

Manual de instalación/despliegue

Proyecto: Sistema para gestión de actividades en el gimnasio del CEET - GymSenApp

Versión: 1.0

SENA

2024

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Mariana Lizeth Marín Rojas, Santiago Salamanca Díaz, Marco Antonio Ortíz Ramírez | Creación del documento | Marzo 25 del 2024 |

Tabla de contenido

[1. Introducción 4](#_Toc162044157)

[*1.1. Alcance del documento.* 4](#_Toc162044158)

[*1.2. Generalidades del proyecto.* 4](#_Toc162044159)

[2. Justificación del manual 5](#_Toc162044160)

[3. Objetivos del manual 5](#_Toc162044161)

[4. Requerimientos de conocimiento para manejar el sistema 6](#_Toc162044162)

[*4.1. Generales.* 6](#_Toc162044163)

[*4.2. Técnicos.* 6](#_Toc162044164)

[5. Desarrollo 7](#_Toc162044165)

[*5.1. Despliegue del Backend.* 7](#_Toc162044166)

[*5.2. Despliegue de la API.* 12](#_Toc162044167)

[*5.3. Despliegue del Frontend.* 16](#_Toc162044168)

[*5.4. Pruebas.* 20](#_Toc162044169)

# **1. Introducción**

## ***1.1. Alcance del documento.***

El propósito del presente documento es proporcionar una guía sobre cómo se realizó el despliegue del aplicativo en la nube, teniendo en cuenta que en servidores se alojaron sus tres componentes principales, por lo que se hará la descripción de las tres plataformas usadas y cómo se realizó el proceso.

En este sentido, se deben tener claras unas nociones generales sobre el proyecto, las cuales permitirán obtener una contextualización idónea sobre el proyecto de software. Veamos pues las consideraciones:

## ***1.2. Generalidades del proyecto.***

El sistema para gestión de actividades en el gimnasio del CEET, llamado “GymSenApp”, es un proyecto formativo el cual busca optimizar los procesos que engloba el uso del gimnasio del centro, permitiendo así una mejor planificación de las rutinas de ejercicios de los aprendices, así como también permite apoyar el acompañamiento y comunicación por parte del personal del gimnasio hacia los aprendices que asistan al lugar.

El sistema está compuesto de tres (3) grupos de actividades generales agrupadas según el tipo de usuario que puede acceder a ellas:

* + Administradores
  + Aprendices
  + Instructores

Cada uno de estos grupos o roles, está compuesto por actividades específicas:

* + Administradores
    - Gestión de anuncios
    - Gestión de usuarios
    - Generación de reportes
  + Aprendices
    - Visualizar anuncios
    - Visualizar recomendaciones saludables
    - Planificar rutinas
  + Instructores
    - Gestionar asistencias

# **2. Justificación** **del manual**

Este manual se está creando debido a su importancia a la hora de darle de alguna forma un final al software, de modo tal que se abren las puertas a que otras personas se unan al proyecto y nos ayuden a hacerlo crecer, por lo que empezará a hacerse necesario explicar a cada nuevo integrante cómo funciona el sistema, qué contiene y demás aspectos importantes del sistema. Ante este escenario, es imperativo preparar documentos que nos sirvan de apoyo para brindarles la capacitación necesaria a los nuevos integrantes y desarrolladores del proyecto, a fin de que conozcan por medio de elementos ajenos a nosotros, los creadores del proyecto, cómo desarrollar sus funciones dentro del sistema. De tal forma que, en este caso, vamos a ahondar en las plataformas en las que se desarrolló el despliegue del aplicativo, así como en la forma en la que se desplegaron los componentes dentro de dichas plataformas.

# **3. Objetivos del manual**

El manual de instalación o despliegue siendo creado en este documento tiene como objetivo principal brindarles a los nuevos desarrolladores que se unan al equipo, información relacionada al despliegue del aplicativo para que conozcan los procesos de despliegue que se realizaron para dejar el proyecto funcionando en la nube. Adicionalmente, comprende 2 objetivos específicos:

1. Brindar una guía para que los usuarios correspondientes, sepan cómo reproducir el proceso ejecutado para realizar el despliegue del aplicativo en la nube.
2. Brindar ítems que les permita conocer a los usuarios correspondientes un poco sobre el manejo de las plataformas usadas para desplegar los componentes del sistema.

# **4. Requerimientos de conocimiento para manejar el sistema**

## ***4.1. Generales.***

El usuario debe tener conocimientos básicos y hasta un poco específicos sobre Frontend, Backend y bases de datos, así como deben saber interpretar gráficos que acompañan el desarrollo de software, esperando que logren comprender fácilmente diagramas como lo son los UML.

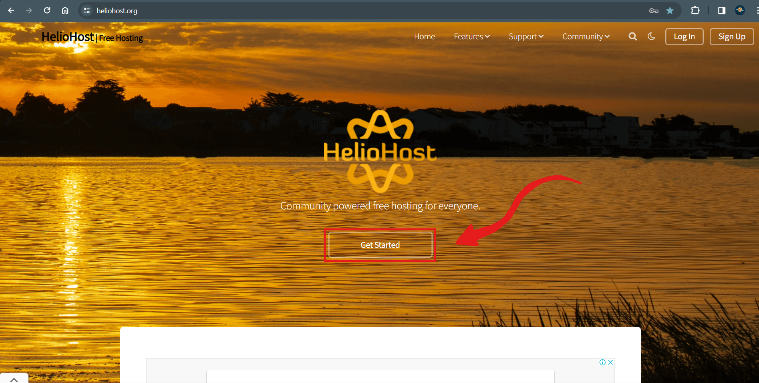
## ***4.2. Técnicos.***

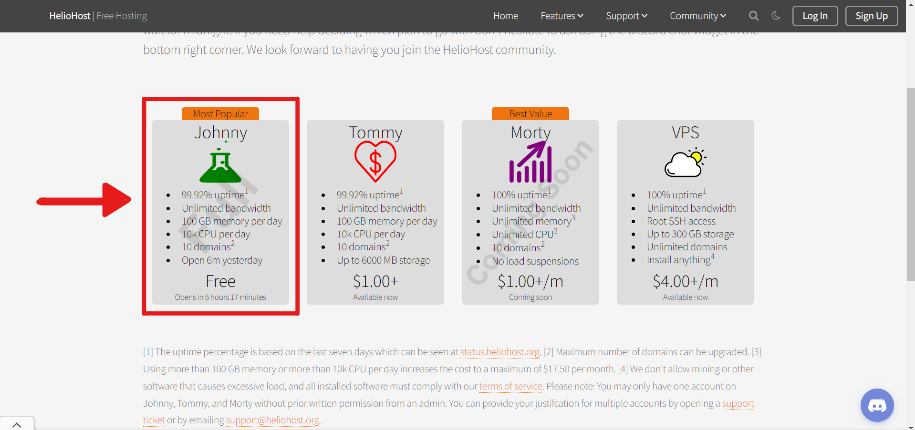
En cuanto a hardware se requerirá mínimo de los elementos básicos que contiene un computador (teclado, mouse, y demás).

En cuanto a software de forma general se requerirá de un sistema operativo en lo posible recientemente sacado, pensando en que se puedan aprovechar las características que estos sistemas operativos brinden; así como se requerirá, por otro lado, de mínimo un navegador de internet. Todo lo anterior enfocado en que se quiera analizar el sistema de forma general, ya desplegado. Si se desea trabajar en el código, se requiere de un editor de código, se requiere tener instalado NodeJS (en lo posible la versión 18.6.6), se requiere tener instalado Angular en el equipo, también se requiere de un emulador de servidor (como Xampp) y de un gestor de bases de datos, en el caso MySQL.

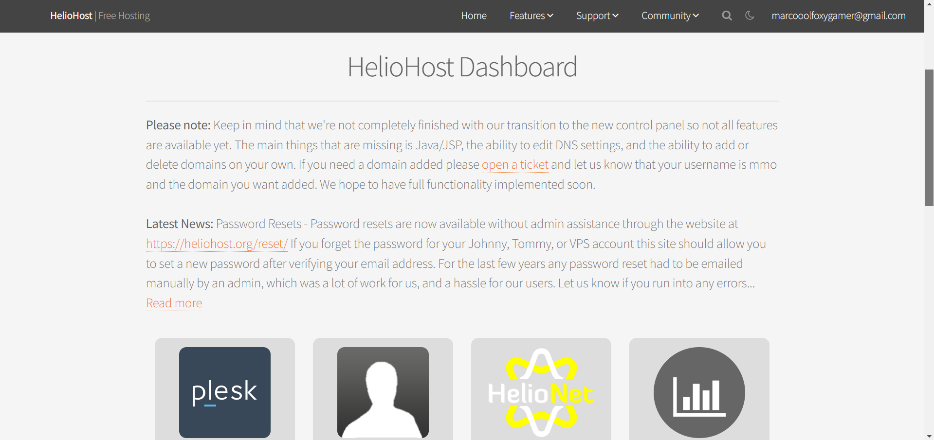
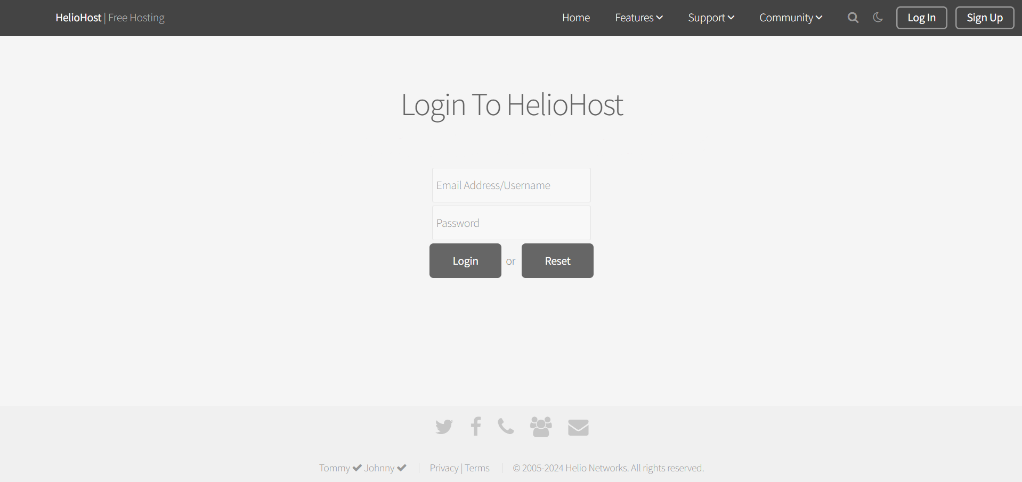
# **5. Desarrollo**

