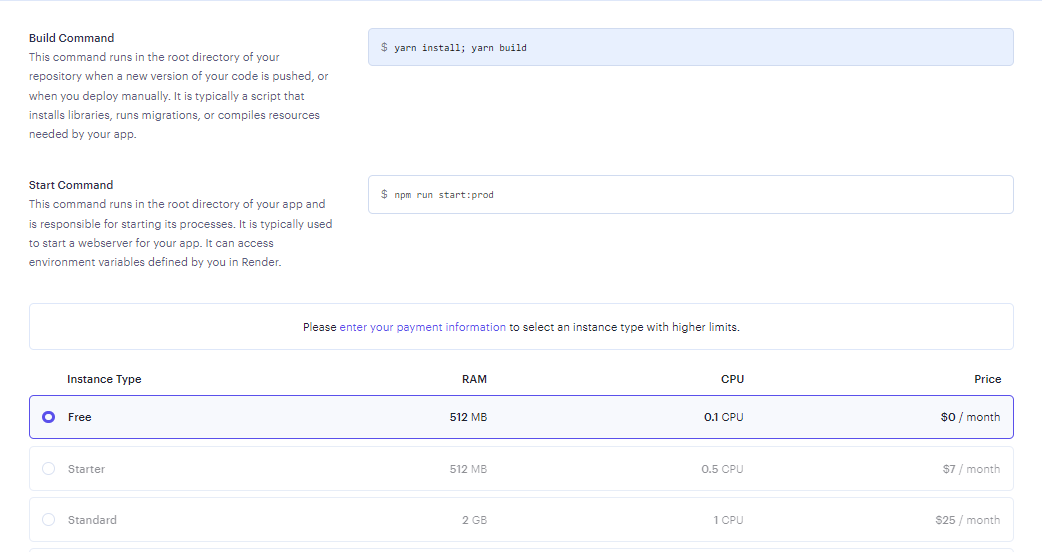


#### Paso 3: Comandos para Build y Start

En el campo "Build Command", introduce los comandos necesarios para instalar dependencias y realizar el build de la aplicación. Por ejemplo, en NestJS (un framework de Node), se pueden ejecutar comandos para instalar dependencias y construir el código para producción.

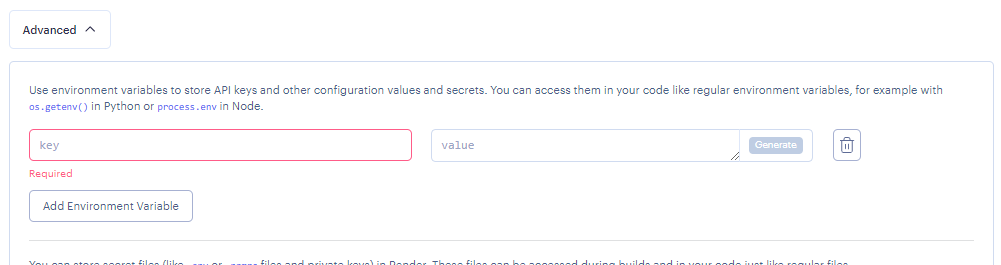
El campo "Start Command" debe ejecutar la carpeta dist, que contiene el código minificado listo para producción, obtenido generalmente desde el package.json.

