

DOCUMENTACIÓN DE PLAN DE MIGRACIÓN Y RESPALDO DE LOS DATOS  
DEL SOFTWARE

AUTOR:

BRAYAN ALEXANDER PUENTES MARTINEZ

INSTRUCTOR:

ANDRES RUBIANO CUCARIAN

SENA

ANALISIS Y DESARROLLO DE SOFTWARE(ADSO)

CENTRO DE SERVICIOS FINANCIEROS

BOGOTÁ D.C. 14 DE MAYO DE 2024

## TABLA DE CONTENIDO

INTRODUCCION.....	4
DESCRIPCIÓN DEL SISTEMA .....	5
FUNCIONALIDADES PRINCIPALES.....	6
REGISTRO DE USUARIOS.....	6
INICIO DE SESIÓN SEGURO .....	6
CAMBIO DE CONTRASEÑA.....	6
REGISTRAR MEDIDAS CORPORALES:.....	6
CREACIÓN DE RUTINAS PERSONALIZADAS.....	6
ALMACENAMIENTO DE ENLACES DE VIDEOS .....	6
TECNOLOGÍAS UTILIZADAS .....	6
IDE.....	7
Visual Studio Code .....	7
FRONT-END .....	7
Html .....	7
Css.....	7
FRAMEWORK.....	8
React.....	8
BACKEND .....	8
Node .....	8
Express .....	8
Java Script .....	8
BASES DE DATOS.....	9
MySQL .....	9
PROCESO DE IMPLEMENTACIÓN.....	10
FASES DEL DESARROLLO.....	10
ANÁLISIS DE REQUISITOS .....	10
DISEÑO.....	10
DESARROLLO.....	10
PRUEBAS .....	10
DESPLIEGUE .....	10
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS .....	11
GESTIÓN DE PROYECTOS.....	11

CONTROL DE VERSIONES .....	11
PRUEBAS AUTOMATIZADAS .....	11
ANÁLISIS DE MODIFICACIÓN Y PROBLEMAS .....	11
ÁREAS DE MEJORA .....	12
OPTIMIZACIÓN DE RENDIMIENTO .....	12
MEJORA DE LA EXPERIENCIA DE USUARIO (UX) .....	14
ESCALABILIDAD .....	15
SEGURIDAD .....	17
REGISTRO DE PROBLEMAS .....	19
PASO A PASO DEL PLAN DE EJECUCIÓN.....	19
PASO 1: IDENTIFICACIÓN DEL ERROR.....	19
PASO 2: REGISTRO DEL PROBLEMA .....	19
PASO 3: CLASIFICACIÓN DEL PROBLEMA .....	19
PASO 4: ASIGNACIÓN DEL PROBLEMA .....	19
PASO 5: INVESTIGACIÓN Y RESOLUCIÓN .....	19
PASO 6: PRUEBAS.....	20
PASO 7: VERIFICACIÓN Y CIERRE .....	20
IMPLEMENTACIÓN DE LA MODIFICACIÓN.....	21
PROCESO DE IMPLEMENTACIÓN .....	21
IDENTIFICA LA RAZÓN DE LA MODIFICACIÓN: .....	21
EVALÚA LAS NECESIDADES DE LOS USUARIOS: .....	21
DEFINE LOS OBJETIVOS DE LA MODIFICACIÓN:.....	21
DISEÑA LOS CAMBIOS PROPUESTOS: .....	21
EVALÚA LA VIABILIDAD Y EL ALCANCE: .....	21
ESTABLECE UN PLAN DE IMPLEMENTACIÓN: .....	21
COMUNICA LOS CAMBIOS: .....	21
IMPLEMENTA LOS CAMBIOS .....	21
RECOPILA COMENTARIOS Y RETROALIMENTACIÓN: .....	21
REALIZA AJUSTES SI ES NECESARIO: .....	22
MIGRACIÓN.....	23
PASO A PASO PARA UN PLAN DE MIGRACION .....	23
EVALÚA LA RAZÓN DE LA MIGRACIÓN.....	23
COMPRENDE LOS REQUISITOS Y LIMITACIONES.....	23

SELECCIONA LA PLATAFORMA DE DESTINO.....	23
PLANIFICA LA ESTRUCTURA Y EL CONTENIDO.....	23
ELABORA UN PLAN DE MIGRACIÓN DETALLADO .....	23
ESTABLECE FECHAS Y PLAZOS.....	23
ASIGNA RECURSOS .....	23
COMUNICA EL PLAN DE MIGRACIÓN .....	23
REALIZA PRUEBAS Y VERIFICACIONES .....	24
EJECUTA LA MIGRACIÓN.....	24
REALIZA VERIFICACIONES POSTERIORES A LA MIGRACIÓN .....	24
RECOPILA COMENTARIOS Y REALIZA AJUSTES .....	24
TABLAS DE MIGRACION.....	25
PLAN PARA LA ELIMINACIÓN DE LA PÁGINA WEB .....	56
PASOS ASEGUIR PARA LA ELIMINACIÓN DE LA PAGINA WEB .....	56
1. EVALUACIÓN DE LA RAZÓN DE LA ELIMINACIÓN.....	56
2. COMUNICACIÓN CON LAS PARTES INTERESADAS .....	56
3. RESPALDO DE DATOS IMPORTANTES.....	56
4. SUSPENSIÓN DE NUEVAS SUSCRIPCIONES Y ACTUALIZACIONE .....	56
5. REDIRECCIONAMIENTO DE TRÁFICO .....	56
6. CANCELACIÓN DE SERVICIOS Y CONTRATOS .....	56
7. ELIMINACIÓN DE LA INFRAESTRUCTURA.....	57
8. NOTIFICACIÓN DE ELIMINACIÓN .....	57
9. VERIFICACIÓN Y CONFIRMACIÓN DE LA ELIMINACIÓN .....	57
10. ANÁLISIS Y DOCUMENTACIÓN .....	57
CONCLUSIONES.....	58

## INTRODUCCION

En el ámbito del desarrollo de software, la documentación del plan de migración y el respaldo de los datos son aspectos críticos para garantizar una transición suave

y segura entre diferentes plataformas o versiones del software. Esta documentación es esencial para minimizar el riesgo de pérdida de datos y asegurar la integridad y disponibilidad de la información durante todo el proceso de migración.

## **DESCRIPCIÓN DEL SISTEMA**

La página web hipertrophy fitness está diseñada para permitir a los usuarios almacenar enlaces de videos de ejercicios de entrenamiento con la intención de permitirle a personas interesadas en el deporte poder tener guías visuales de sus ejercicios favoritos. También permite que los puedan tener un historial de sus medidas corporales.

En el apartado visual la plataforma ofrece una amplia gama de funcionalidades destinadas a mejorar la experiencia del usuario y optimizar la gestión del contenido de fitness. A continuación, se detallan las características principales del sistema:

## **FUNCIONALIDADES PRINCIPALES**

### **REGISTRO DE USUARIOS**

Los usuarios pueden crear cuentas nuevas proporcionando información básica como nombre, documento, celular, contraseña y confirmar contraseña.

### **INICIO DE SESIÓN SEGURO**

Los usuarios pueden iniciar sesión de forma segura utilizando sus credenciales previamente registradas.

### **CAMBIO DE CONTRASEÑA**

Los usuarios podrán realizar el cambio de contraseña ingresando a la sección de olvido su contraseña que se encuentra en pagina iniciar sesión. Este cambio lo podrán realizar con los siguientes datos: documento, contraseña y cambiar contraseña.

### **REGISTRAR MEDIDAS CORPORALES:**

En este apartado el usuario podrá encontrar un formulario para ingresar sus medidas corporales, una sección donde ver su historial de medidas registradas posteriormente y una parte donde podrá ver su primera medida ingresada con el fin de que el usuario pueda tener una motivación sobre su evolución en el mundo del ejercicio

### **CREACIÓN DE RUTINAS PERSONALIZADAS**

Los usuarios pueden crear rutinas de entrenamiento personalizadas, seleccionando ejercicios específicos que previamente guardaron en sus perfiles.

### **ALMACENAMIENTO DE ENLACES DE VIDEOS**

Los usuarios pueden agregar enlaces de videos de ejercicios a sus rutinas personalizadas para acceder fácilmente a ellos en el futuro.

## **TECNOLOGÍAS UTILIZADAS**

## **IDE**

### **Visual Studio Code**

Es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C#, Java, Python, PHP y Go) y tiempos de ejecución (como .NET y Unity). Además, Visual Studio Code le permite ampliar a los usuarios su capacidad a través de extensiones. Las extensiones de Visual Studio Code pueden agregar más características a la experiencia general. (Microsoft, s.f.).

## **FRONT-END**

### **Html**

(Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Donde "Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. También, HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, entre otros. (Mozilla, s.f.).

### **Css**

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). También, CSS es utilizado para diseñar y dar estilo a las páginas web, por ejemplo, alterando la fuente, color, tamaño y espaciado del contenido, dividirlo en múltiples columnas o agregar animaciones y otras características decorativas. Este módulo proporciona un suave comienzo hacia el dominio de CSS con los conceptos básicos acerca de su funcionamiento, la sintaxis y la manera en que el usuario puede comenzar a utilizarlo para agregar estilos al HTML. (Mozilla, s.f.).

## **FRAMEWORK**

### **React**

ReactJS es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web. Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.

Es importante señalar que ReactJS no es un framework de JavaScript. Esto porque sólo es responsable de renderizar los componentes de la capa de vista de una aplicación. React es una alternativa a frameworks como Angular y Vue, que permiten crear funciones complejas. (Hostinger, s.f.).

## **BACKEND**

### **Node**

Node.js es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real. (Kinsta, s.f.).

### **Express**

Express.js es el framework backend más popular para Node.js, y es una parte extensa del ecosistema JavaScript. Está diseñado para construir aplicaciones web de una sola página, multipágina e híbridas, también se ha convertido en el estándar para desarrollar aplicaciones backend con Node.js, y es la parte backend de algo conocido como la pila MEVN. (Kinsta, s.f.).

### **Java Script**

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc. (Mozilla, s.f.).



## **BASES DE DATOS**

### **MySQL**

MySQL es un sistema de administración de bases de datos relacionales. Es un software de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. (HubSpot, s.f.).

## **PROCESO DE IMPLEMENTACIÓN**

El proceso de implementación del sistema Hypertrophy fitness se dividió en varias etapas claramente definidas, cada una de las cuales se ejecutó siguiendo las mejores prácticas de desarrollo de software.

### **FASES DEL DESARROLLO**

#### **ANÁLISIS DE REQUISITOS**

Se llevó a cabo un análisis exhaustivo de los requisitos del sistema, en el que se identificaron y documentaron los objetivos del proyecto, las características principales y los casos de uso.

#### **DISEÑO**

Se elaboraron diseños detallados de la arquitectura del sistema, la interfaz de usuario y la estructura de la base de datos, teniendo en cuenta los requisitos del usuario y los estándares de diseño.

#### **DESARROLLO**

Se procedió con la implementación de las funcionalidades del sistema utilizando un enfoque basado en componentes y módulos reutilizables, lo que facilita la escalabilidad y el mantenimiento del código.

#### **PRUEBAS**

Se realizaron pruebas exhaustivas en todas las etapas del desarrollo, incluyendo pruebas unitarias, de integración y de aceptación, para garantizar la calidad y la fiabilidad del software.

#### **DESPLIEGUE**

El sistema se desplegó gradualmente en entornos de desarrollo, pruebas y producción, utilizando herramientas de automatización y gestión de configuraciones para facilitar el proceso.

### **METODOLOGÍA ÁGIL**

El desarrollo del sistema siguió una metodología ágil, específicamente el marco de trabajo Scrum, que se caracteriza por ciclos de desarrollo iterativos y entregas incrementales. Todos los roles de este trabajo fueron desempeñados por un solo aprendiz debido a el retiro del programa de los compañeros del grupo GAES formado en las primeras semanas del programa

## **HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS**

### **GESTIÓN DE PROYECTOS**

Se utilizaron herramientas de gestión de proyectos como Trello para planificar, organizar y seguir el progreso del desarrollo luego se pasó a trabajar con notas en un archivo de Word debido la metodología del cambio a un solo estudiante.

### **CONTROL DE VERSIONES**

Se utilizó Git como sistema de control de versiones para gestionar el código fuente además de servir por medio de los commit como guía de avances y cumplimientos de objetivos.

### **PRUEBAS AUTOMATIZADAS**

Se implementaron pruebas automatizadas utilizando herramientas como Postman y Jmeter para garantizar el GET, POST, PUT, DELETE y pruebas de desempeño al conectar múltiples usuarios al mismo tiempo.

.

## **ANÁLISIS DE MODIFICACIÓN Y PROBLEMAS**

Durante el desarrollo y la implementación de la página web, se identificaron varias áreas potenciales para futuras modificaciones y mejoras:

## ÁREAS DE MEJORA

### OPTIMIZACIÓN DE RENDIMIENTO

AREAS DE MEJORA				
OPTIMIZACIÓN DE RENDIMIENTO				
ID	NOMBRE DE LA MEJORA	DESCRIPCIÓN DE LA MEJORA	ESTADO	RESPONSABLE
01	Memorización de resultados de solicitud HTTP	Puedes usar bibliotecas como react-query para almacenar en caché los resultados de las solicitudes HTTP y evitar solicitudes innecesarias al servidor.	Pendiente	Brayan Alexander Puentes Martinez
02	Chunking de JavaScript y CSS	Utiliza herramientas como Webpack para dividir tu código en fragmentos más pequeños (chunks) y cargarlos de manera diferida cuando sea necesario.	Pendiente	Brayan Alexander Puentes Martinez
03	Optimización de imágenes	Utiliza herramientas como ImageMagick o servicios en la nube como Cloudinary para comprimir y optimizar las imágenes antes de cargarlas en tu aplicación.	Pendiente	Brayan Alexander Puentes Martinez

04	Prefetching de rutas	Puedes usar la función de prefetching de React Router para precargar los recursos necesarios para las rutas a las que es probable que el usuario vaya a navegar.	Pendiente	Brayan Alexander Puentes Martinez
----	----------------------	--	-----------	-----------------------------------

## MEJORA DE LA EXPERIENCIA DE USUARIO (UX)

AREAS DE MEJORA				
EXPERIENCIA DE USUARIO UX				
ID	NOMBRE DE LA MEJORA	DESCRIPCIÓN DE LA MEJORA	ESTADO	RESPONSABLE
01	Animaciones de carga	Utiliza animaciones CSS o bibliotecas como Lottie para mostrar animaciones de carga mientras se espera la respuesta del servidor.	Pendiente	Brayan Alexander Puentes Martinez
02	Indicadores de progreso	Muestra una barra de progreso o un indicador circular para informar al usuario sobre el progreso de la acción en curso.	Pendiente	Brayan Alexander Puentes Martinez
03	Mensajes de carga	Muestra mensajes de carga específicos para diferentes acciones, como "Cargando datos" o "Enviando formulario".	Pendiente	Brayan Alexander Puentes Martinez
04	Bloqueo de pantalla	Implementa un bloqueo de pantalla temporal para evitar que el usuario interactúe con la interfaz mientras se realiza una acción que requiere tiempo.	Pendiente	Brayan Alexander Puentes Martinez

## ESCALABILIDAD

AREAS DE MEJORA				
ESCALABILIDAD				
ID	NOMBRE DE LA MEJORA	DESCRIPCIÓN DE LA MEJORA	ESTADO	RESPONSABLE
01	Implementación de una capa de caché	Configura una capa de caché distribuida, como Redis o Memcached, para almacenar en caché resultados de consultas a la base de datos, respuestas de API u otros datos frecuentemente solicitados.	Pendiente	Brayan Alexander Puentes Martinez
02	Reducción de la carga en la base de datos	Almacenar en caché datos comúnmente utilizados reduce la carga en la base de datos, lo que ayuda a mejorar el rendimiento y la capacidad de respuesta de tu aplicación.	Pendiente	Brayan Alexander Puentes Martinez
03	Escalamiento horizontal de la capa de caché	Diseña la arquitectura de tu capa de caché para que también pueda escalar horizontalmente, permitiendo agregar más nodos según sea necesario para manejar mayores volúmenes de datos y solicitudes.	Pendiente	Brayan Alexander Puentes Martinez

04	Configuración de políticas de expiración	Establece políticas de expiración para los datos en caché para garantizar que la información almacenada en la caché se mantenga actualizada y se eliminen los datos obsoletos automáticamente.	Pendiente	Brayan Alexander Puentes Martinez
----	--	--	-----------	-----------------------------------



## SEGURIDAD

AREAS DE MEJORA				
SEGURIDAD				
ID	NOMBRE DE LA MEJORA	DESCRIPCIÓN DE LA MEJORA	ESTADO	RESPONSABLE
01	Análisis de vulnerabilidades	Realiza análisis de seguridad periódicos en tu aplicación para identificar y abordar posibles vulnerabilidades, siguiendo las recomendaciones de la Fundación OWASP.	Pendiente	Brayan Alexander Puentes Martinez
02	Prevención de ataques de secuencias de comandos entre sitios (XSS)	Implementa medidas de seguridad, como la sanitización de entradas del usuario y el uso de encabezados de seguridad HTTP, para prevenir los ataques XSS que podrían comprometer la seguridad de tu aplicación.	Pendiente	Brayan Alexander Puentes Martinez
03	Prevención de ataques de falsificación de solicitudes entre sitios (CSRF)	Utiliza tokens CSRF y verifica los encabezados de origen en las solicitudes para prevenir los ataques CSRF que podrían permitir a un atacante realizar acciones no autorizadas en nombre del	Pendiente	Brayan Alexander Puentes Martinez

		usuario.		
04	Actualizaciones regulares y parches de seguridad	Mantén tu aplicación y todas sus dependencias actualizadas con los últimos parches de seguridad para mitigar los riesgos de seguridad conocidos y evitar posibles brechas de seguridad.	Pendiente	Brayan Alexander Puentes Martinez

## **REGISTRO DE PROBLEMAS**

Se estableció un sistema de seguimiento de problemas para registrar y priorizar los problemas identificados durante el desarrollo y la operación del sistema. Los problemas se clasificaron según su gravedad y se asignaron al único miembro del grupo.