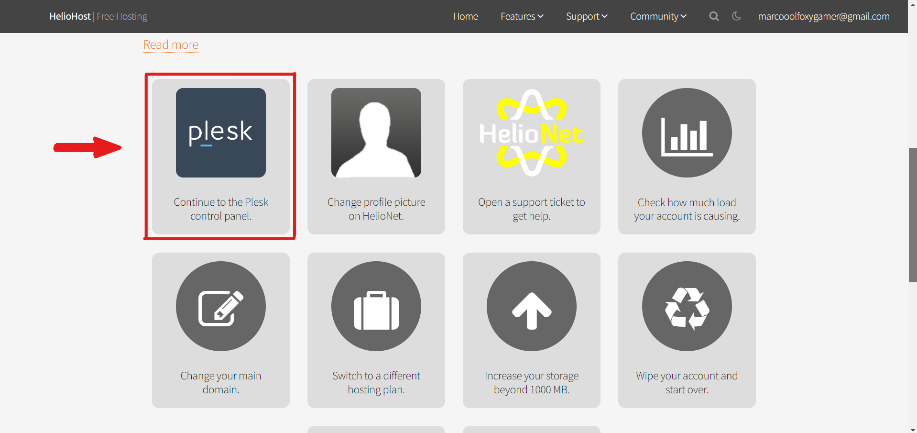
## ***5.1. Despliegue del Backend.***

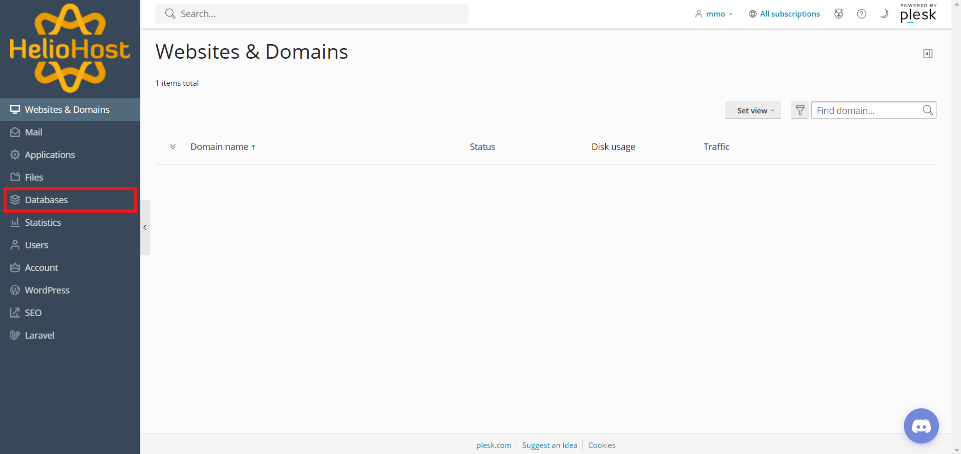
Empezamos accediendo a la página del hospedador, en este caso, HelioHost. Luego le damos en “Get Started” para empezar el proceso de despliegue de la base de datos:

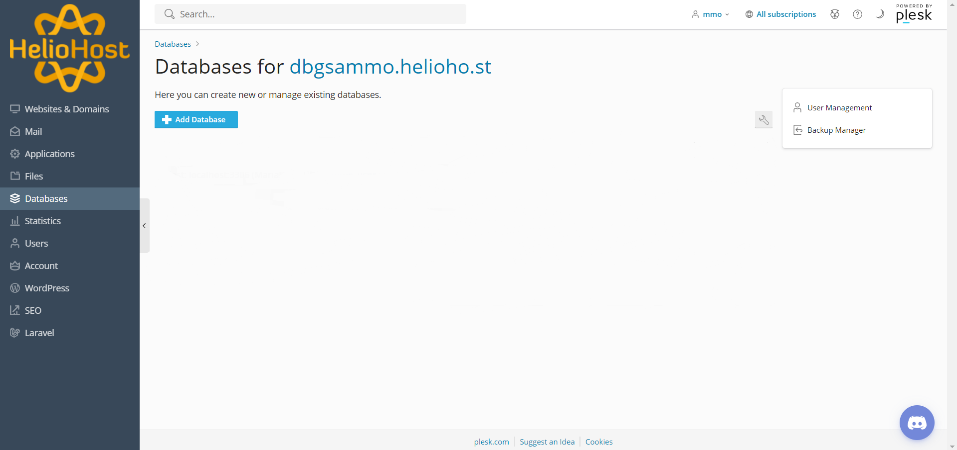
Ahora se nos mostrarán los planes con los que cuentan; para acceder al gratuito (que es el primer plan, llamado “Johnny”) tendremos que esperar 24 horas (suponiendo que nadie más tome el plan cuando sea liberado, o si no, habrá que esperar otras 24 horas):

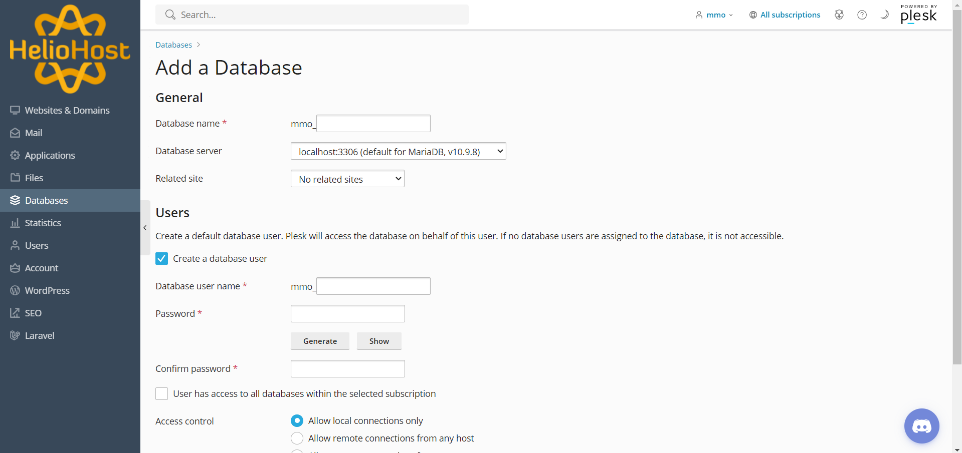
Cuando logremos seleccionar el plan, nos saldrá una página para registrarnos, allí ponemos nuestros datos. Luego tendremos que revisar nuestro correo ya que allí nos será enviado un mensaje para verificar que el correo está activo. Posterior a la confirmación de la existencia del correo, la página nos redireccionará al inicio de sesión; allí ponemos la información que corresponde:

Luego de esto, se nos redirigirá a la siguiente página:

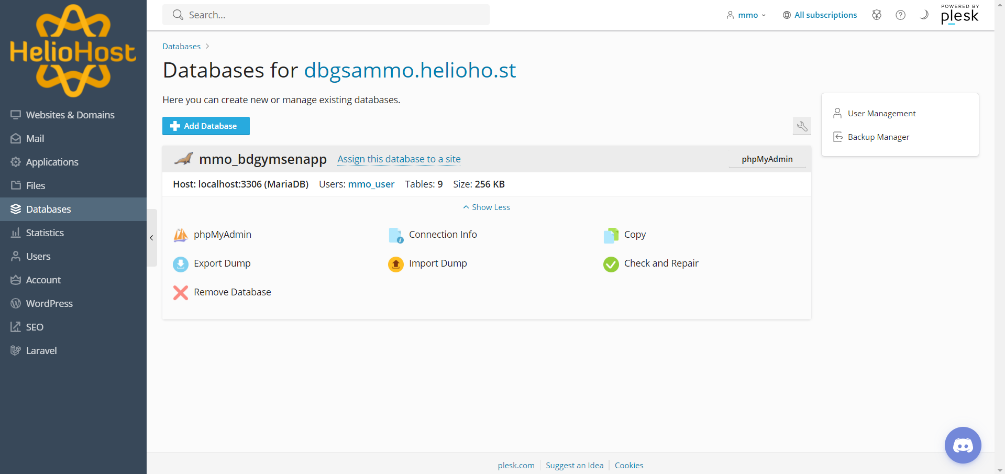
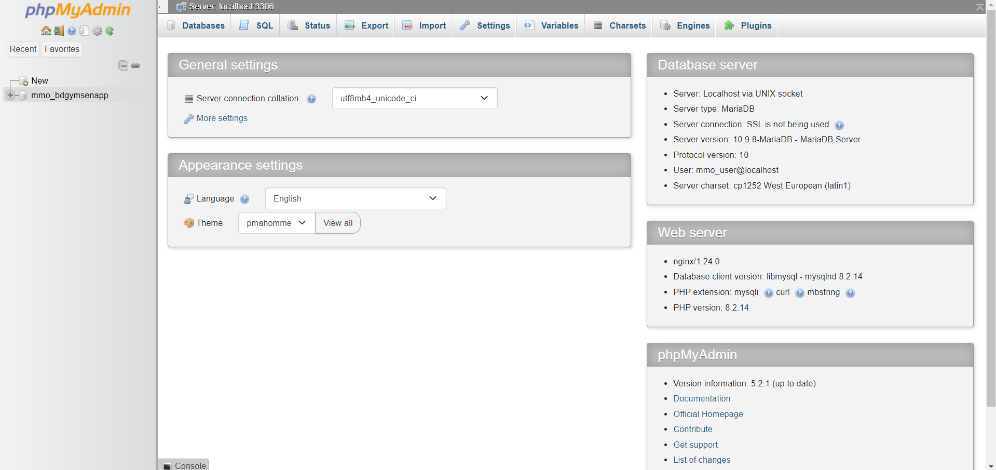
 Allí bajamos y seleccionamos la opción para continuar con el panel de control Plesk:

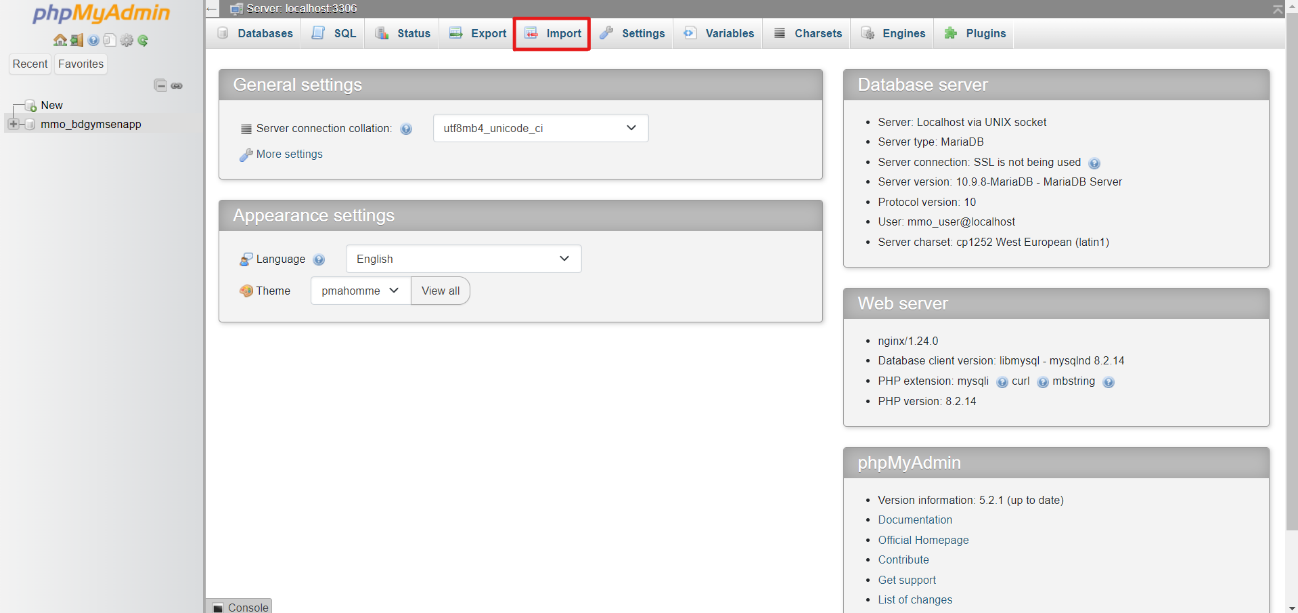
 Seremos redirigidos al panel de control que nos permite gestionar lo relacionado a los servicios que tengamos con este proveedor; en el panel izquierdo seleccionamos la opción “Databases” para ejecutar las acciones correspondientes a la subida de la base de datos del sistema:

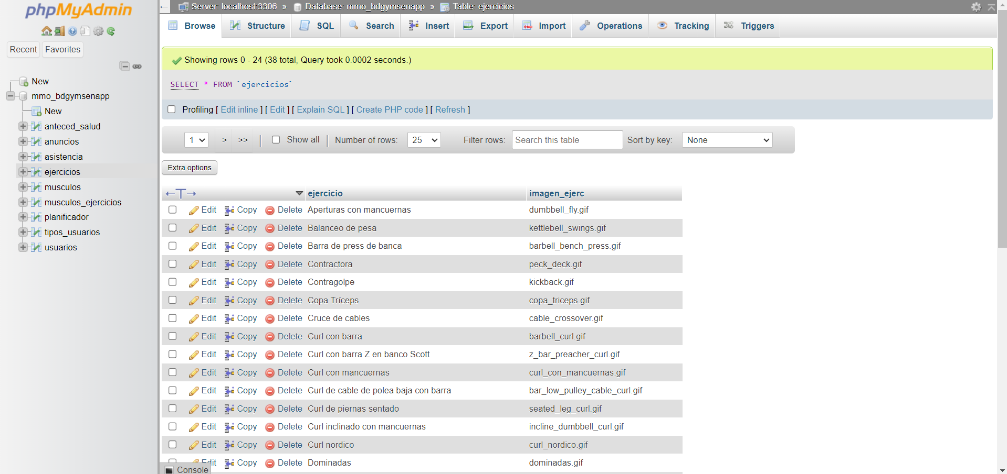


 Ya estando aquí, seleccionamos la opción “**+** Add Database”:

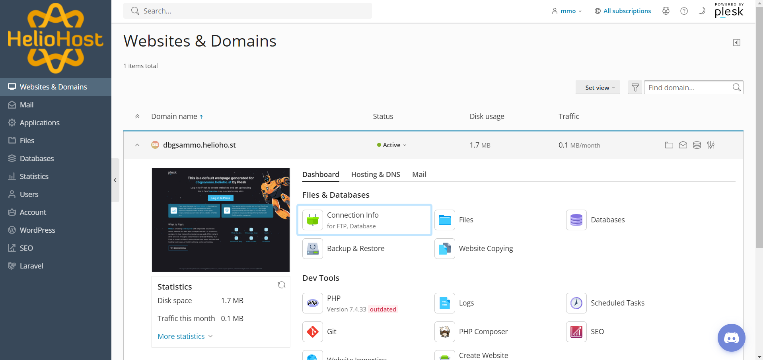
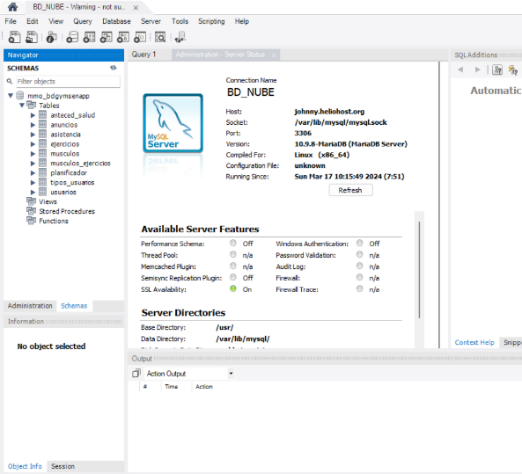
En el formulario llenaremos los campos que correspondan, teniendo en cuenta que aquí ya estaremos creando la base de datos que alojará la información del sistema. Cuando ya la tengamos creada, se nos mostrará en la sección “Databases” la nueva base de datos creada, como se muestra a continuación:

 Ahora seleccionamos la opción phpMyAdmin y se nos abrirá en otra página el gestor de bases de datos con la base de datos recién creada:

 Lo que luego haremos será exportar nuestra base de datos local para posteriormente importarla dentro de la base de datos de la nube:

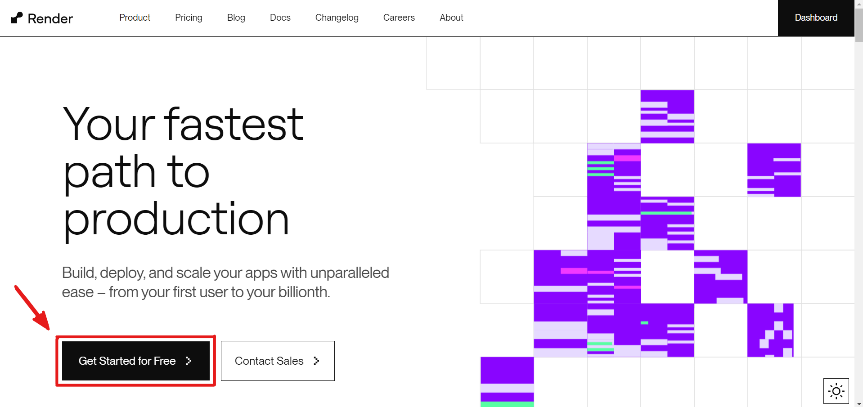
 Corroboramos que la información se haya recuperado correctamente:

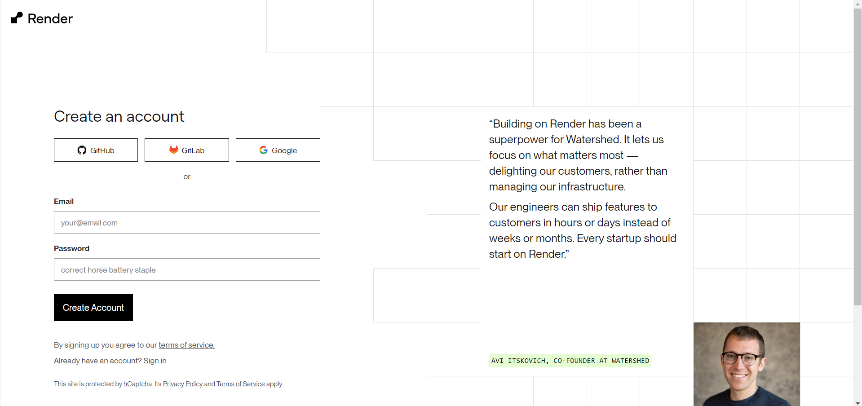
De esta forma ya tendremos en el servidor nuestra base de datos funcionando. Cabe destacar que al crear la misma se nos dieron los datos correspondientes para conectarnos, de modo que esta información podremos usarla para que la API pueda acceder a la información y gestionarla como corresponde.