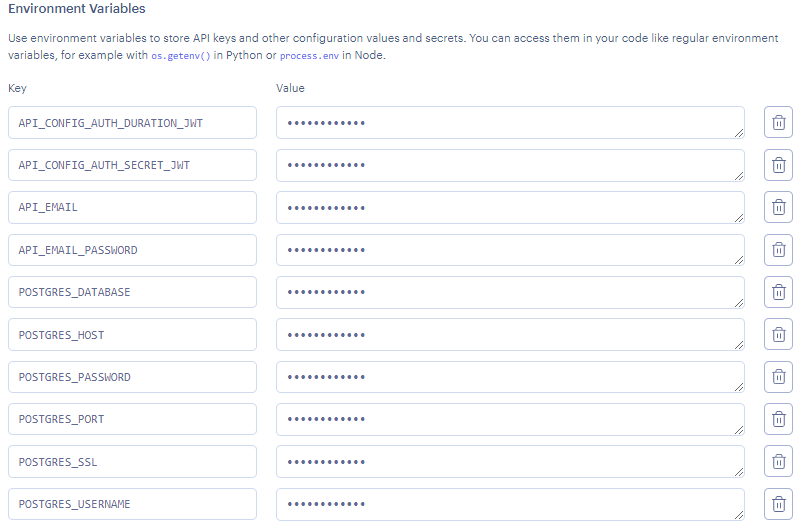




#### Paso 4: Configuración de Variables de Entorno

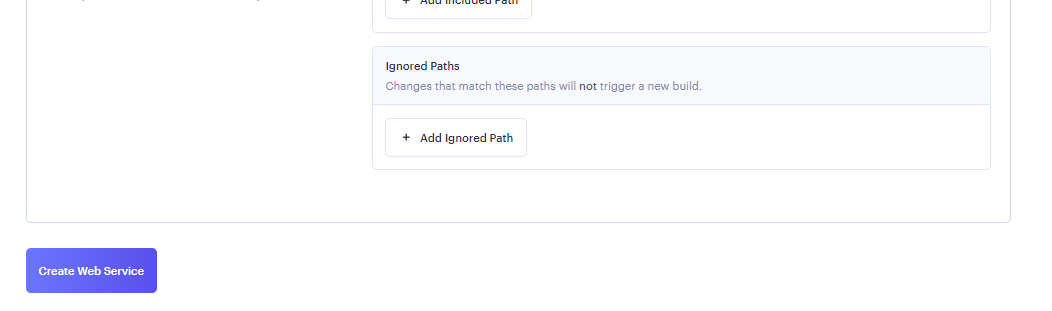
Si la API requiere variables de entorno, accede a "Advanced" antes de crear el servicio web. Aquí, puedes añadir las variables necesarias para el funcionamiento de la aplicación, como datos de conexión a la base de datos PostgreSQL.





#### Paso 5: Creación del Servicio Web

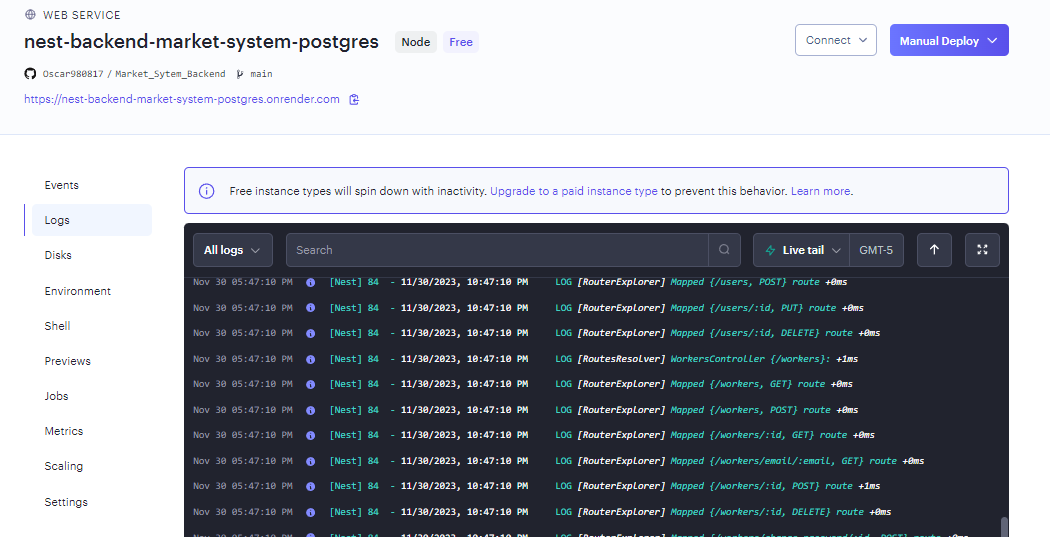
Finalmente, haz clic en "Create Web Service" para establecer el despliegue automático con cada commit en la rama principal del repositorio.



### **3. Verificación y Gestión**

#### Paso 1: Verificación del Enlace Generado

Una vez creado el servicio web, verifica el enlace generado para acceder a tu API o aplicación desplegada. Este enlace te permitirá realizar pruebas y compartir tu proyecto.



#### Paso 2: Revisión de Logs y Solución de Errores

En caso de errores durante la ejecución, accede a los registros (logs) proporcionados por Render. Estos registros te darán información detallada sobre cualquier problema que pueda surgir durante la ejecución de tu aplicación. Utiliza esta información para solucionar cualquier inconveniente.

#### Paso 3: Gestión Automática de Deploys

Recuerda que al configurar el servicio web en Render, se habilita la funcionalidad de deploys automáticos. Cada vez que hagas un push a la rama principal de tu repositorio, Render ejecutará automáticamente el despliegue con los nuevos cambios en tu aplicación.

#### Paso 4: Mantenimiento y Actualizaciones

Realiza un seguimiento periódico de tu servicio desplegado en Render. Asegúrate de mantener tus dependencias actualizadas y realiza pruebas regulares para garantizar un funcionamiento óptimo de tu aplicación o API.

### **4. Despliegue del frontend.**

#### Paso 1: Configurando Firebase

Primero debemos ingresar a la página web de Firebase, creamos una cuenta, una vez creada la cuenta vamos a agregar un nuevo proyecto.