### **PASO A PASO DEL PLAN DE EJECUCIÓN**

#### **PASO 1: IDENTIFICACIÓN DEL ERROR**

- El error puede ser identificado por diversos medios, como comentarios de usuarios, monitoreo de métricas de rendimiento, pruebas de calidad, o reportes internos del equipo.

#### **PASO 2: REGISTRO DEL PROBLEMA**

- El miembro del equipo que identificó el error debe registrar el problema en el sistema de seguimiento de problemas de la página web.
- Debería incluir detalles específicos sobre el error, como una descripción del problema, la página o funcionalidad afectada, el contexto en el que se produjo, y cualquier otra información relevante.

#### **PASO 3: CLASIFICACIÓN DEL PROBLEMA**

- El problema se clasifica según su gravedad, utilizando una escala predefinida (por ejemplo, bajo, medio, alto, crítico).
- La gravedad se determina según el impacto del error en la funcionalidad de la página web y la experiencia del usuario.

#### **PASO 4: ASIGNACIÓN DEL PROBLEMA**

- El problema se asigna a un miembro específico del equipo responsable de su resolución.
- La asignación puede basarse en la especialización del miembro del equipo o en su disponibilidad.

#### **PASO 5: INVESTIGACIÓN Y RESOLUCIÓN**

- El miembro del equipo asignado investiga el problema para comprender su causa raíz.
- Se desarrolla una solución para corregir el error y se implementa en el código de la página web.

## **PASO 6: PRUEBAS**

- Después de implementar la solución, se realizan pruebas exhaustivas para **verificar que el error se haya corregido correctamente y que no haya** introducido nuevas fallas en la página web.

## **PASO 7: VERIFICACIÓN Y CIERRE**

- Una vez que se confirma que el error ha sido corregido y que la página web funciona correctamente, se cierra el registro del problema en el sistema de seguimiento de problemas.
- Se puede agregar información adicional, como notas sobre la solución implementada y las lecciones aprendidas durante el proceso de resolución. y necesito lo siguiente

## **IMPLEMENTACIÓN DE LA MODIFICACIÓN**

### **PROCESO DE IMPLEMENTACIÓN**

Paso a Paso para Planificar una Modificación en una Página Web

#### **IDENTIFICA LA RAZÓN DE LA MODIFICACIÓN:**

¿Qué aspecto de la página web quieres cambiar y por qué? Por ejemplo, ¿quieres mejorar la experiencia del usuario, añadir nuevas características, actualizar el diseño, etc.?

#### **EVALÚA LAS NECESIDADES DE LOS USUARIOS:**

Considera las opiniones y comentarios de los usuarios actuales para entender qué cambios serían más beneficiosos para ellos.

#### **DEFINE LOS OBJETIVOS DE LA MODIFICACIÓN:**

Establece objetivos claros y medibles que desees alcanzar con la modificación. Por ejemplo, aumentar el tiempo promedio de permanencia en la página, mejorar la tasa de conversión, etc.

#### **DISEÑA LOS CAMBIOS PROPUESTOS:**

Visualiza cómo se verán los cambios en la página web. Puedes hacer bocetos o utilizar herramientas de diseño para crear un prototipo.

#### **EVALÚA LA VIABILIDAD Y EL ALCANCE:**

Determina si los cambios propuestos son factibles dentro de tu marco de tiempo y recursos disponibles.

#### **ESTABLECE UN PLAN DE IMPLEMENTACIÓN:**

Define un cronograma detallado que incluya todas las etapas del proceso, desde el diseño hasta la implementación y la prueba.

#### **COMUNICA LOS CAMBIOS:**

Informa a los interesados, como el equipo de desarrollo, el equipo de diseño y los usuarios, sobre los cambios planificados y cómo les afectarán.

#### **IMPLEMENTA LOS CAMBIOS**

Lleva a cabo los cambios según lo planificado, asegurándote de seguir las mejores prácticas de desarrollo web y de realizar pruebas exhaustivas antes de lanzar los cambios en producción.

#### **RECOPILA COMENTARIOS Y RETROALIMENTACIÓN:**

Después de implementar los cambios, recopila comentarios de los usuarios y evalúa su satisfacción y la efectividad de los cambios en relación con los objetivos establecidos.

**REALIZA AJUSTES SI ES NECESARIO:**

Basándote en los comentarios recibidos, realiza ajustes adicionales en la página web para mejorar aún más la experiencia del usuario y alcanzar los objetivos establecidos.

# **MIGRACIÓN**

## **PASO A PASO PARA UN PLAN DE MIGRACION**

### **EVALÚA LA RAZÓN DE LA MIGRACIÓN**

- ¿Por qué necesitas migrar la página web? Identifica los problemas actuales o los requisitos nuevos que están impulsando la necesidad de migrar.

### **COMPRENDE LOS REQUISITOS Y LIMITACIONES**

- Entiende qué requisitos técnicos y funcionales debe cumplir la nueva plataforma o infraestructura. Considera también cualquier limitación en términos de tiempo, recursos y presupuesto.

### **SELECCIONA LA PLATAFORMA DE DESTINO**

- Investiga y elige la plataforma o sistema de alojamiento al que migrarás tu página web. Considera factores como la escalabilidad, la seguridad, la facilidad de uso y el costo.

### **PLANIFICA LA ESTRUCTURA Y EL CONTENIDO**

- Diseña la nueva estructura de la página web y decide qué contenido se migrará y cómo se organizará en la nueva plataforma.

### **ELABORA UN PLAN DE MIGRACIÓN DETALLADO**

- Crea un plan detallado que incluya todas las etapas del proceso de migración, desde la preparación hasta la implementación y la verificación.

### **ESTABLECE FECHAS Y PLAZOS**

- Define fechas límite claras para cada etapa del proceso de migración. Asegúrate de ser realista en cuanto a los plazos, considerando cualquier posible retraso.

### **ASIGNA RECURSOS**

- Asigna el personal adecuado y los recursos necesarios para llevar a cabo la migración de manera eficiente y efectiva. Esto puede incluir desarrolladores, diseñadores, personal de soporte técnico, etc.

### **COMUNICA EL PLAN DE MIGRACIÓN**

- Informa a todas las partes interesadas sobre el plan de migración, incluido el equipo interno, los clientes y los proveedores de servicios externos.

## **REALIZA PRUEBAS Y VERIFICACIONES**

- Antes de la migración real, realiza pruebas exhaustivas en un entorno de prueba para asegurarte de que todo funcione correctamente. Verifica la funcionalidad, el rendimiento y la seguridad de la nueva plataforma.

## **EJECUTA LA MIGRACIÓN**

- Lleva a cabo la migración según el plan establecido, asegurándote de seguir los pasos cuidadosamente y de mantener una comunicación clara con todo el equipo involucrado.

## **REALIZA VERIFICACIONES POSTERIORES A LA MIGRACIÓN**

- Después de completar la migración, verifica que todo esté funcionando correctamente en la nueva plataforma. Soluciona cualquier problema que pueda surgir.

## **RECOPILA COMENTARIOS Y REALIZA AJUSTES**

- Solicita retroalimentación de los usuarios y realiza ajustes adicionales según sea necesario para garantizar una transición suave y una experiencia positiva para los usuarios.



## TABLAS DE MIGRACION

TABLA DE PLAN DE MIGRACIÓN			
Nombre De La Página Web	Hypertrophy fitness	Versión	1.0
descripción De La Página Web	Pagina web para almacenamiento de videos de rutinas de ejercicio y creación de rutinas de ejerici.o		
Categoría Del Sistema	Pagina web	Autor	Brayan Alexander Puentes Martinez
Lenguajes de programación utilizados	HTML CSS JAVASCRIPT	Arquitectura	Modelo vista controlador
Documentación de la migración	<ul style="list-style-type: none"><li>• Home</li><li>• Acerca de nosotros.</li><li>• Iniciar sesión.</li><li>• Registrarse.</li><li>• Cambio de contraseña.</li><li>• Consejos.</li><li>• Medidas corporales.</li><li>• Crear rutina.</li><li>• Plan de entrenamiento</li></ul>		
descripción de paginas web			
REQUISITOS MINIMOS DE HARDWARE Y SOFTWARE			
HARDWARE			
<ul style="list-style-type: none"><li>• Procesador: Intel Core i3 de cualquier generación o equivalente de otro fabricante.</li><li>• Memoria RAM: 4 GB.</li><li>• Almacenamiento: Disco duro con al menos 128 GB de espacio disponible.</li><li>• Conectividad: Conexión a Internet (por cable o inalámbrica).</li></ul> <p>Pantalla: Cualquier pantalla que permita visualizar contenido web de forma legible.</p>			
SOFTWARE			
<ul style="list-style-type: none"><li>• Sistema operativo: Windows 7 o superior, macOS 10.12 o superior, o una distribución de Linux compatible.</li></ul> <p>Navegador web: Cualquier navegador moderno y actualizado, como Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, entre otros.</p>			
PLANIFICACIÓN DE LA MIGRACIÓN		SE REALIZO	

	<b>SI</b>	<b>NO</b>
¿Se ha definido un plan detallado para la migración de la base de datos?		
Se ha establecido un calendario para la migración, teniendo en cuenta los tiempos de inactividad mínimos requeridos?		
<b>REQUISITOS DEL SISTEMA DESTINO</b>	<b>SI</b>	<b>NO</b>
¿El sistema destino cumple con los requisitos de hardware y software especificados en la documentación?		
<b>COMPATIBILIDAD CON LA BASE DE DATOS</b>	<b>SI</b>	<b>NO</b>
¿El sistema destino es compatible con el tipo de base de datos utilizadas por la pagina web?		
<b>COPIAS DE SEGURIDAD</b>	<b>SI</b>	<b>NO</b>
¿Se han realizado copias de seguridad completas y válidas de la base de datos actuales.		
¿Se han probado las copias de seguridad para verificar su integridad y accesibilidad?		
<b>TRANSFERENCIA DE DATOS</b>	<b>SI</b>	<b>NO</b>
¿Se ha establecido un método seguro y eficiente para transferir los datos de la base de datos actual al sistema destino?		
¿Se ha verificado la integridad de los datos transferidos después de la migración?		
<b>ACTUALIZACIÓN DE CONFIGURACIONES</b>	<b>SI</b>	<b>NO</b>
¿Se han actualizado todas las configuraciones relacionadas con la base de datos en el sistema destino nombre de usuario, contraseña, dirección del servidor.?		
<b>VERIFICACIÓN POST-MIGRACIÓN</b>	<b>SI</b>	<b>NO</b>
¿Se ha verificado que todas las funciones y características de la página web funcionen correctamente después de la migración?		

¿Se han realizado pruebas exhaustivas para garantizar que no haya pérdida de datos ni errores de funcionamiento?		
<b>CAPACITACIÓN Y DOCUMENTACIÓN</b>	<b>SI</b>	<b>NO</b>
¿Se realizo capacitación de la página web migrado?		
¿Se realizo la documentación pertinente de la pagina web migrada?		

# MIGRACION PLATAFORMA

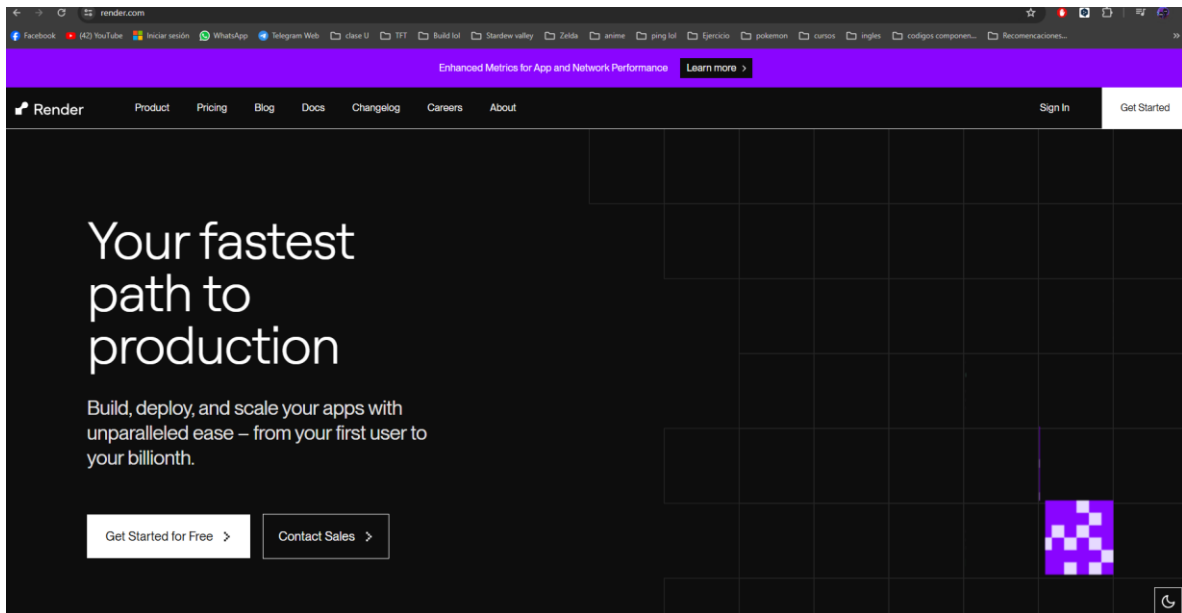
## HOSTING

Para realizar el hosting, es decir, cambiar del modo desarrollador a producción, donde las personas puedan ver la página web mediante un enlace, link o URL, se utilizaron dos herramientas. A continuación, se detalla el proceso paso a paso utilizando Render para desplegar tanto el frontend como el backend.

## RENDER

### FRONT END

Inicialmente se entra en la página principal de render



Posteriormente, se realiza un registro en Render, donde en este caso se seleccionó la opción de GitHub, ya que allí se subió todo el proyecto desarrollado en Visual Studio Code.

### Create an account



Email

your@email.com

▲ Required

Password

correct horse battery staple

▲ Required

Create Account

By signing up you agree to our [terms of service](#).

Already have an account? [Sign in](#)

This site is protected by Cloudflare, its Privacy Policy and Terms of Service apply.


"Building on Render has been a superpower for Watershed. It lets us focus on what matters most — delighting our customers, rather than managing our infrastructure.


Our engineers can ship features to customers in hours or days instead of weeks or months. Every startup should start on Render."

AVI ITSKOVICH, CO-FOUNDER AT WATERSHED




Se selecciona la cuenta para seguir haciendo el registro correspondiente en render.