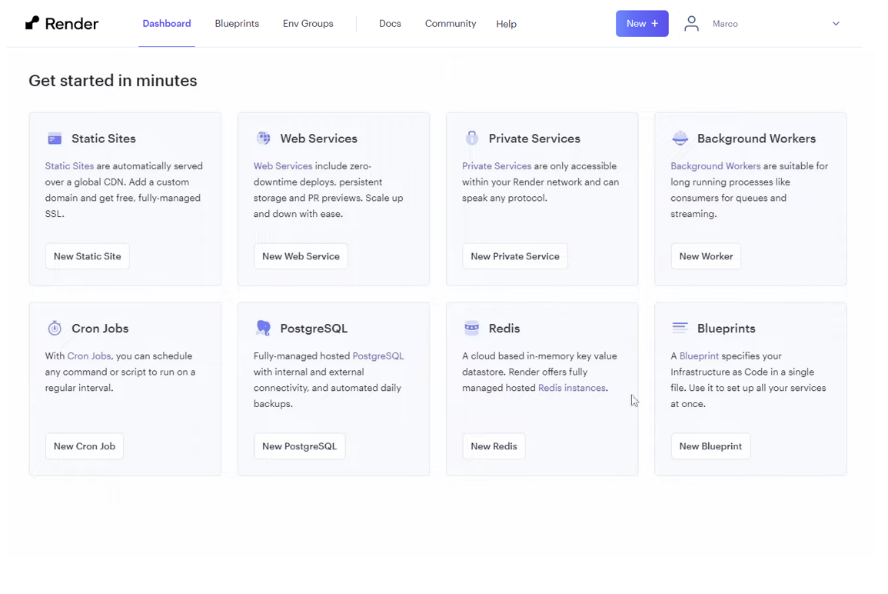
 Por si se nos pierde la información de conexión, en la opción “Websites & Domains” en el panel de HelioHost, se nos muestra una opción que dice “Connection Info”; le podemos dar clic y allí se nos mostrará la información correspondiente para conectarnos a la base de datos, e incluso se nos permite visualizar la dirección IP para, por ejemplo, acceder a la base de datos desde herramientas como MySQL Workbench o incluso desde la terminal):

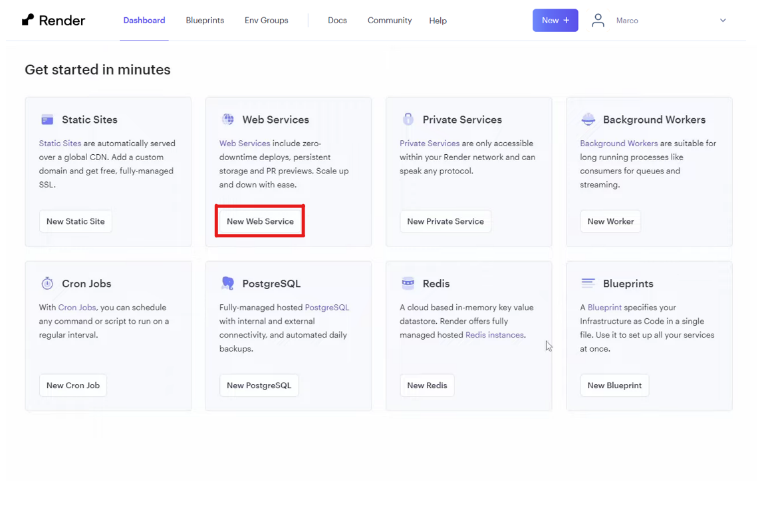
## ***5.2. Despliegue de la API.***

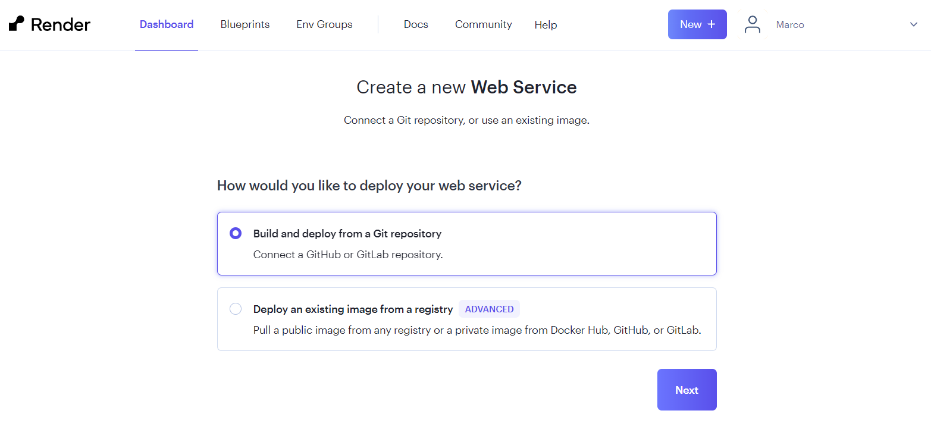
Vamos a realizar el despliegue de la API por medio de Render. Empecemos:

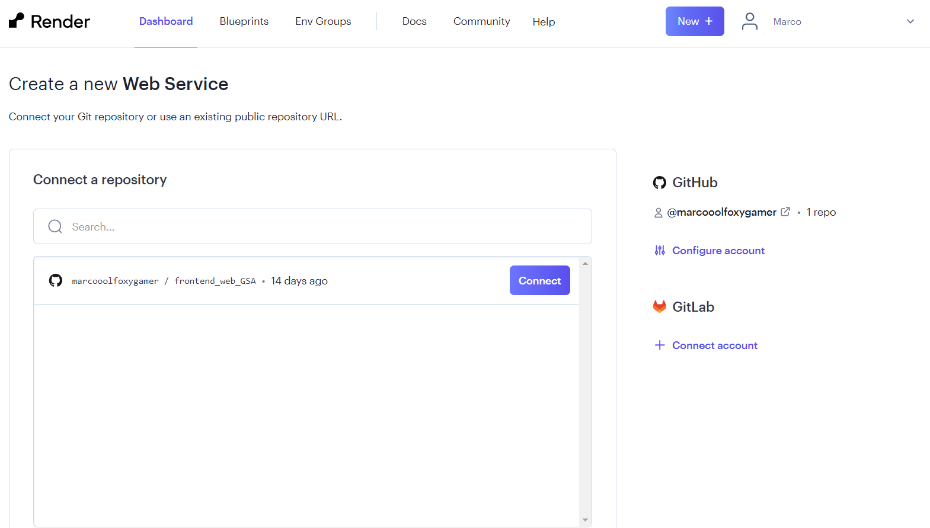
 Primero, debemos entrar a la página de Render (<https://www.render.com/>), y seleccionamos la opción “Get Started for Free”:

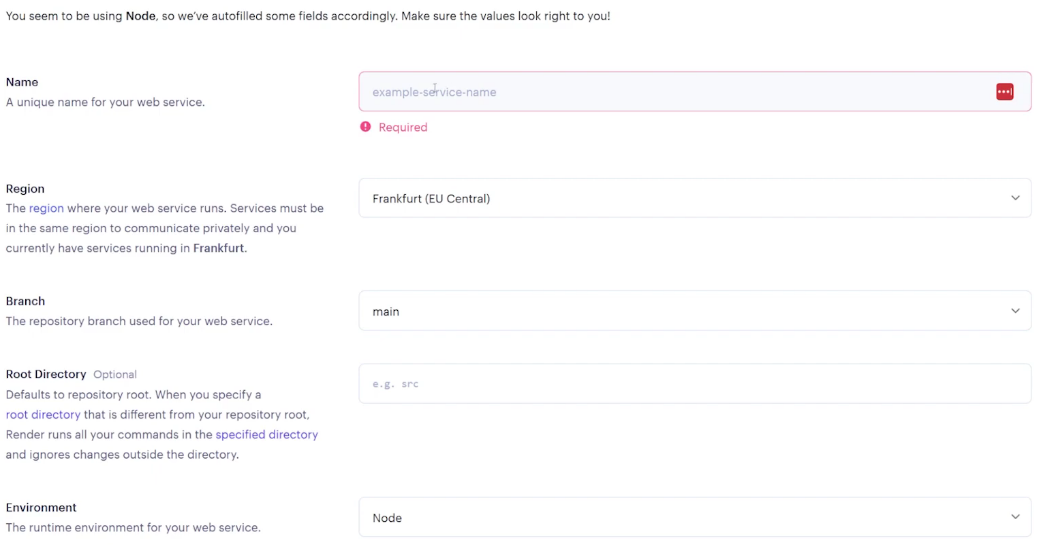
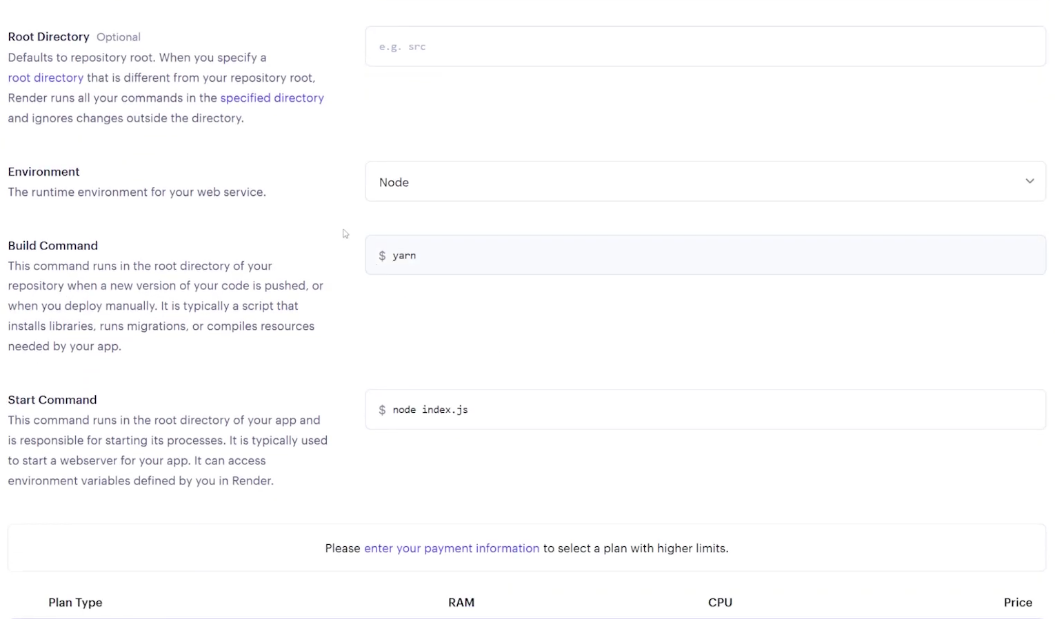
 A continuación, nos registramos:

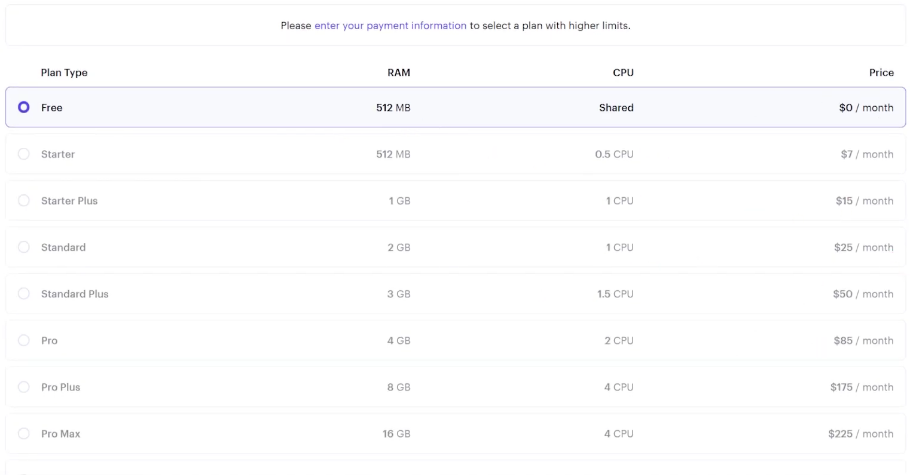
 Luego de hacerlo, nos llegará al correo un mensaje para verificar la dirección de correo electrónico brindada a la plataforma, le damos clic al enlace para verificar, y si todo sale bien se nos iniciará la sesión y nos saldrá algo como lo siguiente:

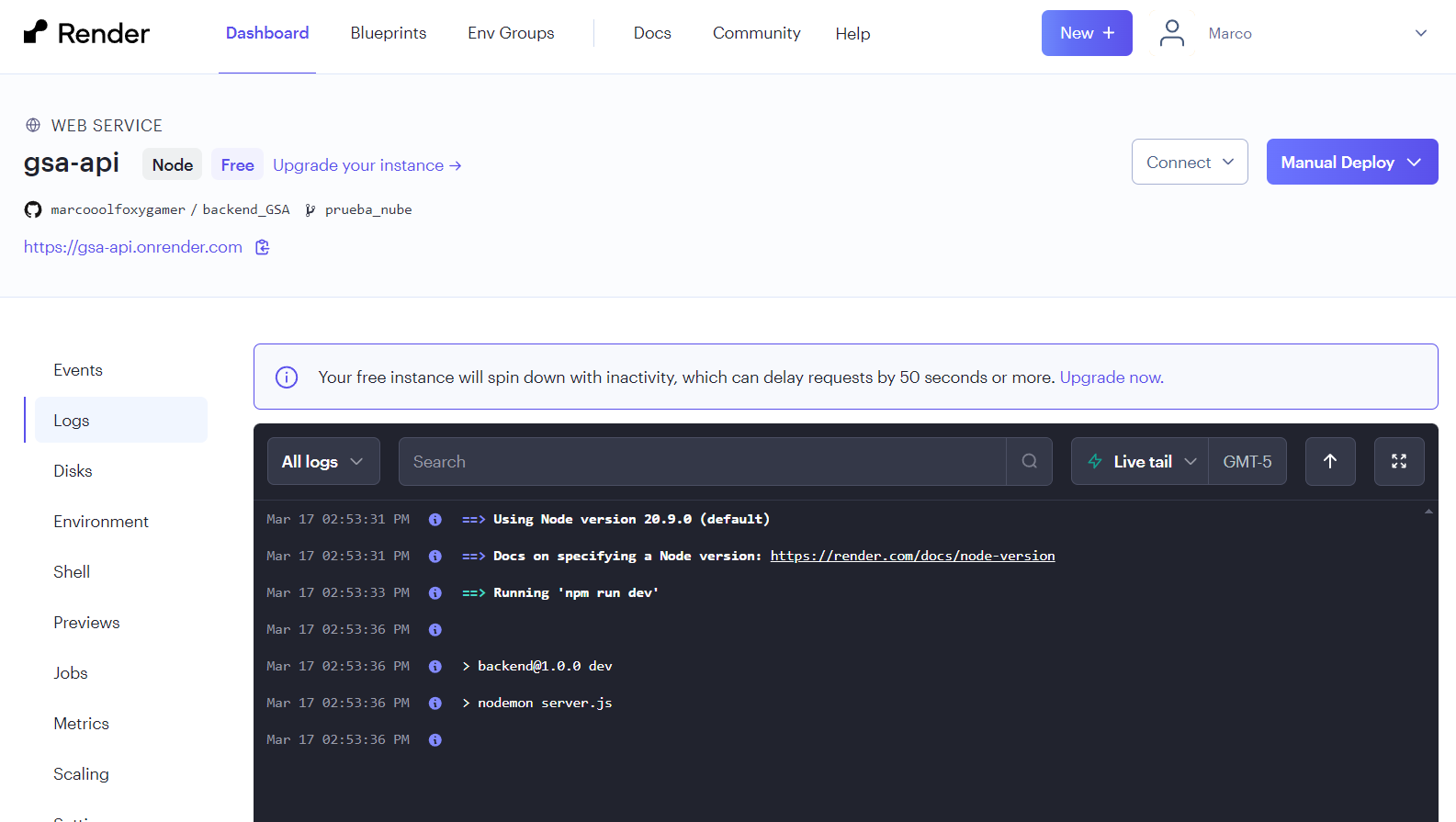
 En esta pestaña se nos permitirá escoger el tipo de servicio que queramos desplegar, en el caso será la API, entonces debemos escoger la opción “Web Services”:

 Al seleccionarla, nos saldrá una pestaña para escoger de dónde se obtendrá y desplegará el servicio, en el caso se desplegará basado en un repositorio de Git:

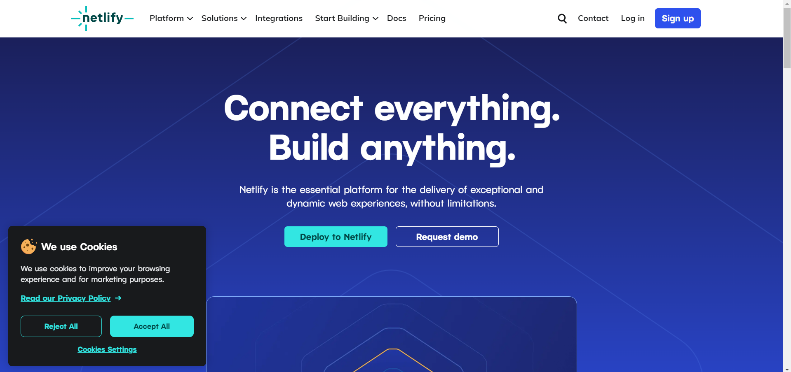
 Ahora se nos pedirá el repositorio de donde vayamos a generar el servicio; tenemos dos formas de vincularlo: Una es ingresando una cuenta de GitHub y seleccionando de todos nuestros repositorios cuál se va a desplegar, o, ponemos el enlace del repositorio público a desplegar; cualquiera de las dos opciones servirá (cabe destacar que, si no nos registramos en render con GitHub y elegimos la primera opción, tendremos que vincular la cuenta de GitHub, teniendo hacer unos pasos de más):

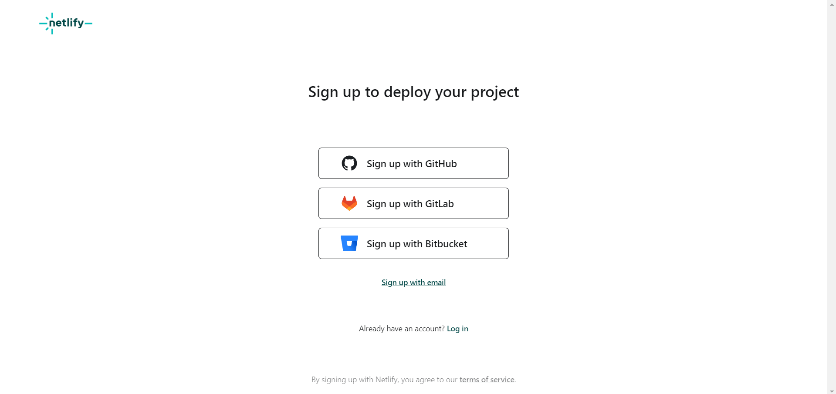
 Ahora, llenaremos los datos del servidor a crear, teniendo en cuenta las ramas del repositorio (a cuál se enlazará para tener funcionando el servicio de acuerdo a los cambios que se realicen en la rama), las características de las dependencias que necesitemos para que nuestro componente funcione como servicio correctamente (en el caso, como es la API construida con NodeJS, necesitaremos de NodeJS, de un comando para ponerlo a funcionar (en el caso “npm run dev”), etc.), así como especificaremos la ruta de los archivos a servir (si es que no están directamente en la carpeta root del repositorio):

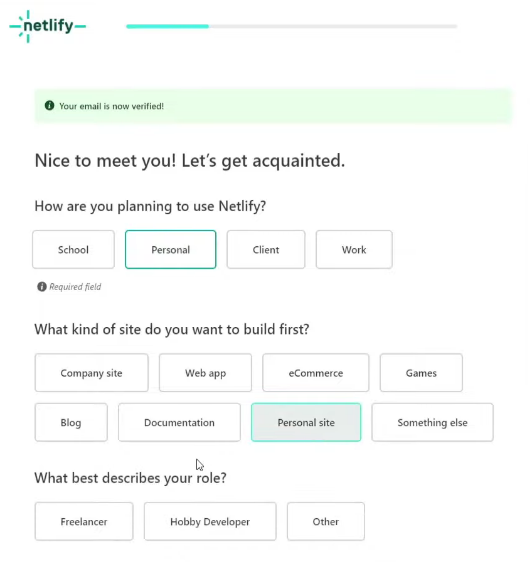
 Más abajo se nos mostrarán las formas de ponerlo a funcionar según los planes a los que podríamos acceder. En el caso, escogeremos el gratuito:

 Ya llenos los campos como corresponde, le damos al botón para crear el servicio y listo, nos cargará una página en la que tendremos todo lo relacionado al servicio; inicialmente se nos mostrará la sección “Logs” y una terminal, en la que se nos mostrarán los procesos que esté realizando render para vincular el repositorio según la rama indicada, y así dejar funcionando el servicio. Luego de un tiempo, llegará el punto en que nos saldrán en esta terminal unos textos iguales a cuando ponemos a funcionar nuestro componente de forma local, de esta forma sabremos que ya está funcionando el servicio con el componente desplegado (en el caso, la API):

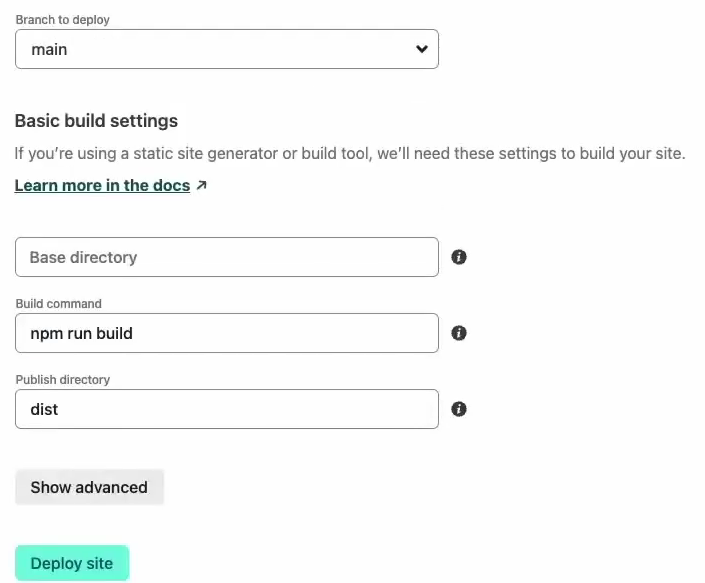
## ***5.3. Despliegue del Frontend.***

Primero, entramos a la página de netlify que será en donde desplegaremos el Frontend del proyecto:

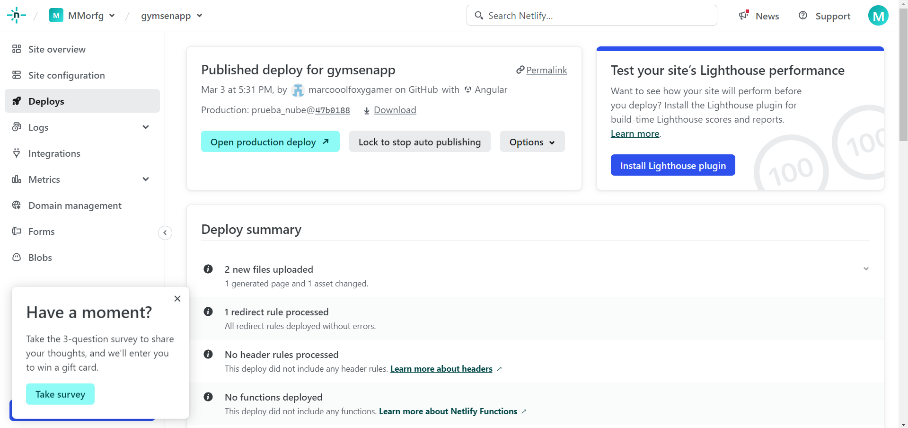
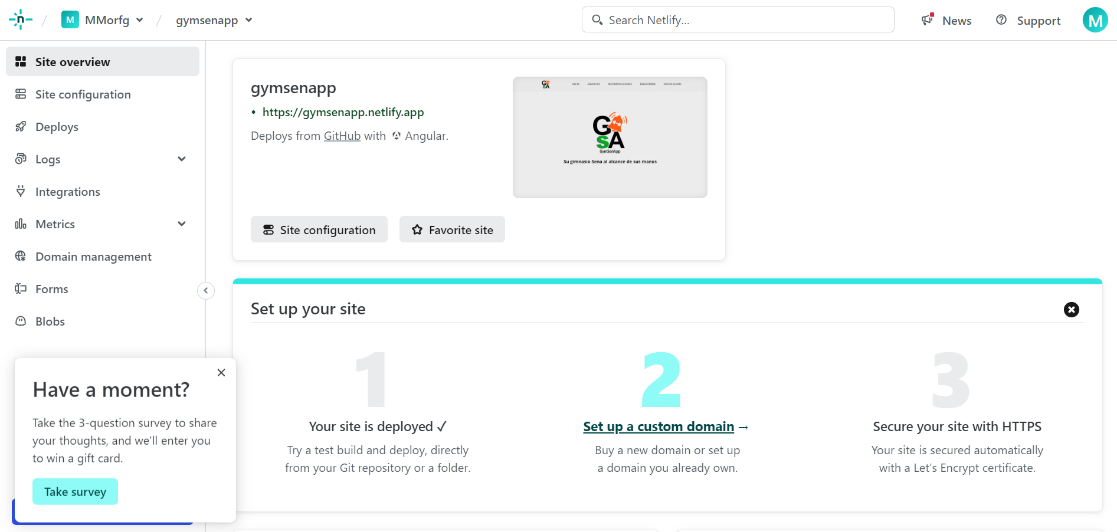
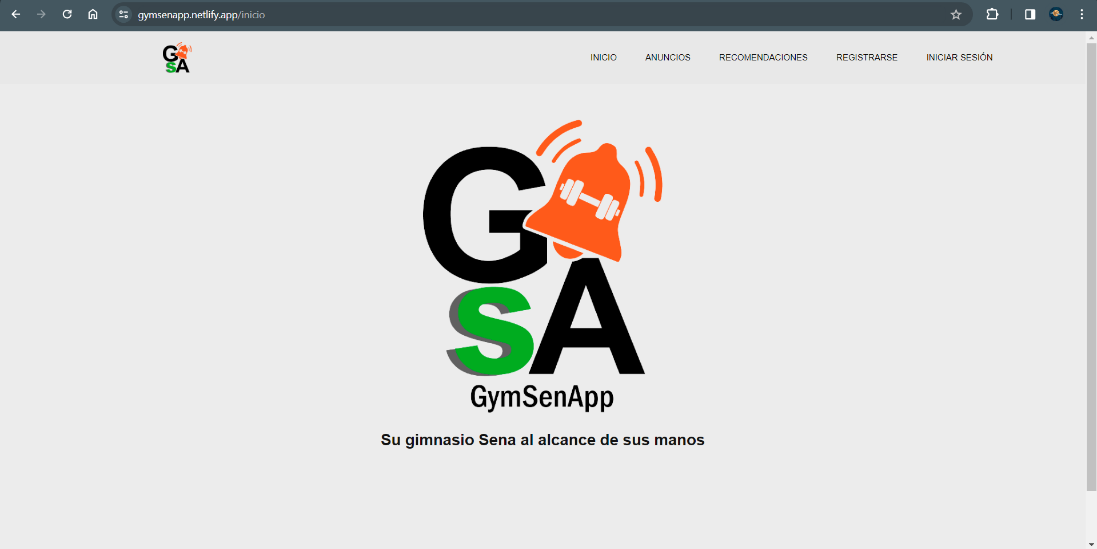
 Le damos clic a “Deploy to Netlify” o a la opción del menú “Sign up”; en la página que nos cargue tendremos que registrarnos:

 Cuando nos registremos debemos mirar el correo ya que se nos enviará un correo de verificación; hecho lo anterior, se nos enviará a una página en la que se nos harán unas preguntas iniciales para personalizar nuestra experiencia con la página y los servicios de netlify. Digitamos los datos según corresponda:

 Luego de diligenciar el formulario, se nos mostrará algo como lo siguiente:

 Seleccionamos la opción que corresponda según donde esté el componente a desplegar, en el caso, seleccionaremos “Deploy with GitHub”. Luego de lo anterior, tendremos que desarrollar un pequeño proceso para vincular nuestra cuenta de GitHub, luego seleccionaremos el repositorio en donde esté el componente a desplegar y llenaremos un formulario describiendo elementos importantes como ¿Qué aplicación usa para funcionar?, ¿Con qué comando se ejecuta?, ¿Hay alguna ruta específica a la que toque dirigirse para poner a funcionar el componente?, entre otros:

Luego de darle en desplegar, se nos mostrará una página con unas secciones y puntos que nos informan sobre qué etapas debe pasar el aplicativo o componente siendo abierto por netlify para concluir que fue desplegado satisfactoriamente, también se nos informa sobre cómo va el proceso de despliegue, así como adicionalmente se nos muestra una terminal en la que visualizamos los procesos que está ejecutando netlify para clonar el repositorio, acceder a sus recursos y ponerlo a funcionar:

 Ya hecho este proceso, en la parte superior debería habilitarse un espacio con un enlace para acceder al componente desplegado. Si no es así, nos podremos dirigir a la opción “Site overview” en el menú de la izquierda y entre los elementos que nos muestre debe estar el enlace general del componente. Si accedemos al mismo, se nos cargará el componente desplegado (en el caso, el Frontend del sistema):