Pantalla de celular con imagen de hombre

Descripción generada automáticamente

Una ves le demos clic a agregar el proyecto, nos redireccionara a otra pantalla donde agregaremos el nombre del proyecto.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Continuamos con la configuración del sitio, por lo que ahora nos mostrara una opción para incorporar Google Analytics, para esta configuración no la necesitamos, por lo que la omitiremos.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

La siguiente pestaña ya nos empezara a generar el proyecto, esperamos que finalice.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una vez finalizada la creación de nuestro proyecto, nos dirigimos al panel izquierdo en la sección configuración y damos clic en el ítem Hoisting.

Captura de pantalla con la imagen de una persona

Descripción generada automáticamente

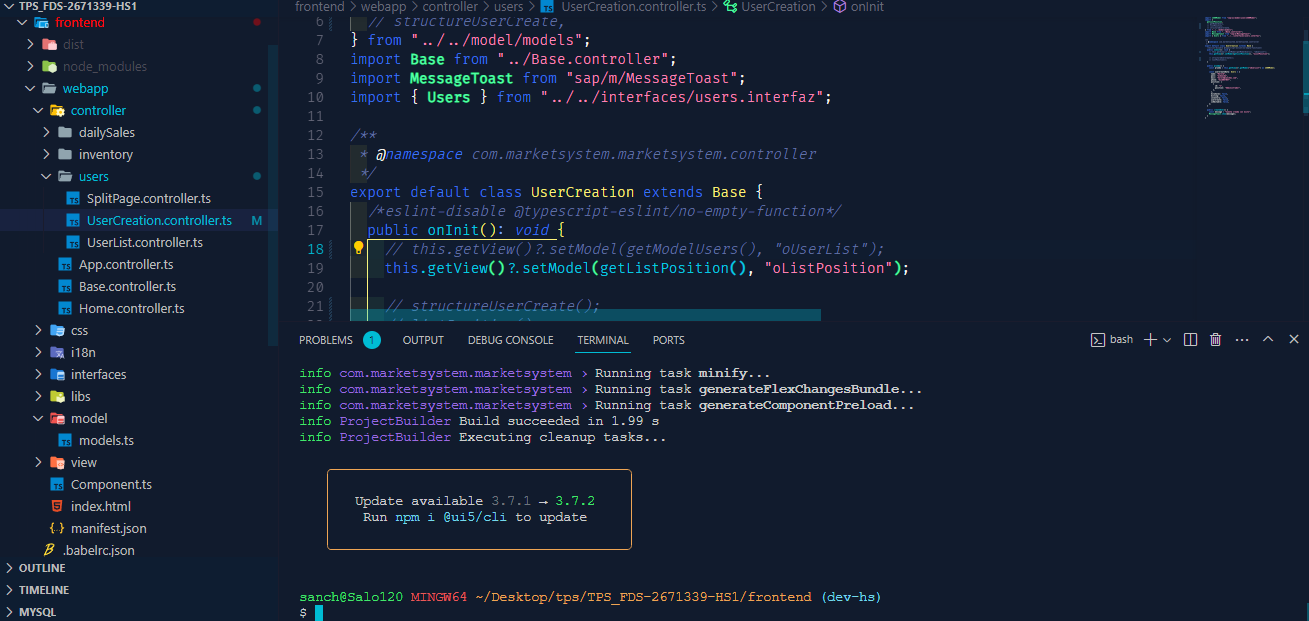
Ahora vamos a configurar el Hoisting, por lo que una vez sigamos el paso anterior, nos mostrara un botón para empezar con la configuración,

Captura de pantalla de un videojuego

Descripción generada automáticamente

#### Paso 2: Configurando el proyecto para el despliegue.

Antes de continuar en la pagina para subir nuestro aplicativo, configuraremos nuestro proyecto para poder subirlo sin dificultades, por lo que tenemos que realizar el comando de Build para que nos genere una carpeta dist, en nuestro caso el comando seria (npm build)



#### Paso 3: Desplegando el fronend.

Una vez que creamos nuestro proyecto y el código lo optimizamos para poder realizar el despliegue de este, ahora seguiremos con los pasos que nos muestra nuestro gestor de nube, por lo que volveremos a la pagina web y seguimos el paso a paso.