 Iniciar sesión con Google



## Selecciona una cuenta

para ir a [Render](#)

**Brayan Alexander Puentes Martinez**  
alepues1@gmail.com

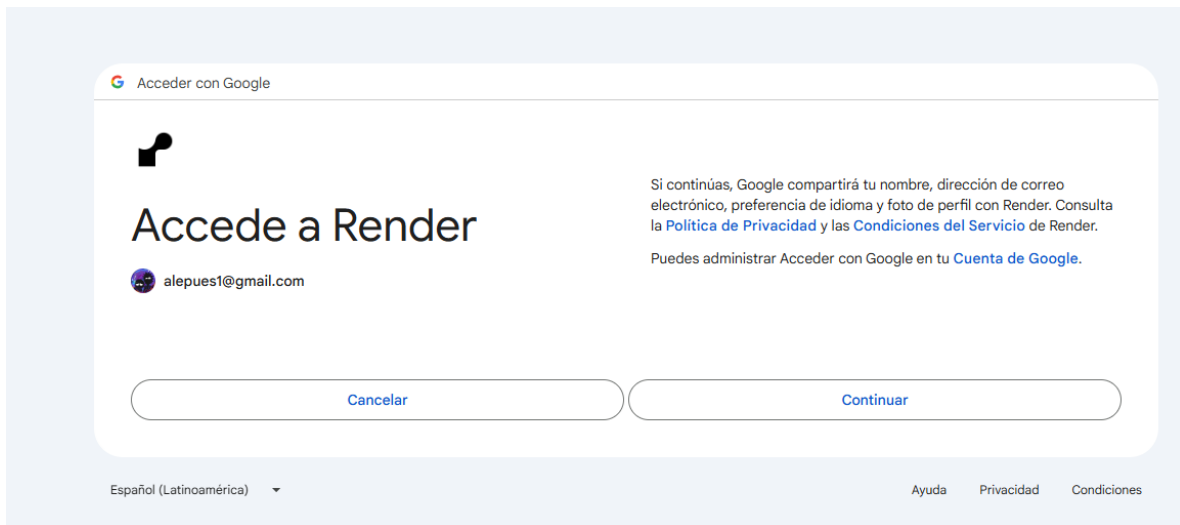
 [Usar otra cuenta](#)

Para continuar, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu foto de perfil con Render. Antes de usar esta aplicación, puedes leer la [política de privacidad](#) y los [términos del servicio](#) de Render.

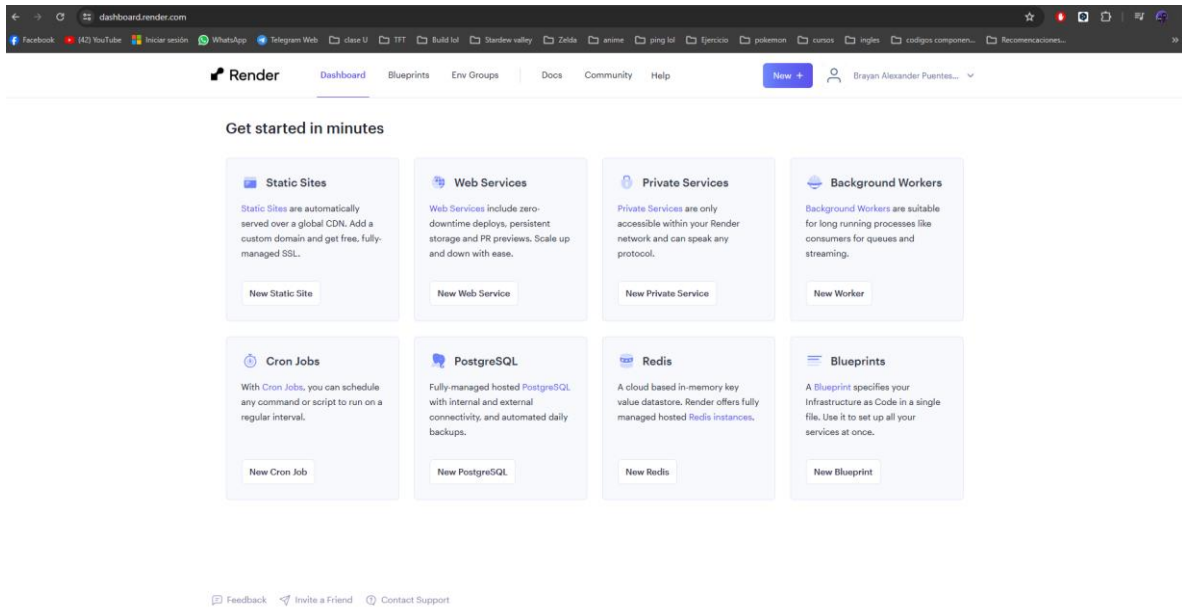
Español (España) ▼

[Ayuda](#) [Privacidad](#) [Términos](#)

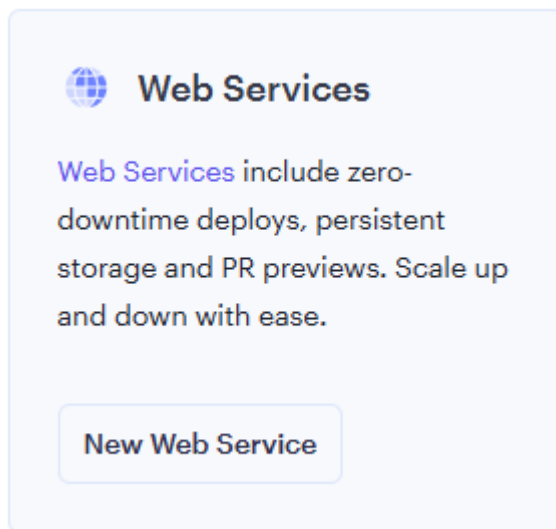
Se aceptan los termino y condiciones de render para poder usarlo sin problemas.



Se inicia sesión de forma exitosa en render y se evidencia en la imagen la pagina de inicio donde aparecen varias opciones.



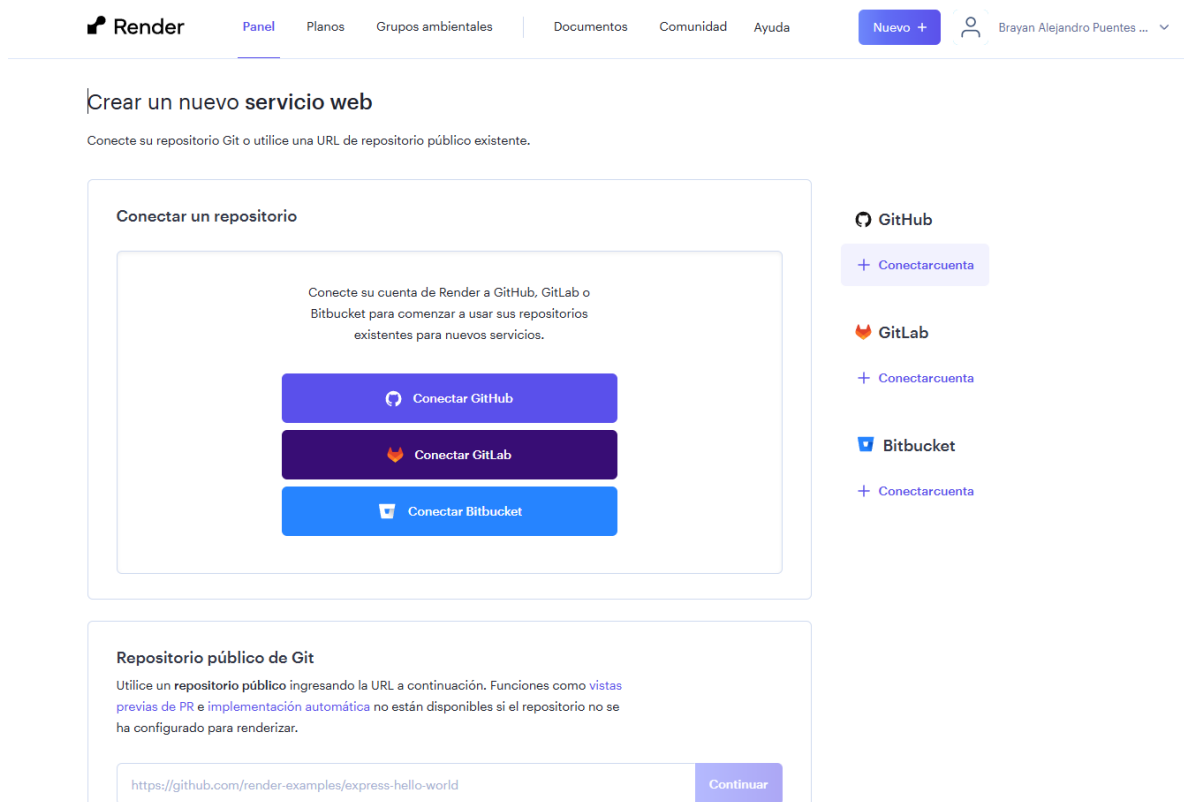
En este caso, se selecciona web services porque ofrece ventajas significativas en términos de tiempo, costo, escalabilidad y eficiencia, especialmente para proyectos grandes o complejos. Además, el proyecto es una página web, por esa razón se eligió esta opción.



En este caso se desea implementar el servicio web mediante GitHub ya que previamente se subió el proyecto a un repositorio.



Luego, se debe crear un nuevo servicio web, seleccionando la opción "Conectar Cuenta" en el apartado de GitHub.





Como se evidencia en la imagen se le da permiso a render de poder acceder al GitHub








Render by **Render** would like permission to:

 Verify your GitHub identity (AlexanderxB)

 Know which resources you can access

 Act on your behalf  
 [Learn more](#)

#### Resources on your account


 **Email addresses** (read)  
View your email addresses

[Learn more about Render](#)


Cancel

Authorize Render

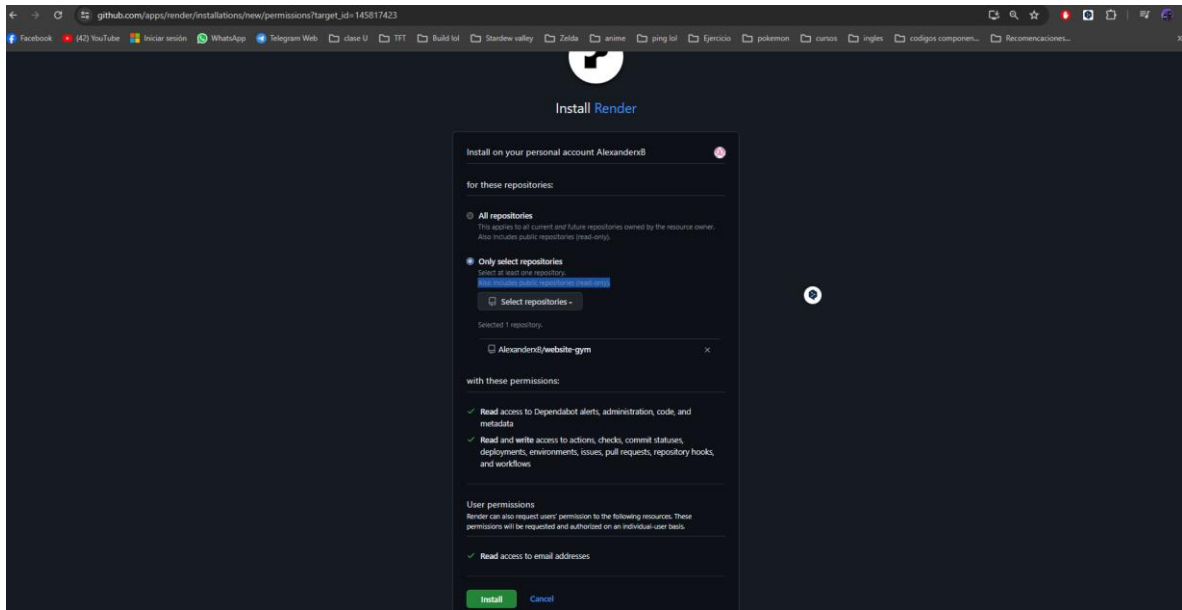
Authorizing will redirect to  
<https://dashboard.render.com>

 Not owned or  
operated by GitHub

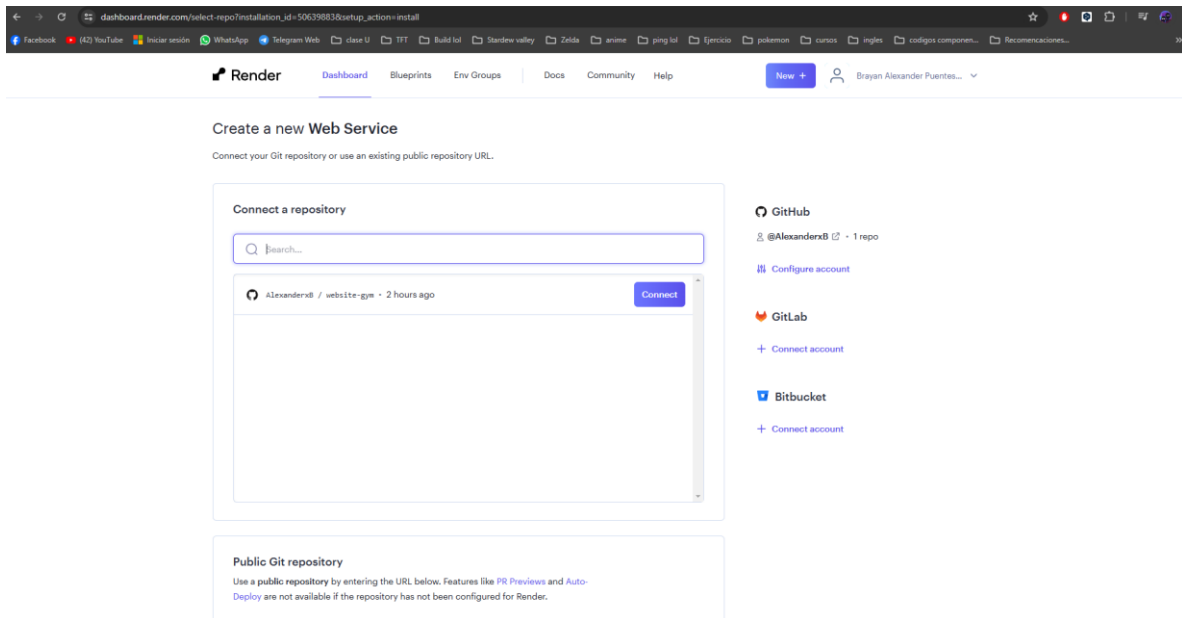
 Created 6 years ago

 More than 1K  
GitHub users

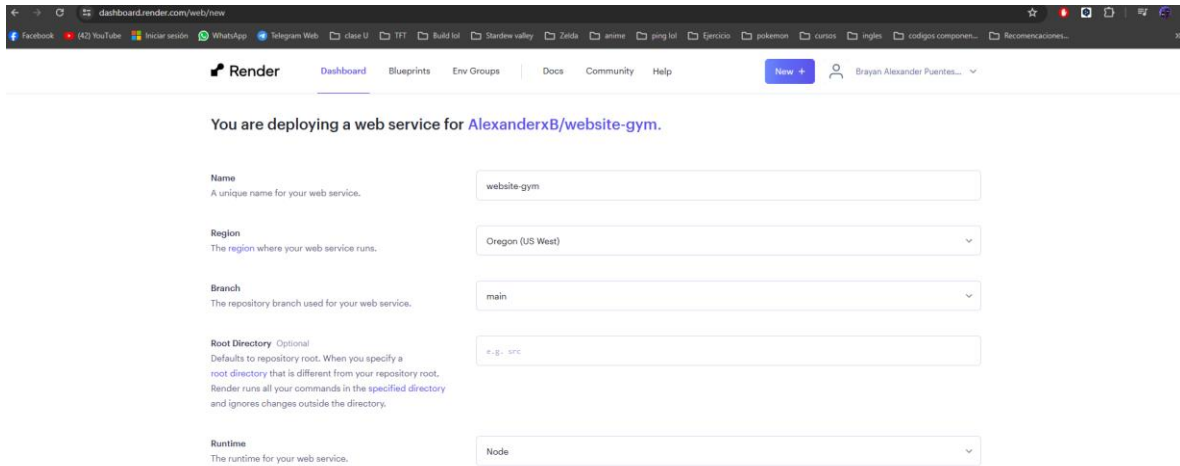
Se le da permiso únicamente para que acceda render al repositorio que dice website-gym