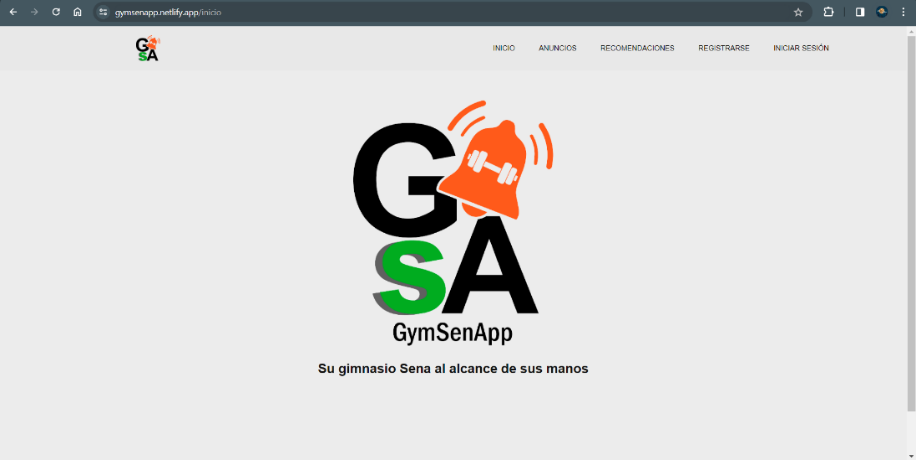
Antes de seguir con el siguiente tema, queremos hacer una aclaración, y es que, en el caso de este Frontend hecho en Angular, nos presentó unas fallas en los redireccionamientos; lo que ocurría era que, al tener el aplicativo que hacer un “refresh” o recargar la página (acción que en el código se indicó debía realzar bajo ciertas condiciones), no las realizaba, y en vez de esto se mostraba un error 404 indicando que la página no existía. Para solucionar este inconveniente tuvimos que realizar dos cambios en el código:

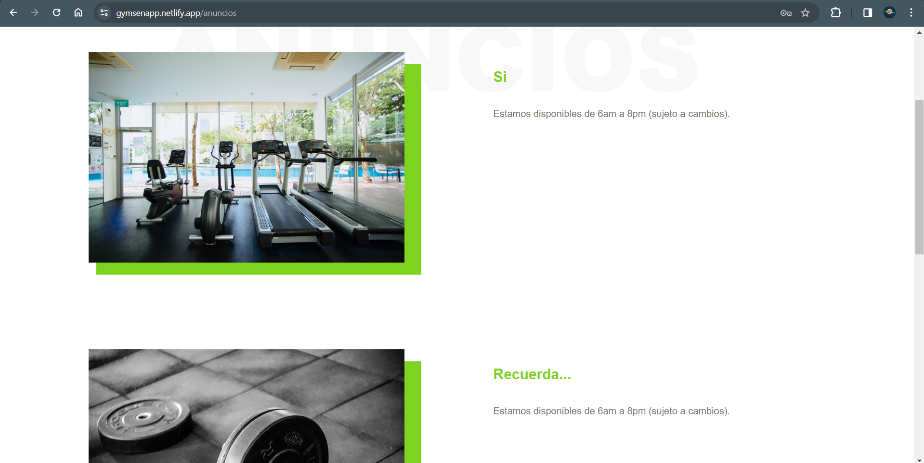
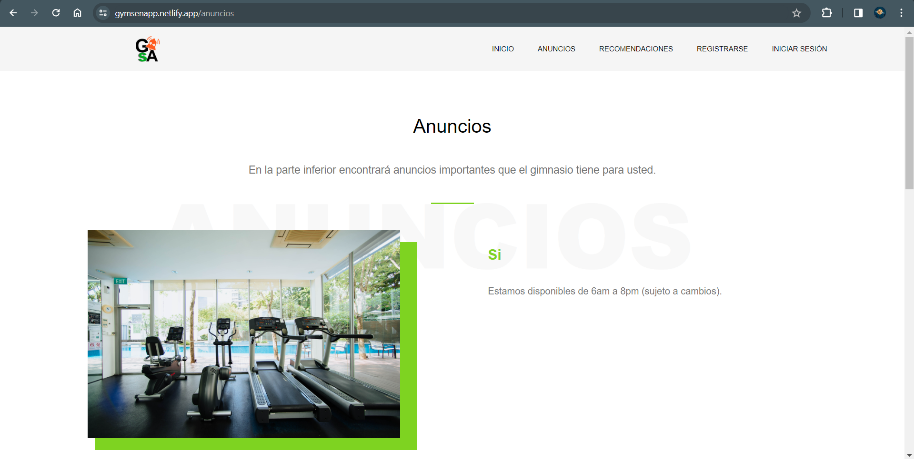
1. Creamos dentro de la carpeta “src” un archivo llamado “\_redirects” y se puso el siguiente código:
2. Dentro del archivo llamado “angular.json”, hubo que añadir este archivo creado anteriormente dentro del arreglo “assets” de la siguiente manera:

Con esto hecho, y realizando un deploy del último commit de la rama, el servicio agregó los cambios y ahora si el Frontend funcionó al 100%, ejecutando las acciones que le corresponden, adecuadamente.

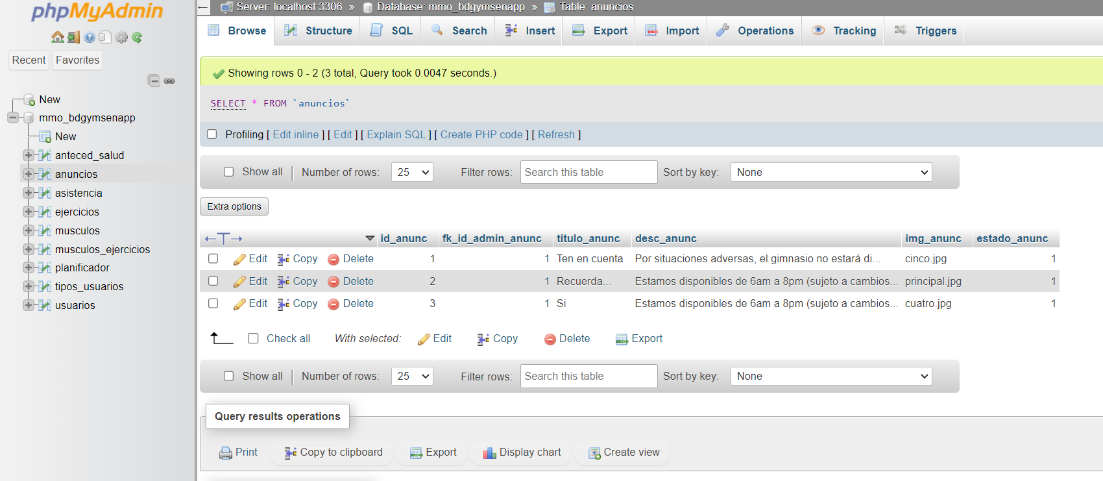
## ***5.4. Pruebas.***

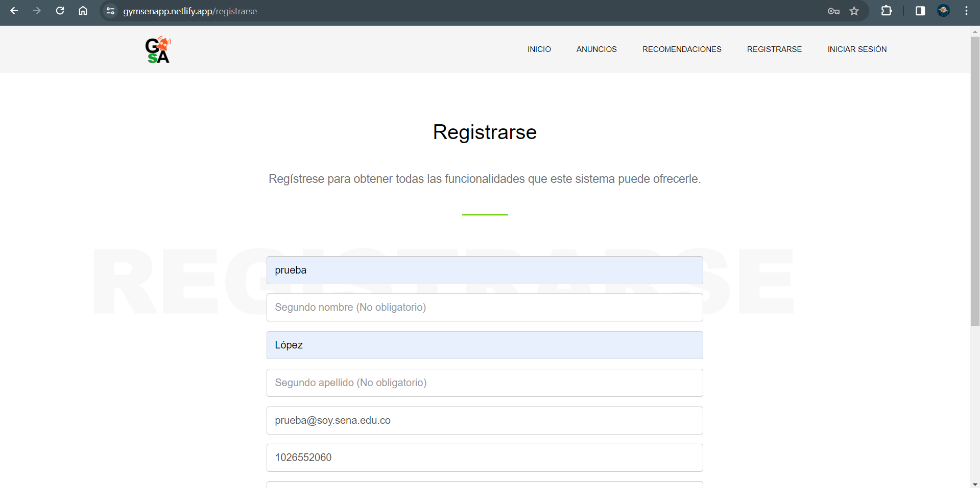
Con el fin de demostrar que el sistema como un todo funciona correctamente, a continuación, mostramos unas pequeñas pruebas del funcionamiento de los componentes unidos entre ellos.

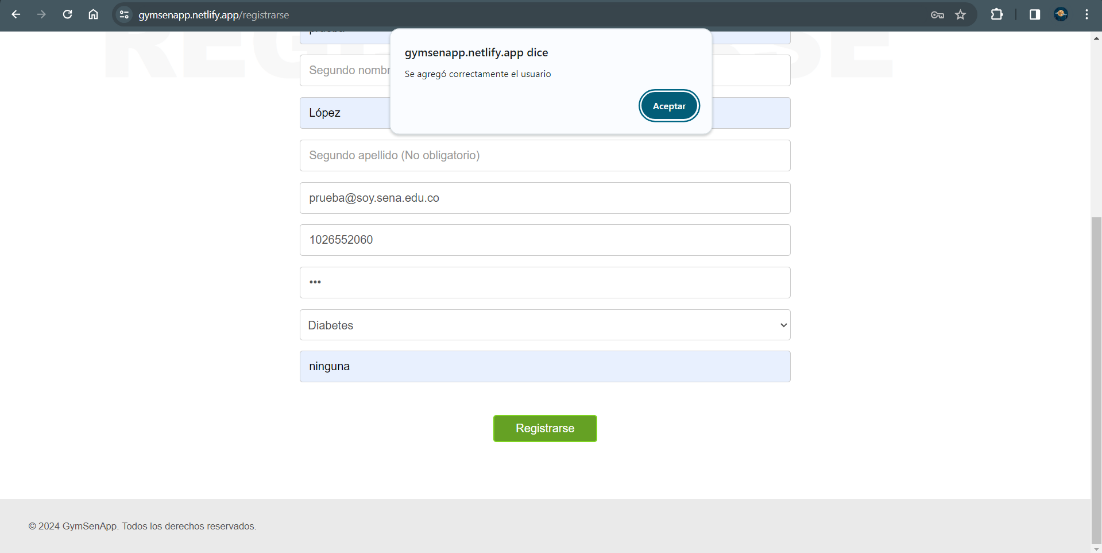
 Primero, tenemos la página inicial del sistema, esta demuestra que al menos el Frontend funciona:

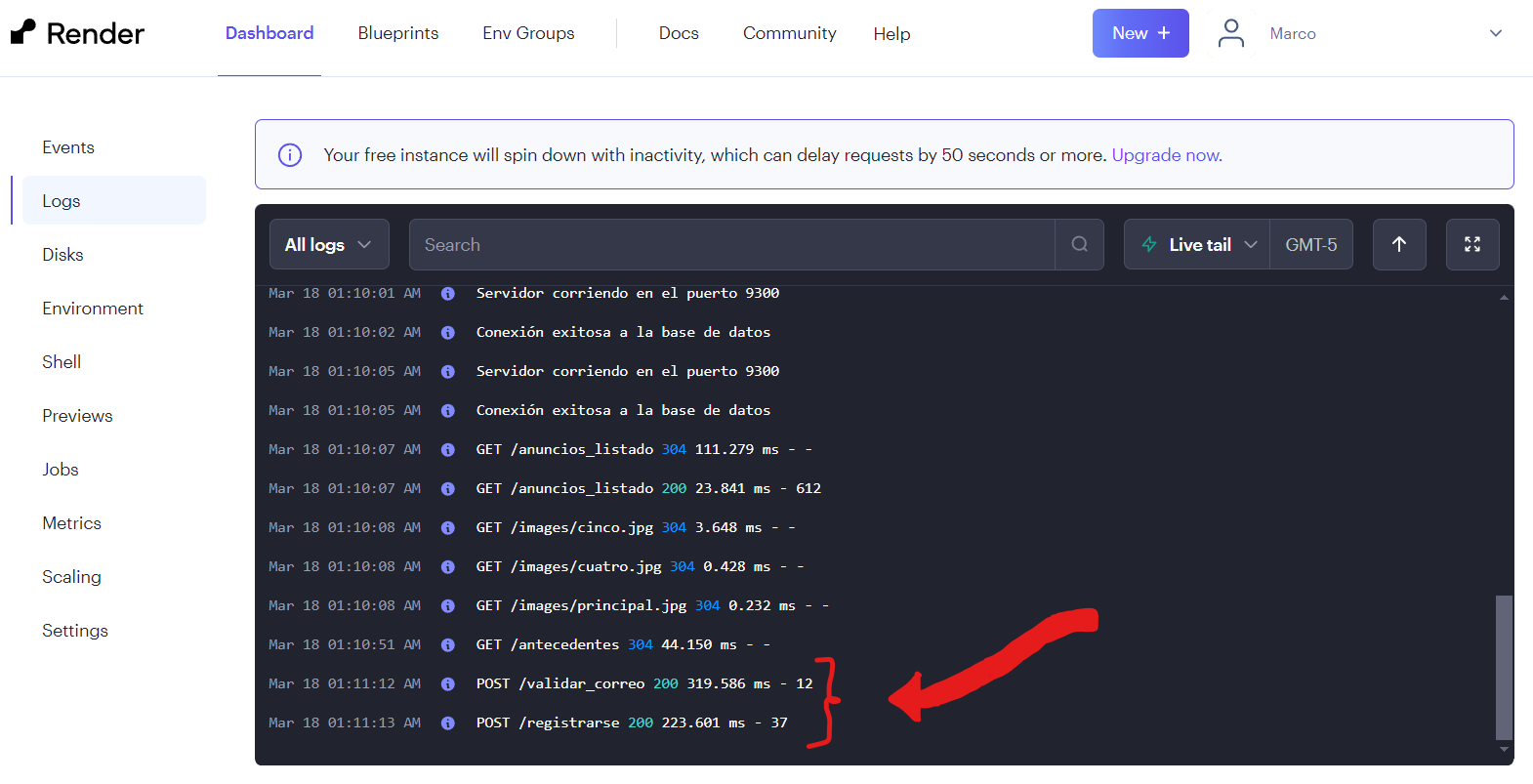
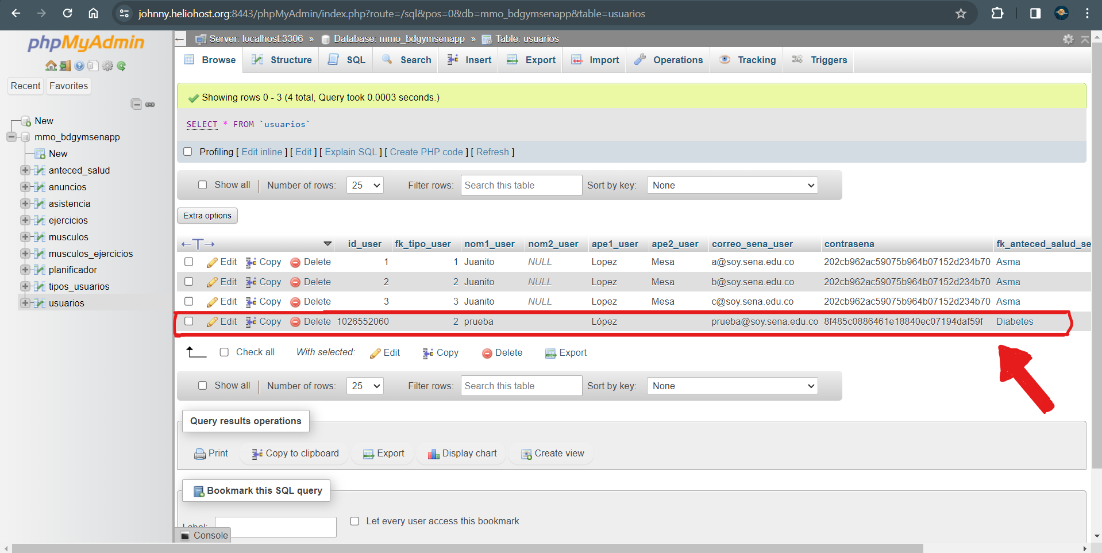
 Ahora vayamos a la sección de anuncios, en donde estos se cargan de la base de datos:

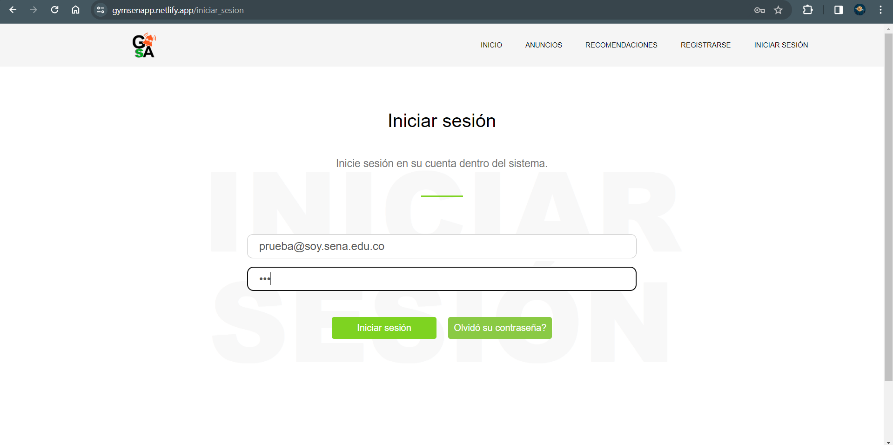
 Aquí apreciamos como se muestran. Si vemos en la terminal de render, en donde está corriendo la API vemos que efectivamente se hizo la petición para el cargue de los anuncios:

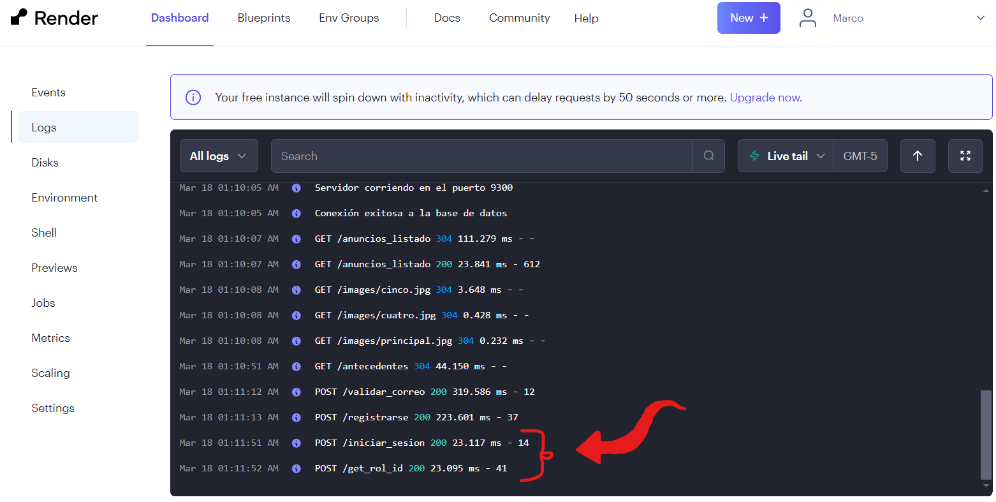
 Adicionalmente, en la base de datos podemos ver que los datos de los anuncios existen:

 Ahora, probemos el registro de usuarios. En el aplicativo nos dirigimos a la opción “registrarse” y llenamos nuestros datos:

 Le damos en el botón “registrarse” y nos saldrá el anuncio que confirma que se realizó efectivamente el registro:

 Si miramos la API y la base de datos confirmaremos el resultado anunciado por el Frontend:

 Hagamos una última prueba iniciando sesión dentro de la cuenta creada:

Miremos la API:

De esta forma corroboramos en este documento que el sistema funciona correctamente (de igual forma le hicimos las pruebas correspondientes a este sistema desplegado y funciona totalmente y según lo esperado).

El link del aplicativo desplegado es: <https://gymsenapp.netlify.app/>

Nota: La API de forma automática se desactiva si no está constantemente en uso, así que le pedimos el grande favor si nos avisa cuando lo vaya a probar para dejar funcionando la API para que corrobore el funcionamiento del sistema. Gracias.