Tenemos que instalar las dependencias de Firebase dentro de nuestro proyecto, por lo que usaremos el comando (npm install -g firebase-tools)

Texto

Descripción generada automáticamente

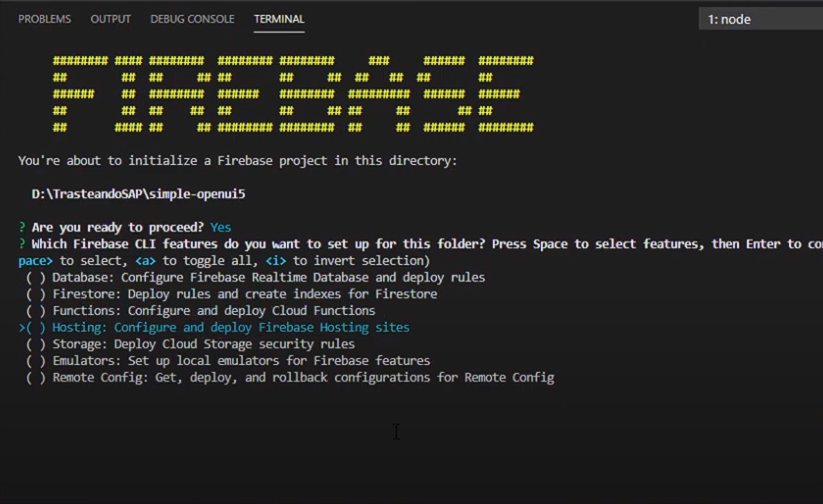
Damos clic en el botón continuar, una vez realizamos la instalación de dependencias de Firebase, ahora iniciamos sesion, para ello usamos el comando (firebase login), como ya tenemos una cuenta Firebase y actualmente nos encontramos logueados, realizara una autenticación SSO para comprobar nuestra cuenta.

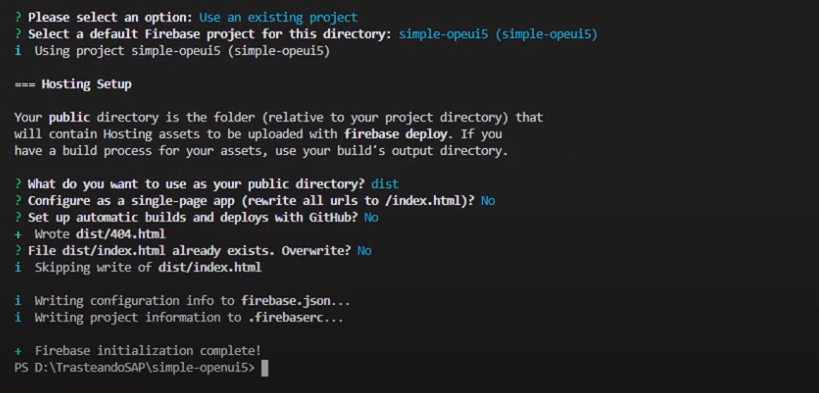
Captura de pantalla de computadora

Descripción generada automáticamente

Cuando nos logueamos ahora tenemos que inicializar Firebase, para ello usamos el comando (firebase init), y realizamos la siguiente configuración

* Marcamos la opción de Hoisting,
* Siguiente paso, nos preguntara si subiremos un proyecto existente o crearemos uno nuevo, por lo que seleccionamos que subiremos uno existente.
* Nos mostrara el proyecto, por lo que damos enter.
* Nos preguntara que carpeta tomara, por lo que le decimos que dist.
* Nos preguntara si queremos configurarla como una single app, damos que no.
* Nos pedirá que, si queremos configurar compiladores y despliegues automáticos, damos que no.
* Nos preguntara si realizaremos la compilación del proyecto, como se realizo anteriormente, marcamos la opción no.

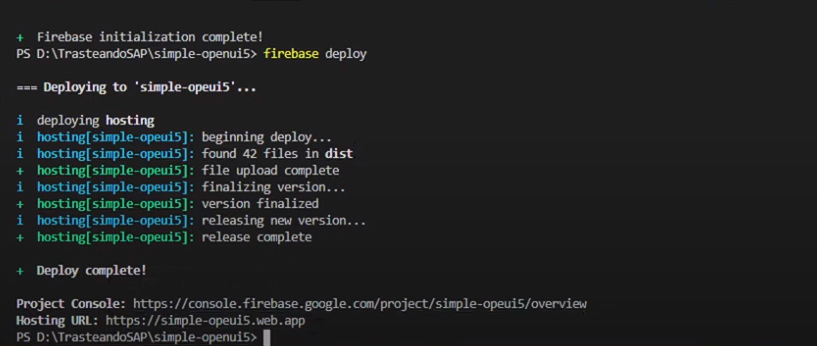




Por último agregamos el ultimo comando el cual es (firebase deploy), aquí empezaremos a subir el proyecto a nuestro proveedor de nube, cuando finalice nos darán una URL del hoisting, también puedes revisar este link desde la página Firebase.

Captura de pantalla de computadora

Descripción generada automáticamente



Captura de pantalla de computadora

Descripción generada automáticamente

Este es el link del aplicativo desplegado: https://marketsystem-1bbb4.web.app/