Se ha creado con éxito la creación de un web service



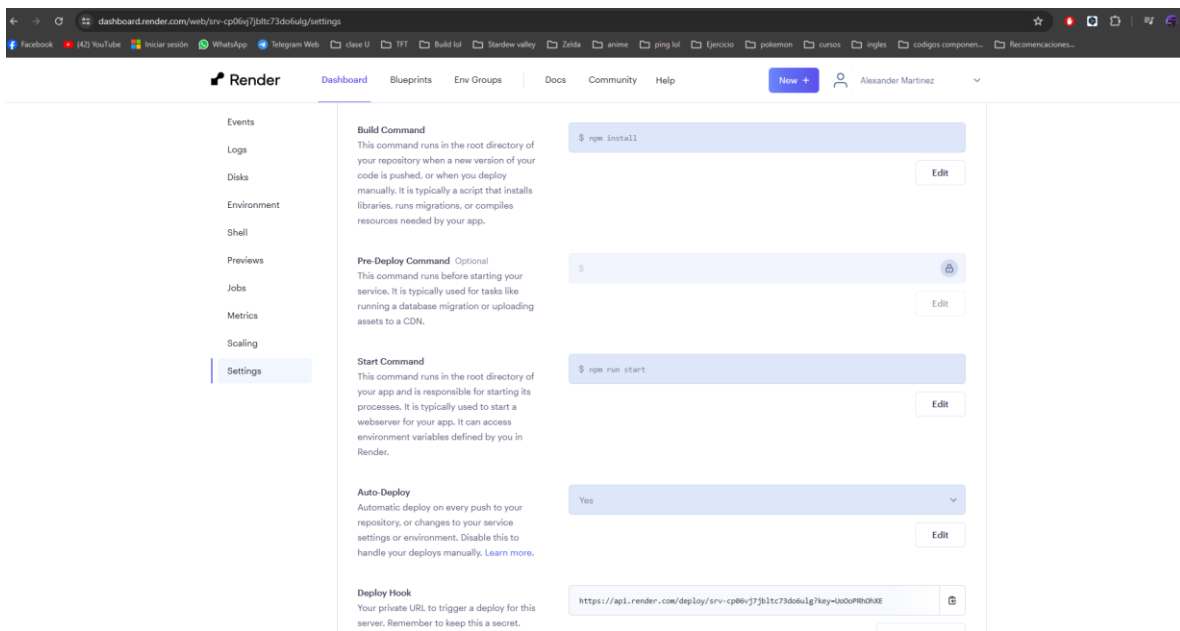
Al dar click al web service creado, se empieza a realizar la configuración de este para poder desplegar el front end del proyecto “Hypertrophy Fitness”



The screenshot shows the Render dashboard with the following configuration for a new web service:

- Name:** website-gym
- Region:** Oregon (US West)
- Branch:** main
- Root Directory:** Optional (set to /src)
- Runtime:** Node

Lo primordial es seleccionar la opción de Node. En el campo "build command", se ingresa el comando npm install, y en el campo "start command", se ingresa el comando npm run start. Además, se elige la opción "free" para utilizar Render de forma gratuita para realizar el despliegue.



The screenshot shows the Render dashboard with the following settings for a web service:

- Build Command:** npm install
- Pre-Deploy Command:** Optional (set to /src)
- Start Command:** npm run start
- Auto-Deploy:** Yes
- Deploy Hook:** https://api.render.com/deploy/srv-cp86v37jbtcc73dolu1g7hey-4a0p8h0k0k0k

Instance Type

**For hobby projects**

<b>Free</b> \$0 / month	512 MB (RAM) 0.1 CPU	<b>Upgrade to enable more features</b> Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.
<b>Starter</b> \$7 / month	512 MB (RAM) 0.5 CPU	

**For professional use**  
For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

<b>Pro</b> \$85 / month	4 GB (RAM) 2 CPU	<b>Standard</b> \$25 / month	2 GB (RAM) 1 CPU
<b>Pro Plus</b> \$175 / month	8 GB (RAM) 4 CPU	<b>Pro Max</b> \$225 / month	16 GB (RAM) 4 CPU
<b>Pro Ultra</b> \$450 / month	32 GB (RAM) 8 CPU		

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Finalmente, se da click sobre el boton que dice “Create web service”

Render Dashboard

Environment Variables

NAME\_OF\_VARIABLE value

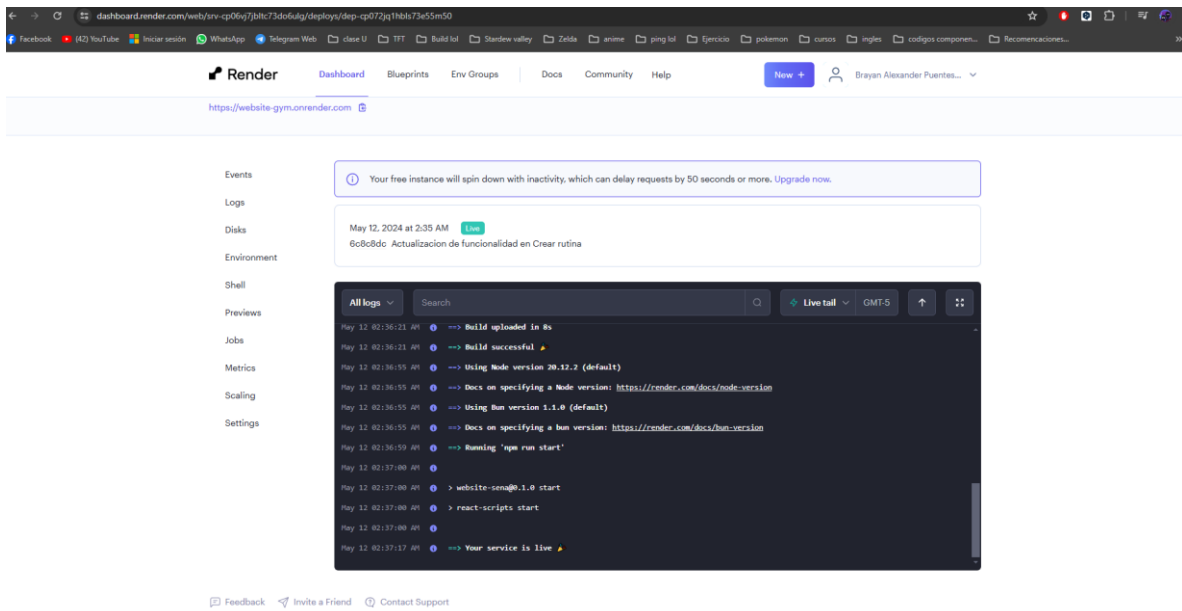
+ Add Environment Variable Add from .env

Advanced

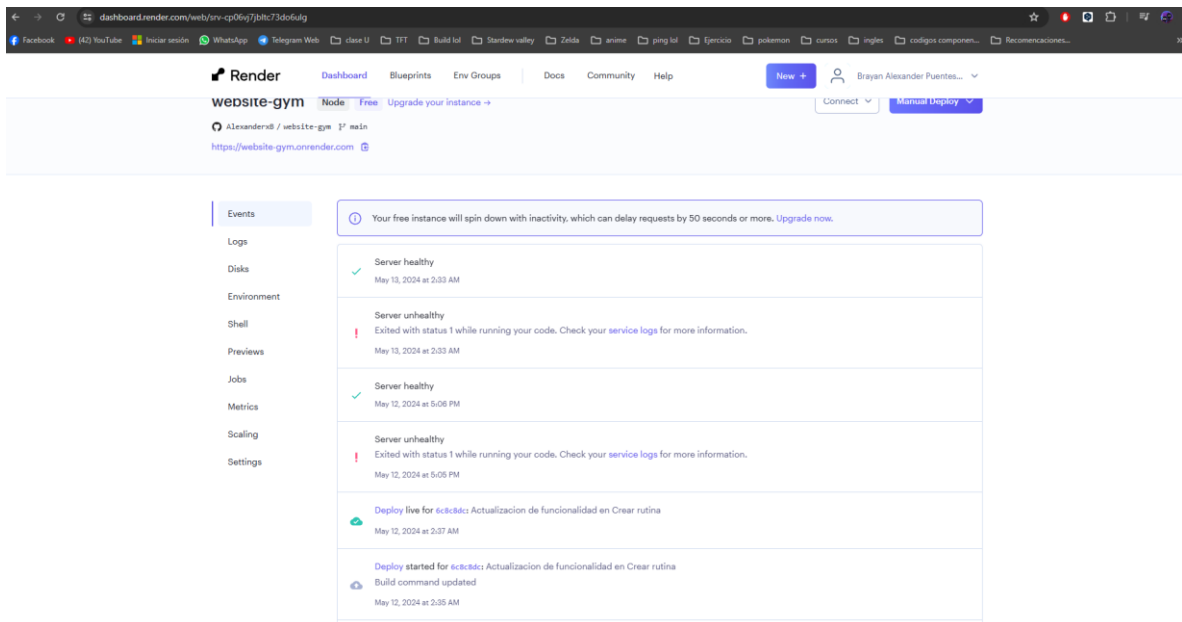
Create Web Service

Feedback Invite a Friend Contact Support

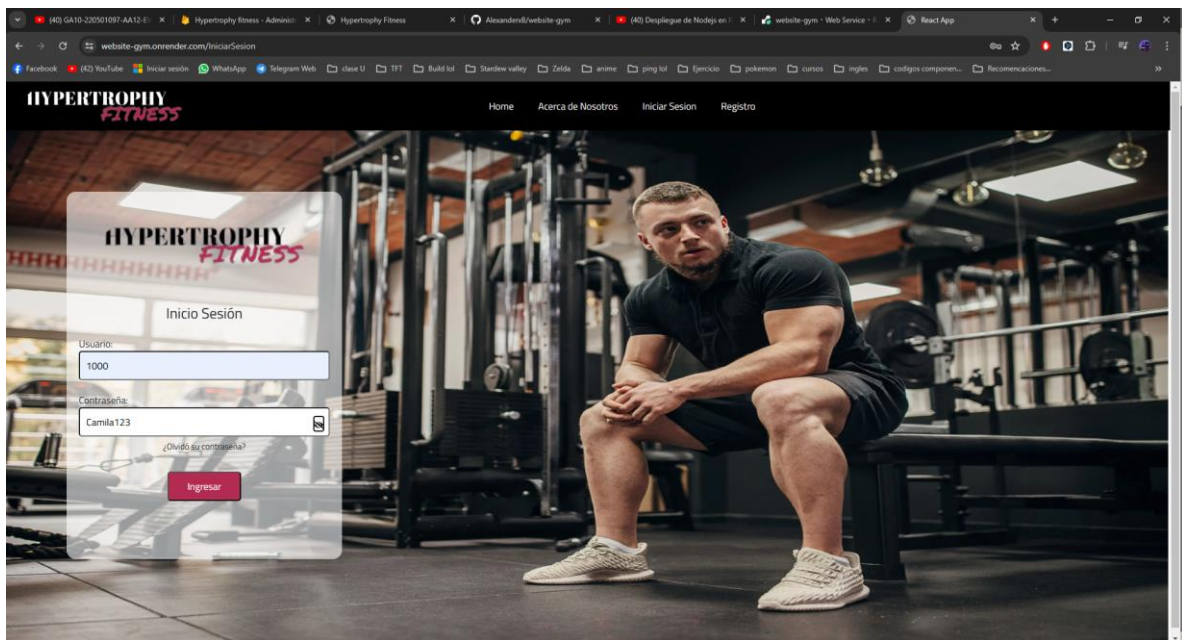
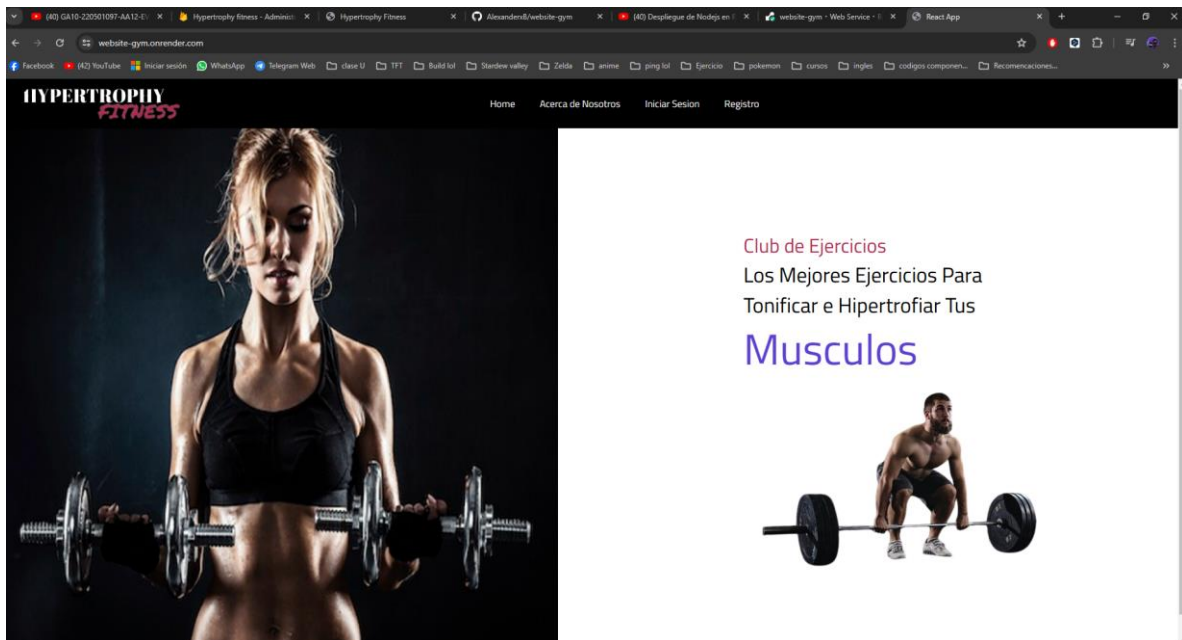
En el lado izquierdo, se selecciona la opción "Logs". Aquí se puede observar el proceso de build, el despliegue, posibles errores y otros detalles relevantes.

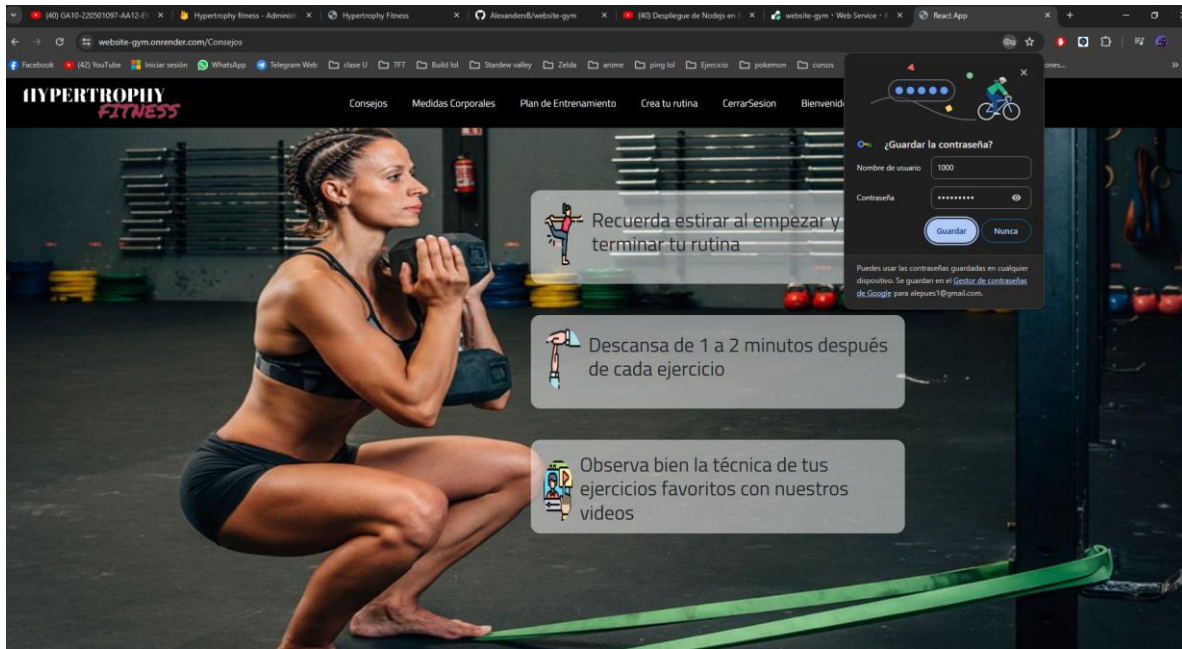


En esta imagen se evidencia con el icono de una nube verde que el despliegue salió exitoso.

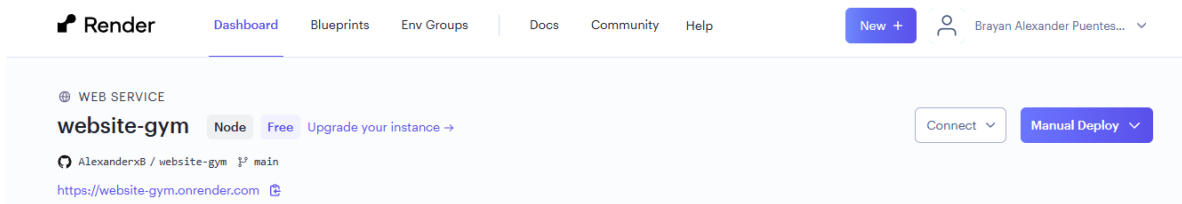


En las imágenes se muestra que al usar la URL generada por Render, se puede verificar que el despliegue del front end fue exitoso.





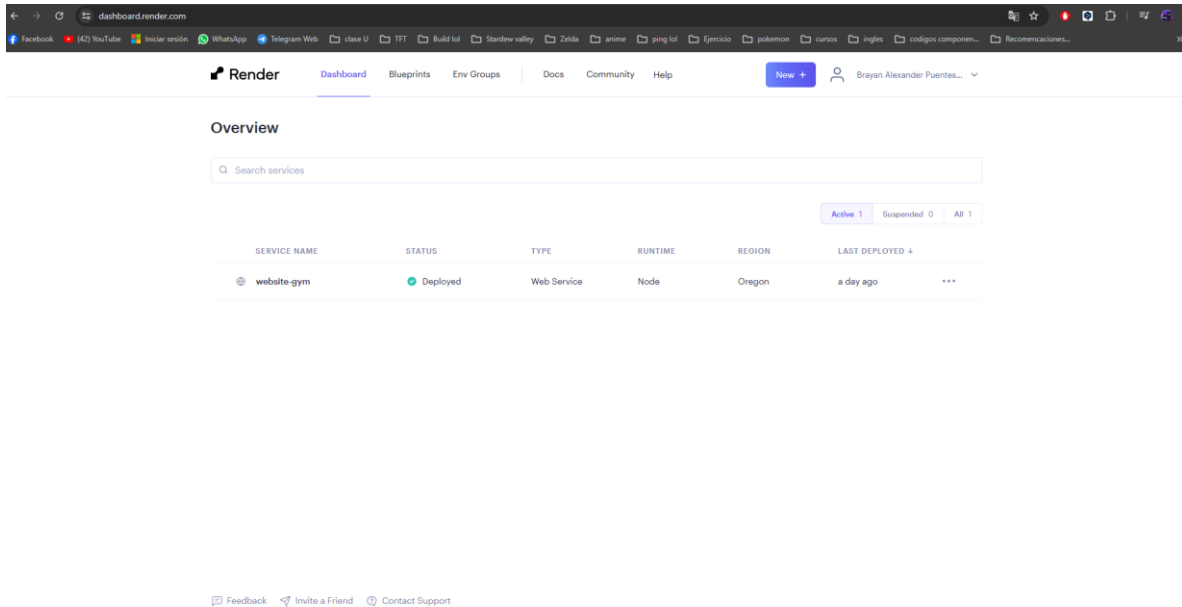
En esta imagen se evidencia la URL que Render genera



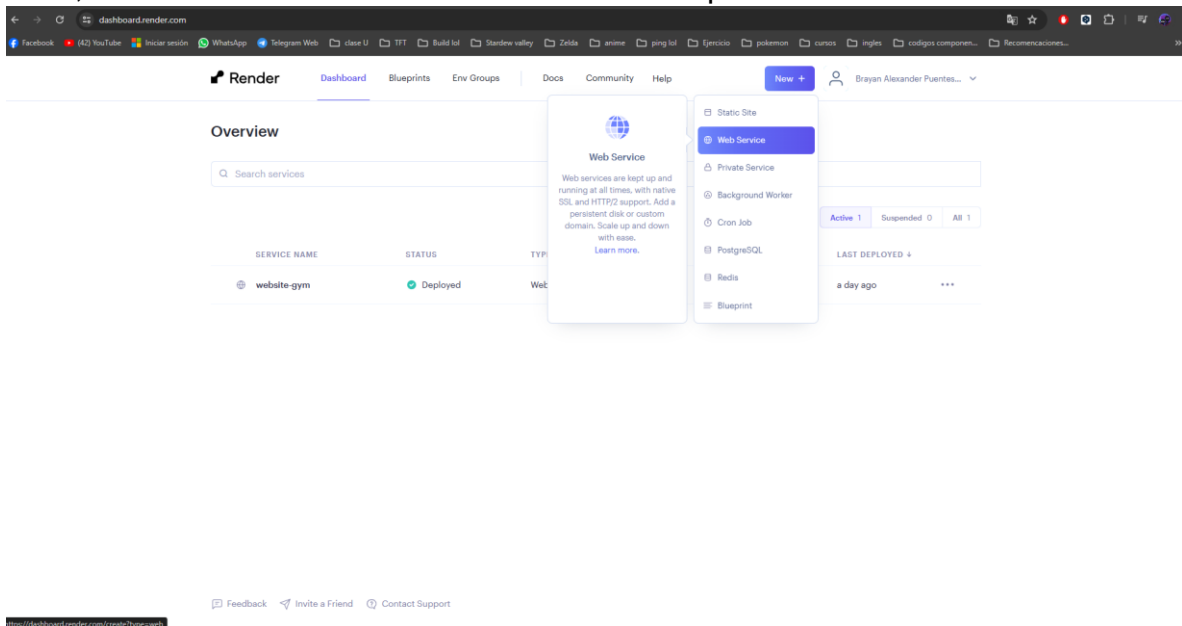
URL DESPLIEGUE FRONT END: <https://website-gym.onrender.com>

## BACK END

Ahora, para realizar el despliegue del back end, se sigue el mismo procedimiento descrito anteriormente. En esta imagen se evidencia el despliegue exitoso del front end.

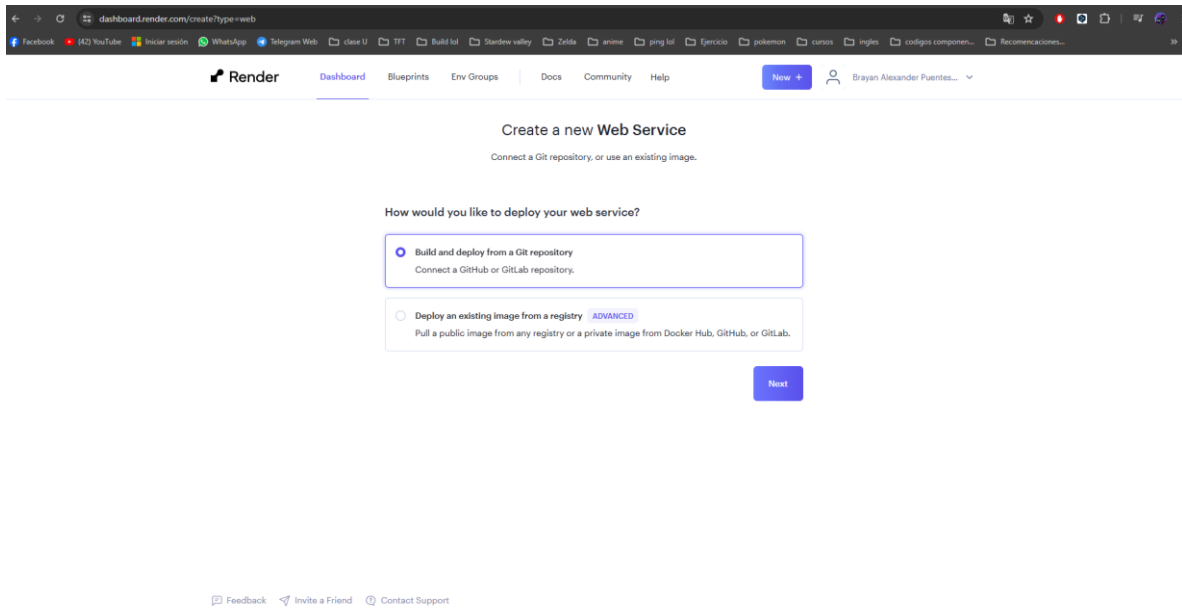


Al hacer clic en el botón que dice "New", se despliegan varias opciones. En este caso, se selecciona la opción "Web Service".

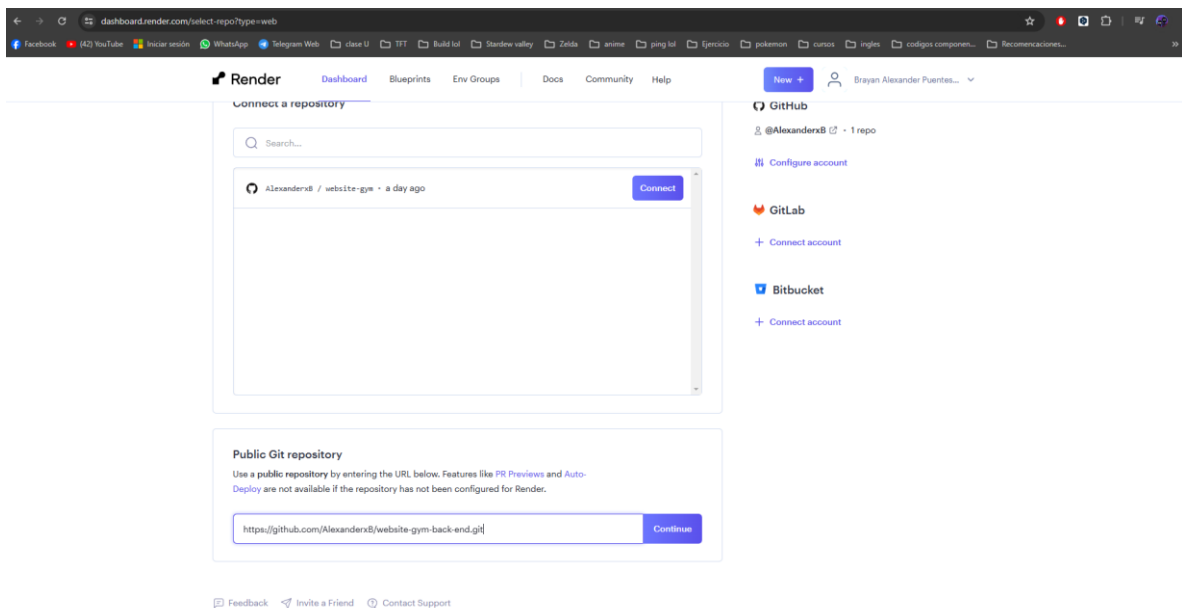




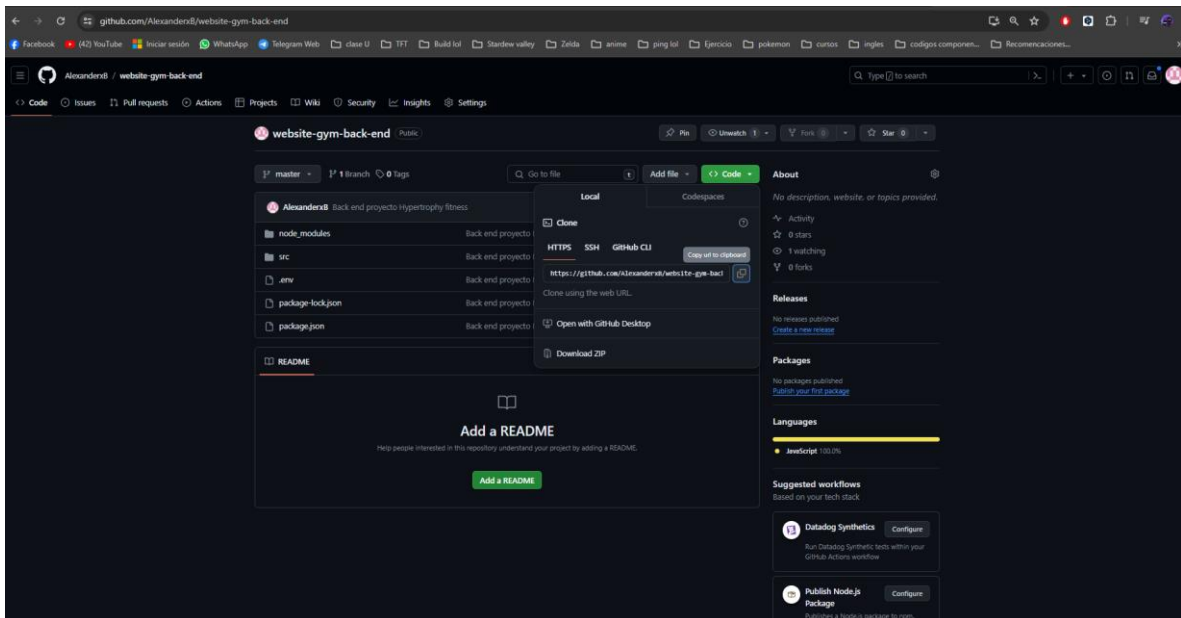
Se vuelve a crear el web service para realizar el despliegue del back, conectando el GitHub



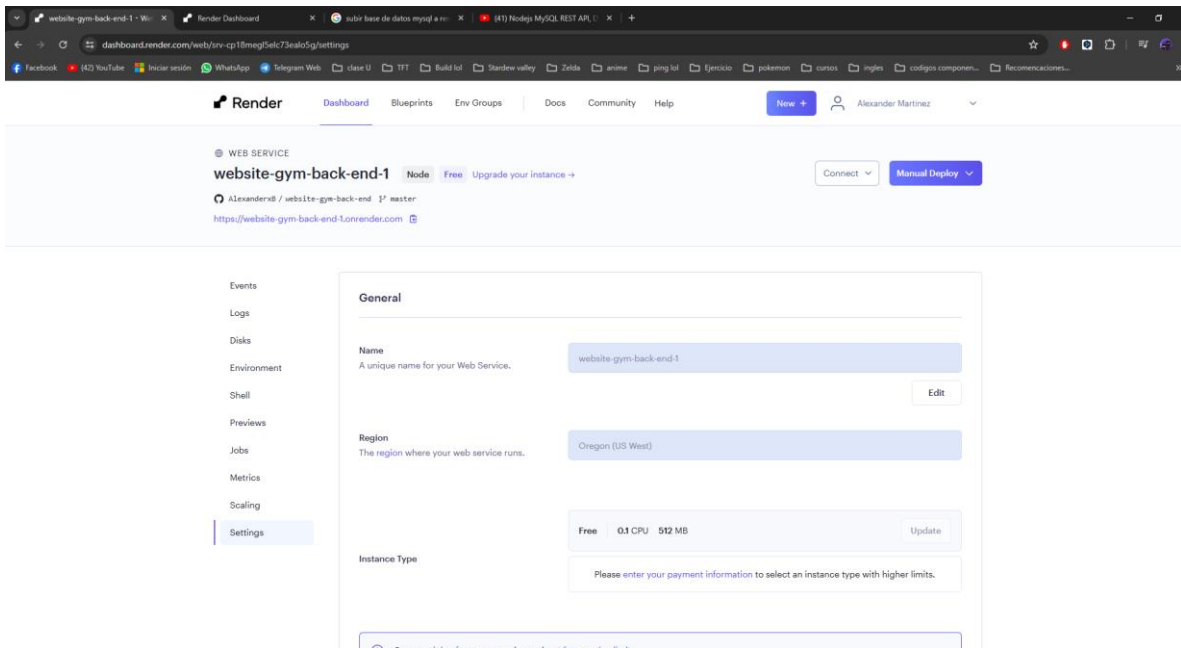
En la sección de "Public Git repository", se introduce la URL del repositorio de GitHub.

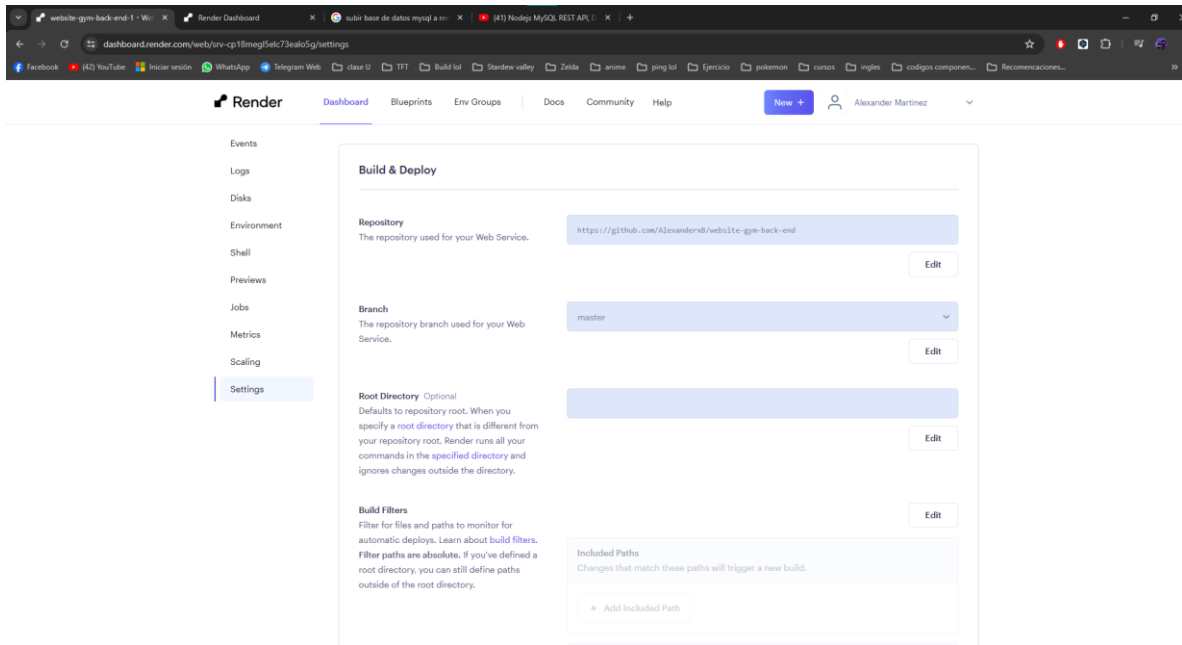


De aquí se genera la URL mencionada en el paso anterior.

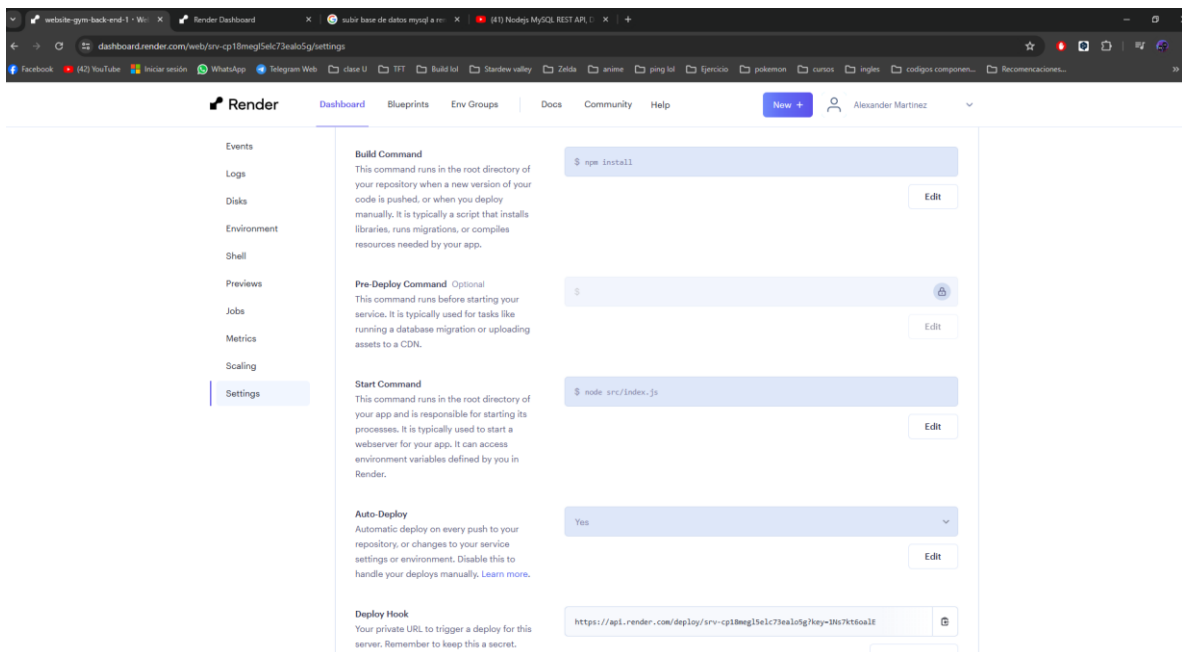


Se vuelve a realizar la configuración para realizar el despliegue y el build

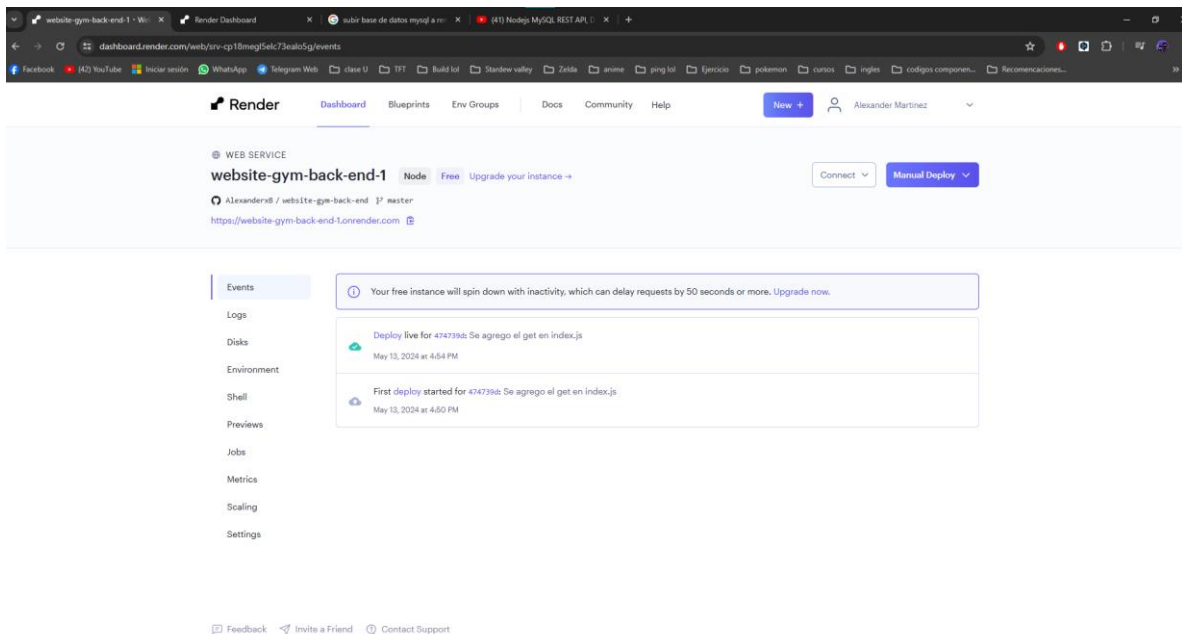




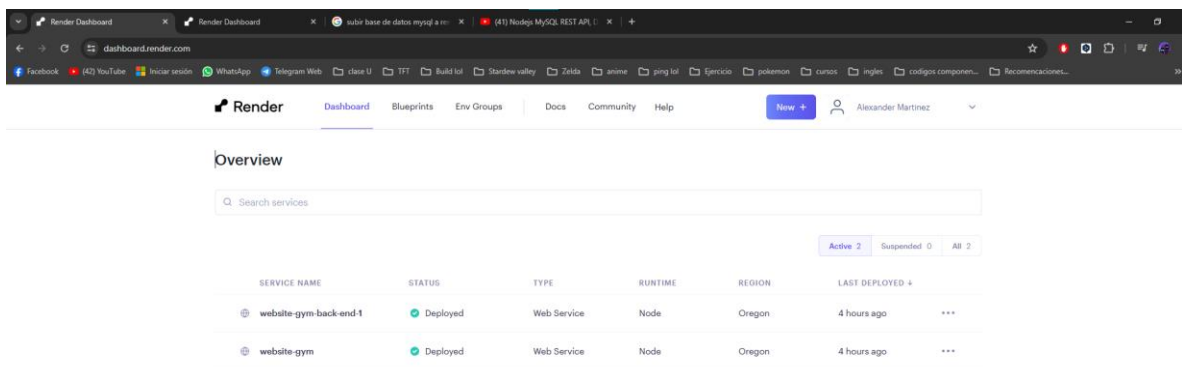
En la sección del build se ingresa el comando npm install y en el apartado de start comando se ingresa el comando node src/index.js



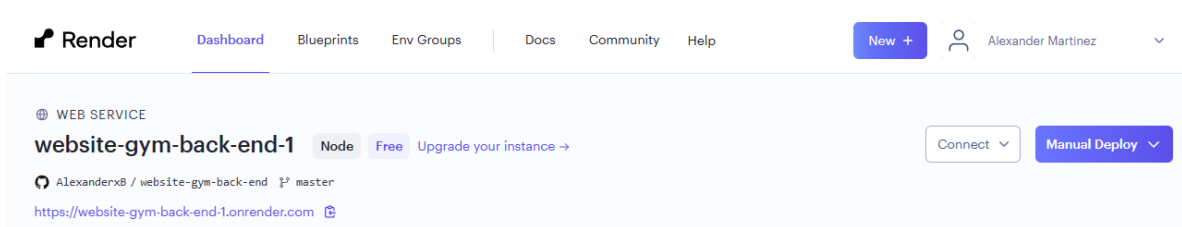
Se evidencia en Events que el despliegue se realizo de forma correcta



Y en la imagen se evidencia el despliegue tanto del front end como del back end

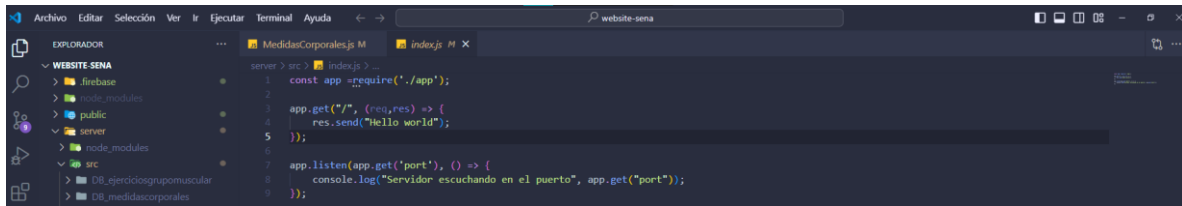


Aquí se puede evidenciar la URL del back end



Se realizan pruebas en Visual Studio donde se coloca un mensaje como "hello world" para verificar que el despliegue del back end se haya realizado

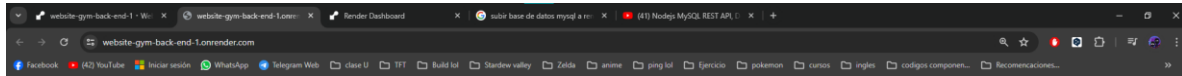
correctamente.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'lib', 'node\_modules', 'public', 'server', and 'src'. The code editor shows a file named 'index.js' with the following code:

```
1 const app = require('./app');
2
3 app.get('/', (req, res) => {
4   res.send("Hello world");
5 });
6
7 app.listen(app.get('port'), () => {
8   console.log("Servidor escuchando en el puerto", app.get("port"));
9 });
```

En esta imagen se evidencia el mensaje hello world, es decir, el despliegue esta funcionando correctamente.

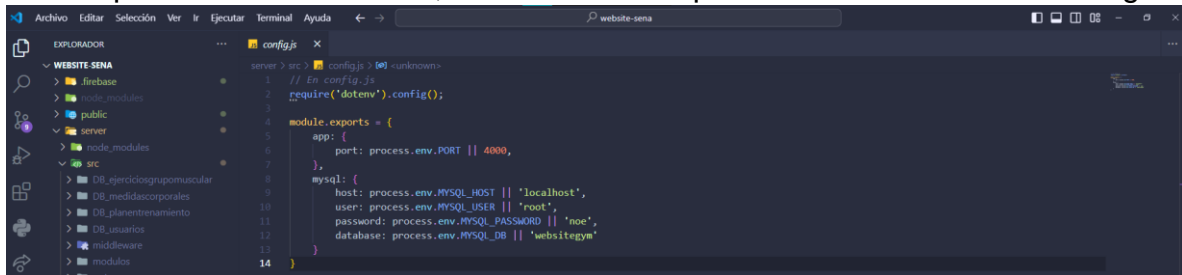


Hello world

URL DESPLIEGUE BACK END: <https://website-gym-back-end-1.onrender.com>

## DESPLIEGUE BASE DE DATOS

Por el momento, tener las credenciales de la base de datos de forma local está bien para el desarrollo, como se puede ver en la imagen.

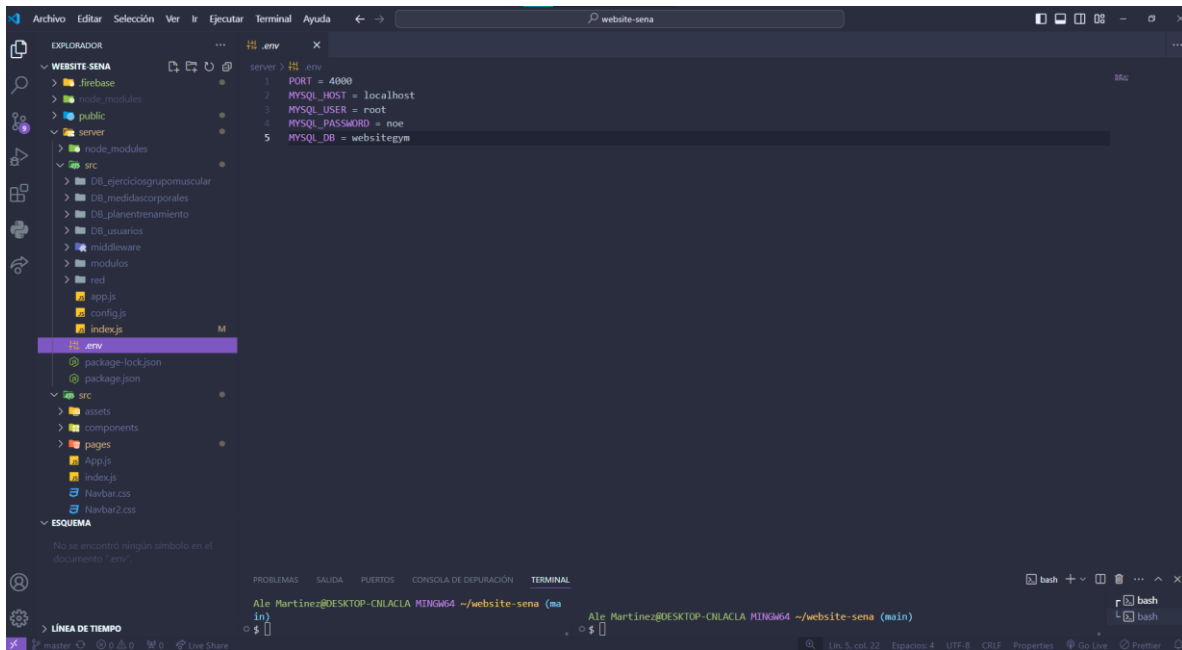


```
server > src > config.js > (unknown)
1 // En config.js
2 require("dotenv").config();
3
4 module.exports = {
5   app: {
6     port: process.env.PORT || 4000,
7   },
8   mysql: {
9     host: process.env.MYSQL_HOST || "localhost",
10    user: process.env.MYSQL_USER || "root",
11    password: process.env.MYSQL_PASSWORD || "noe",
12    database: process.env.MYSQL_DB || "websitegm"
13  }
14 }
```

Para llevar a cabo esto en producción, necesitamos un entorno más preparado. Vamos a utilizar variables de entorno, que son variables almacenadas en el sistema operativo y que nuestra aplicación puede leer y utilizar.

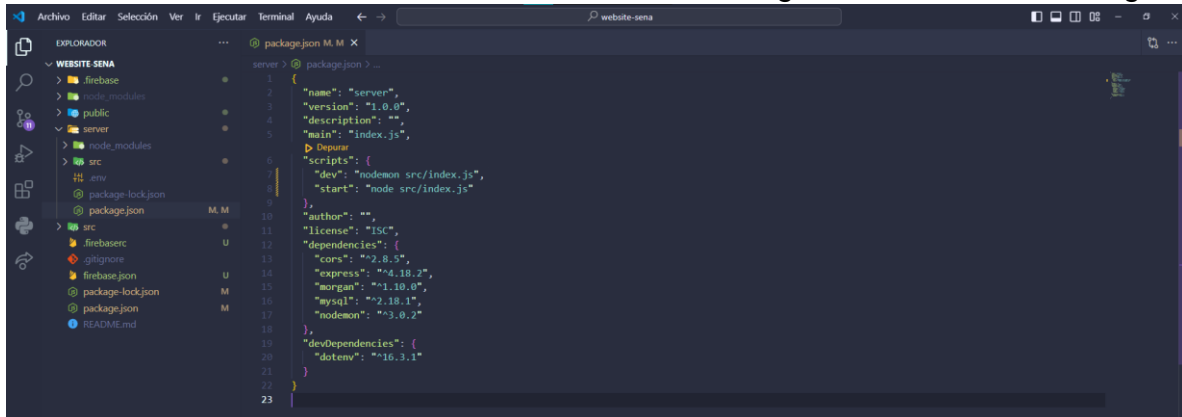
Primero, necesitamos instalar un módulo en la terminal de Visual Studio Code. Podemos hacerlo con el comando `npm i dotenv`.

Este comando nos permite leer un archivo llamado `“.env”`, donde definiremos nuestras variables de entorno.



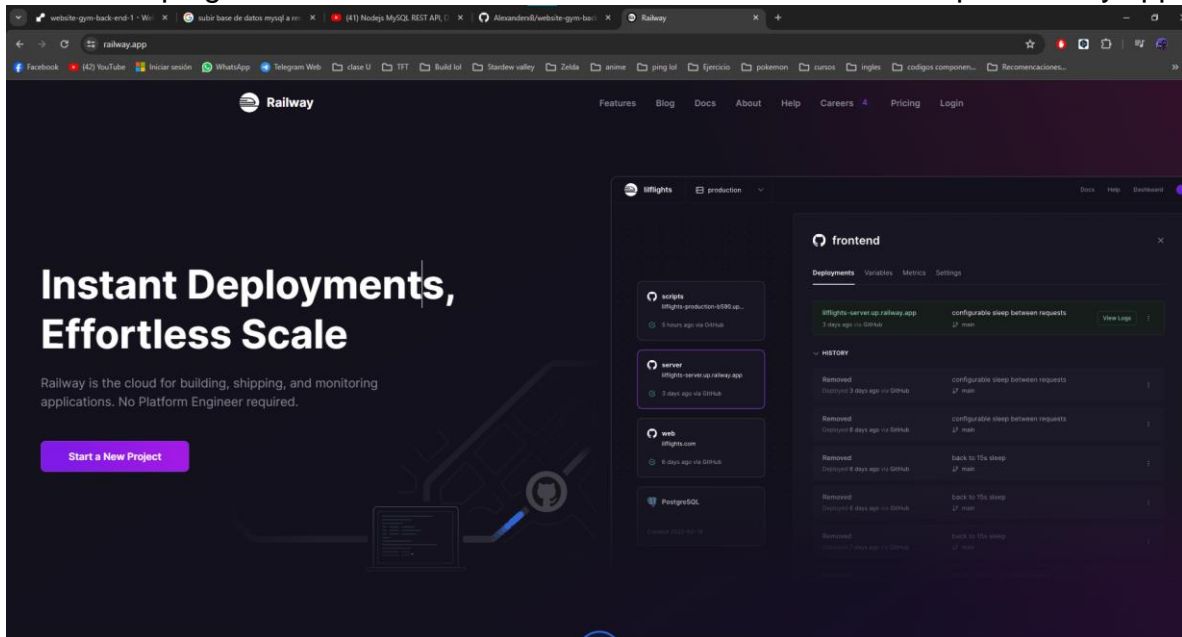
```
server > .env
1 PORT = 4000
2 MYSQL_HOST = localhost
3 MYSQL_USER = root
4 MYSQL_PASSWORD = noe
5 MYSQL_DB = websitegm
```

En el archivo package.json, se tendrán en cuenta dos comandos: "dev" y "start". En desarrollo, se utilizará el comando: "dev": "nodemon src/index.js". Por otro lado, en producción se usará el comando: "start": "node src/index.js", como se muestra en la siguiente imagen:

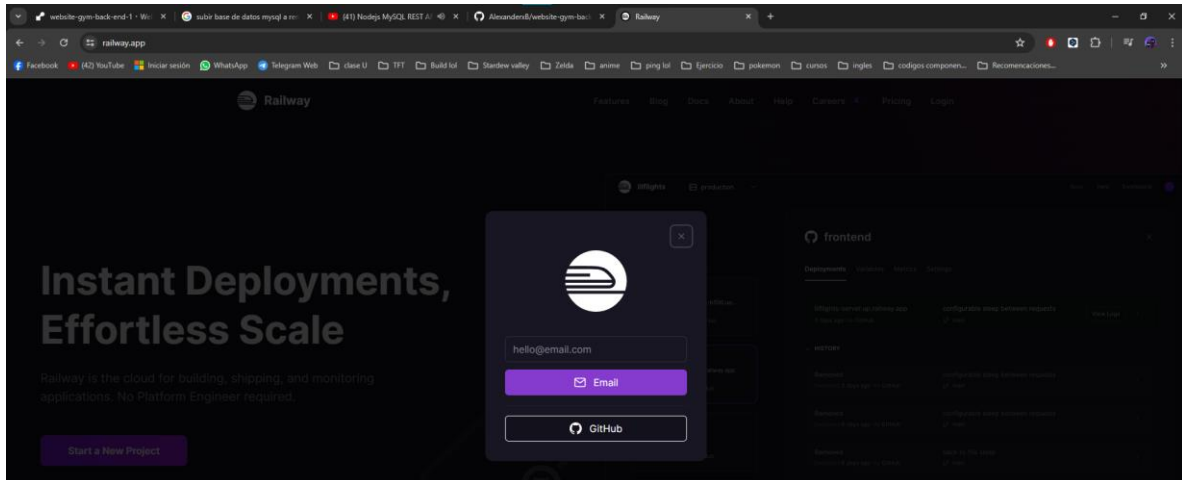


```
1 {
2   "name": "server",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "dev": "nodemon src/index.js",
8     "start": "node src/index.js"
9   },
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "cors": "^2.8.5",
14    "express": "^4.18.2",
15    "morgan": "^1.10.0",
16    "mysql": "^2.18.1",
17    "nodemon": "^3.0.2"
18  },
19  "devDependencies": {
20    "dotenv": "^16.3.1"
21  }
22 }
23
```

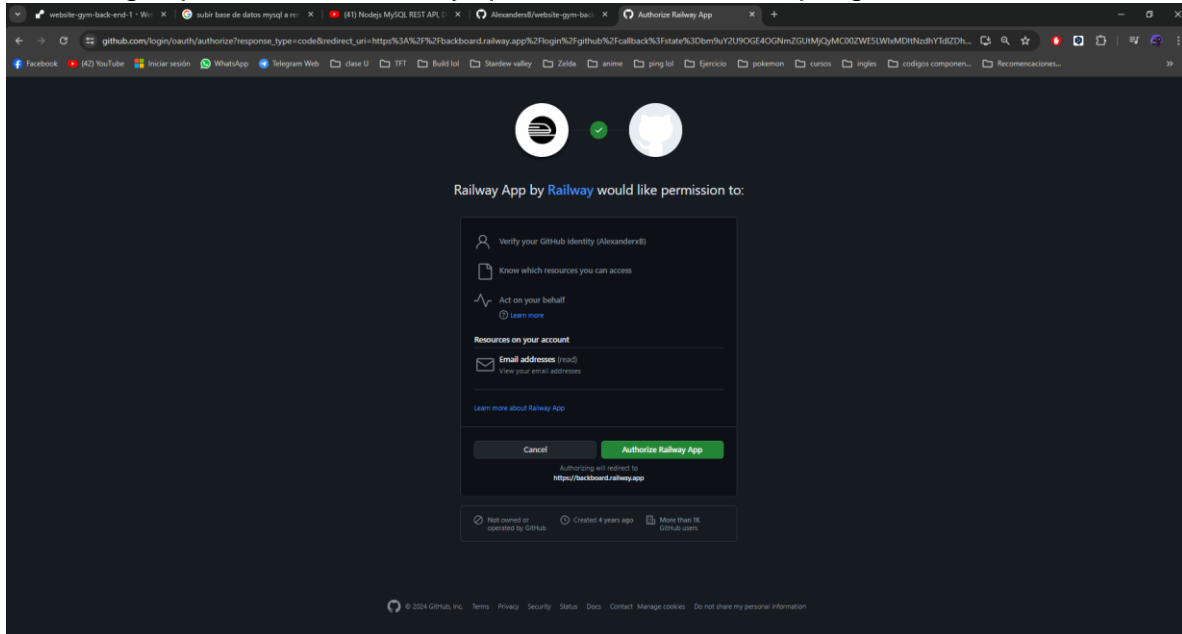
Para desplegar la base de datos se va a usar <https://railway.app/>



Para utilizar Railway, primero debes iniciar sesión. En este caso, lo haremos a través de GitHub.

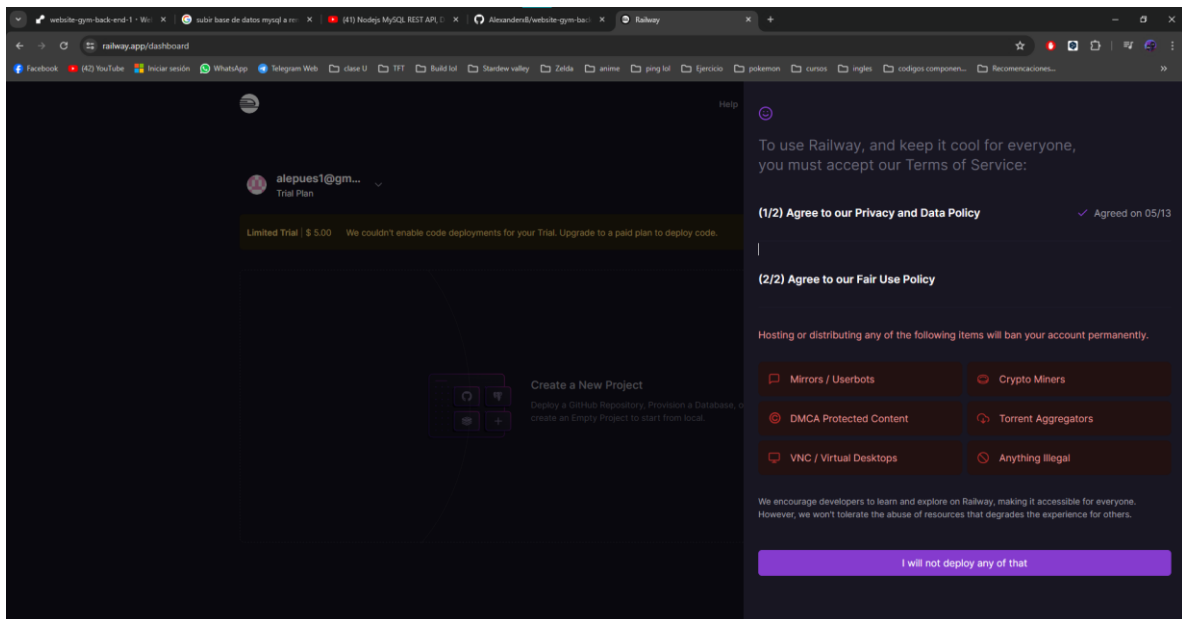
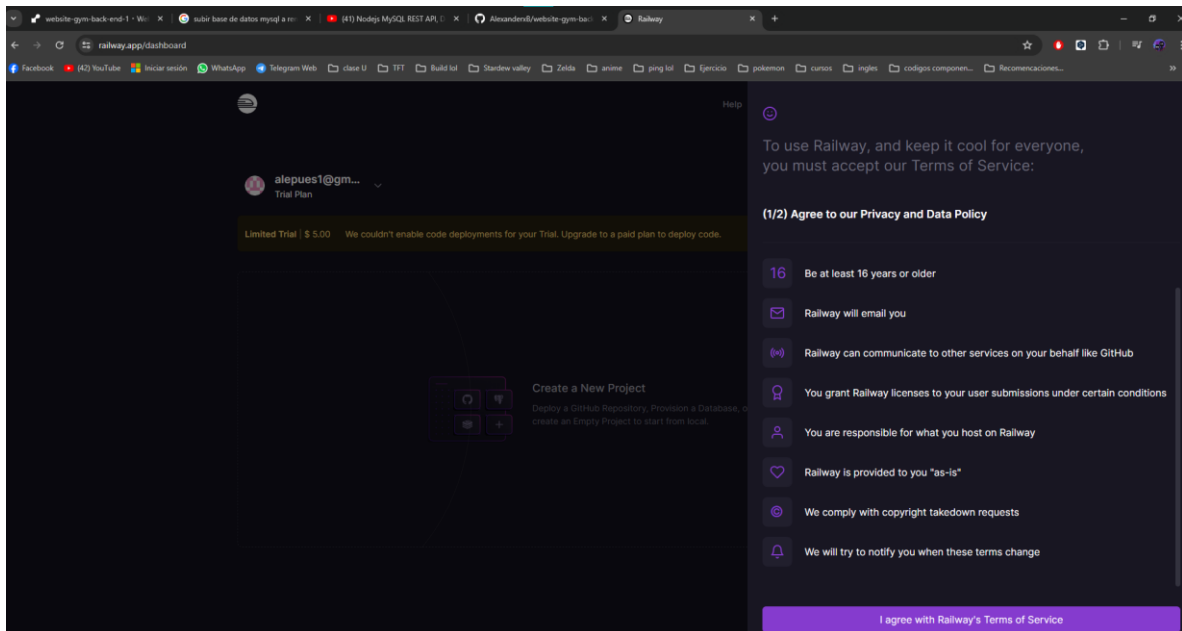


Se otorga permiso a Railway para realizar el despliegue mediante GitHub.

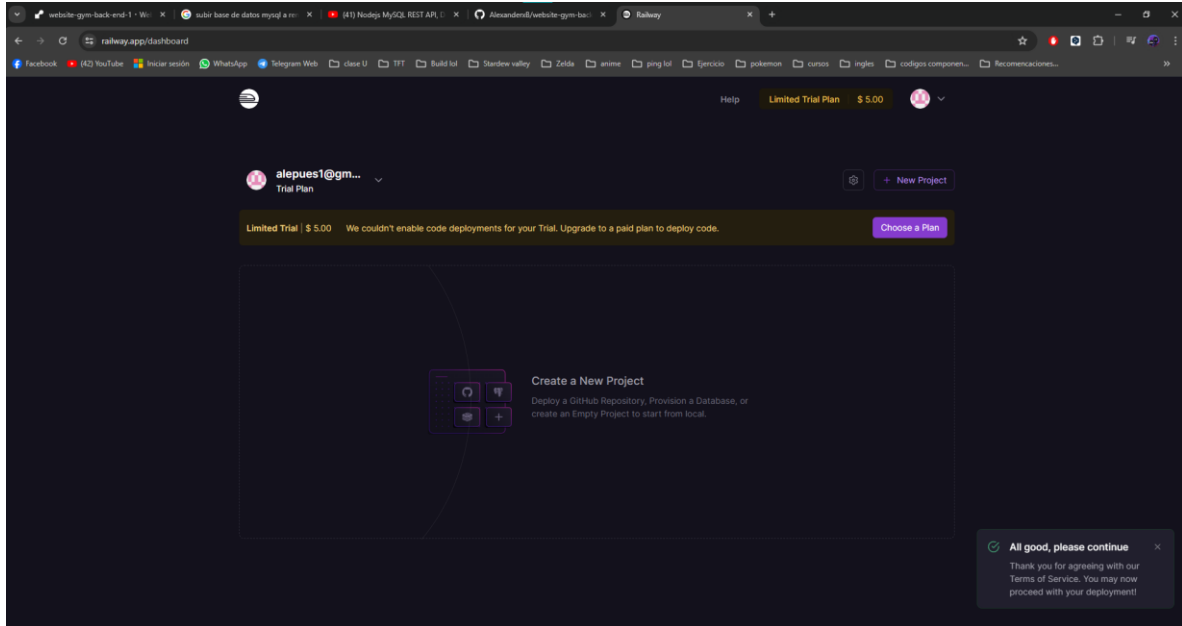




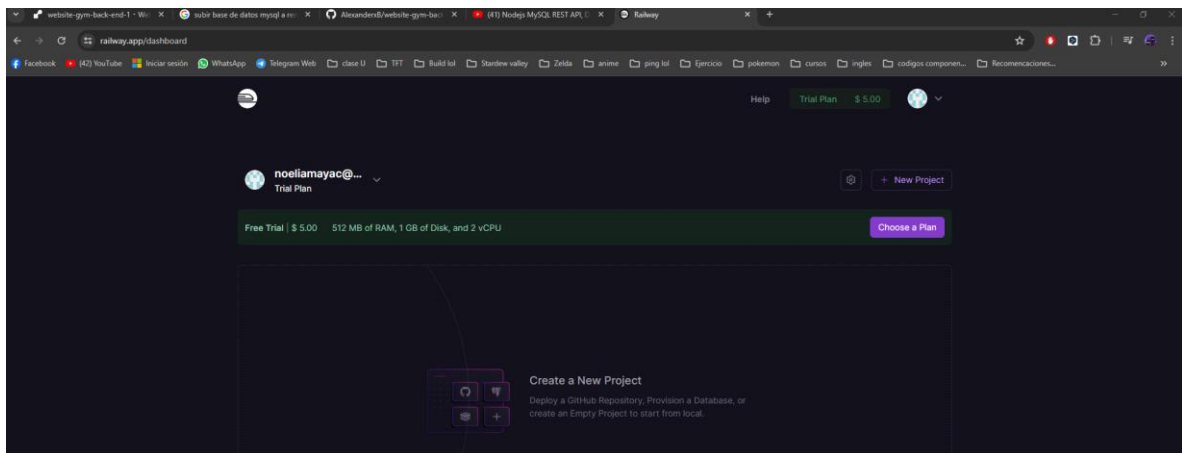
Se aceptan los términos y condiciones para poder usar railway



El despliegue se realizó con éxito en otra cuenta que permite el uso de Railway gratis ya que como se evidencia en la imagen en la cuenta que se creo no es posible usarla porque se debe pagar.



Esta es la cuenta donde se va a realizar el despliegue ya que es gratis

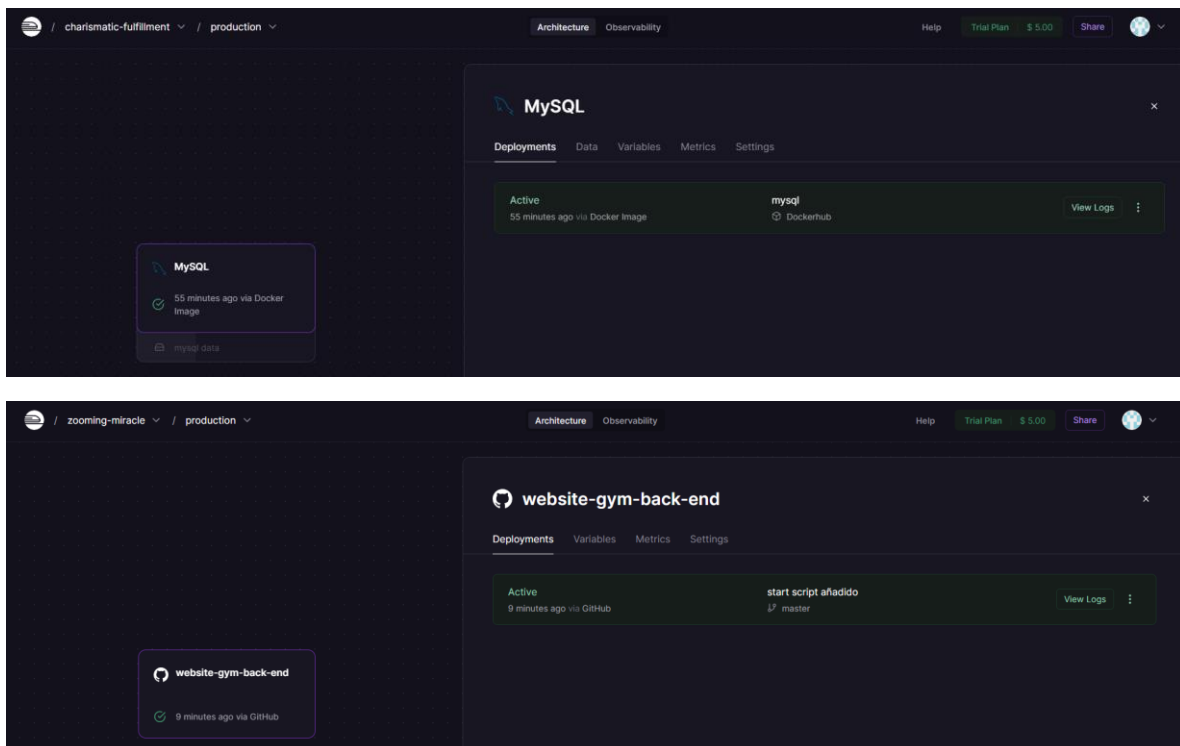


A continuación, se detallará el proceso para utilizar Railway:

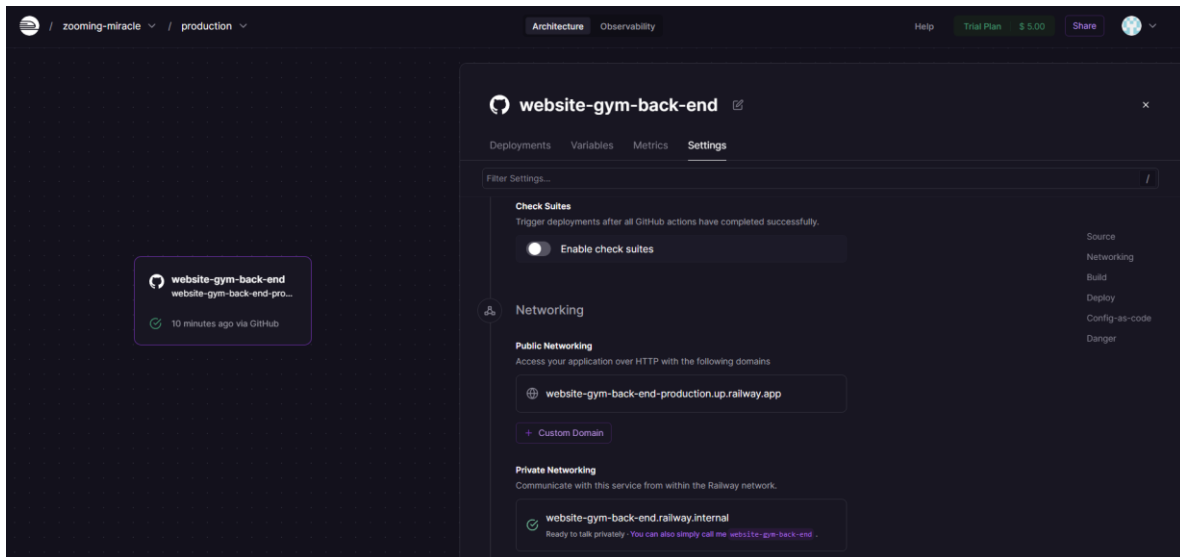
Agregar una base de datos: Una vez dentro de tu proyecto, busca la opción para agregar un nuevo servicio o recurso. Allí encontrarás la opción para añadir una base de datos. Railway es compatible con varias bases de datos, como PostgreSQL, MySQL, MongoDB, entre otras. Selecciona la base de datos que mejor se ajuste a tus necesidades.

En este caso particular, se optó por MySQL debido a que el proyecto fue desarrollado con MySQL y Workbench. Por lo tanto, se crearon dos proyectos en Railway: uno para MySQL y otro para desplegar la base de datos a través del repositorio de GitHub del backend.

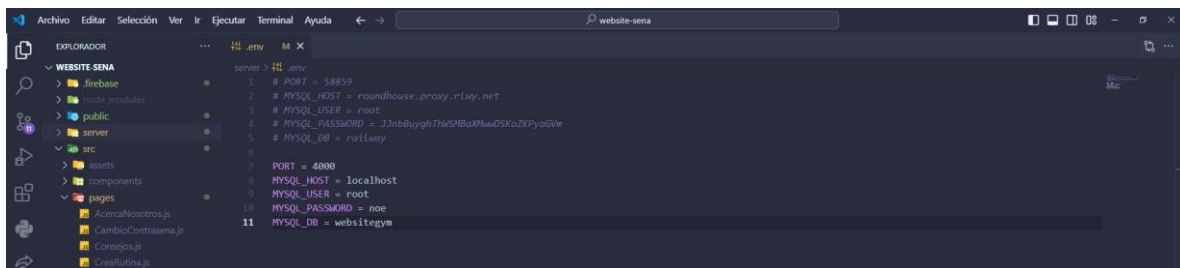
Estos pasos se ilustran en la siguiente imagen:



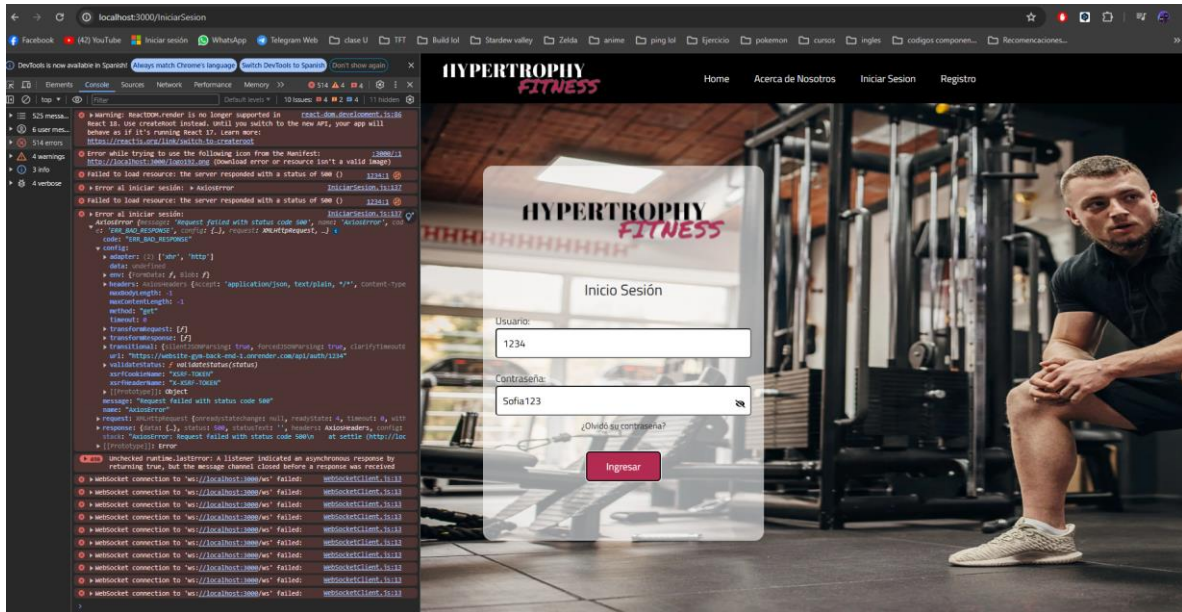
Después de seleccionar el tipo de base de datos, Railway te guiará a través de los pasos necesarios para configurarla. Esto puede incluir la selección de la región del servidor, el plan de precios (si corresponde), y la configuración de credenciales y otras opciones específicas de la base de datos.



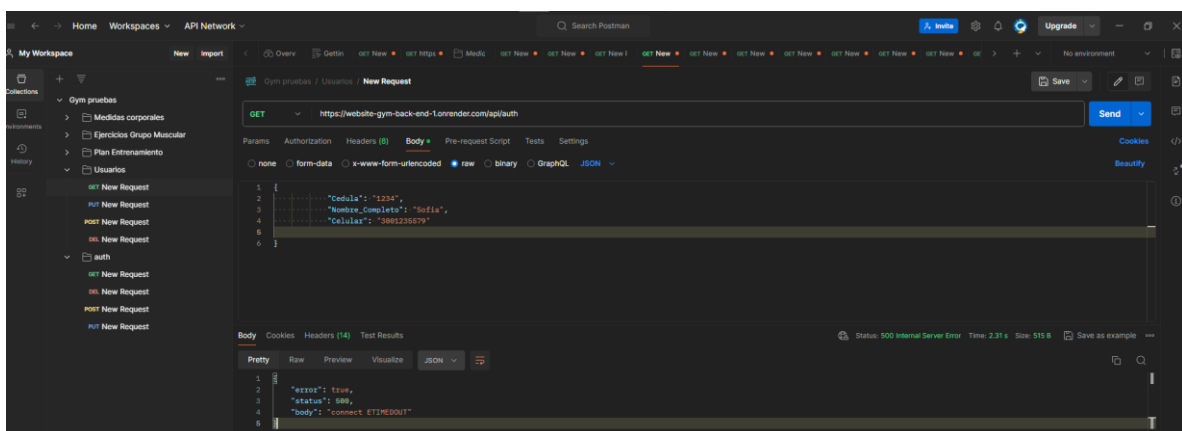
En esta imagen se evidencian las variables de entorno para hacer la conexión con la base de datos, por el momento se están usando los datos para acceder a la pagina web de forma local porque a pesar de que la base de datos tuvo un deploy exitoso, y se hicieron las configuraciones respectivas en el back end como se evidencia en la imagen



Sin embargo, al hacer pruebas se generan los siguientes errores como se muestra en la imagen



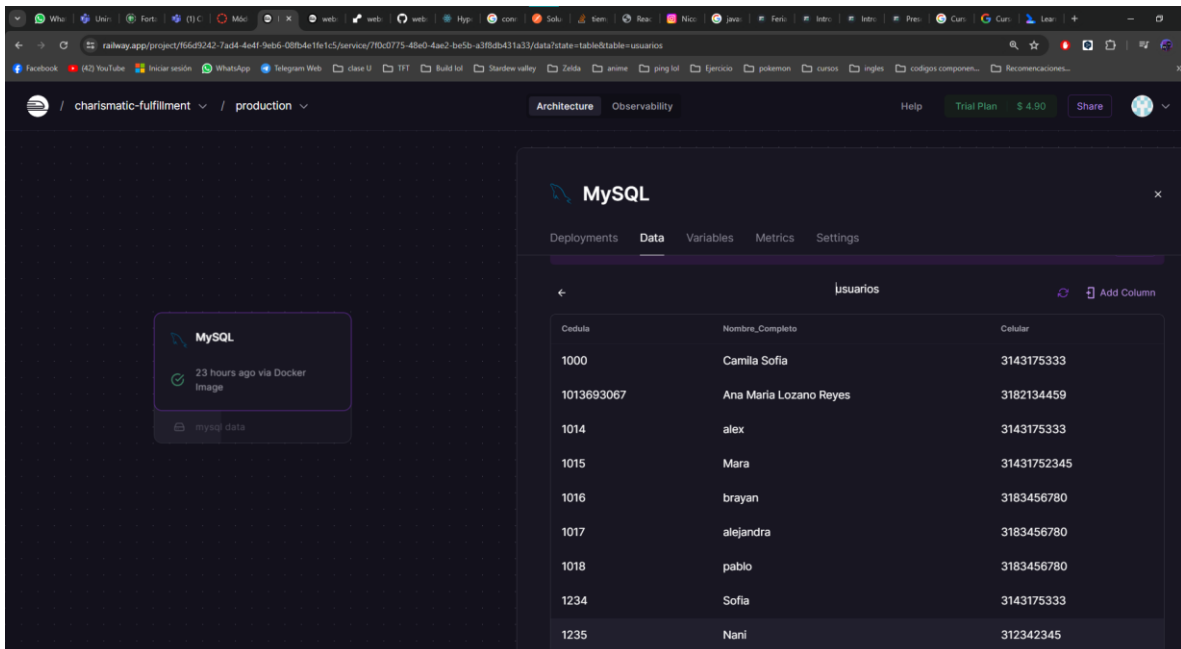
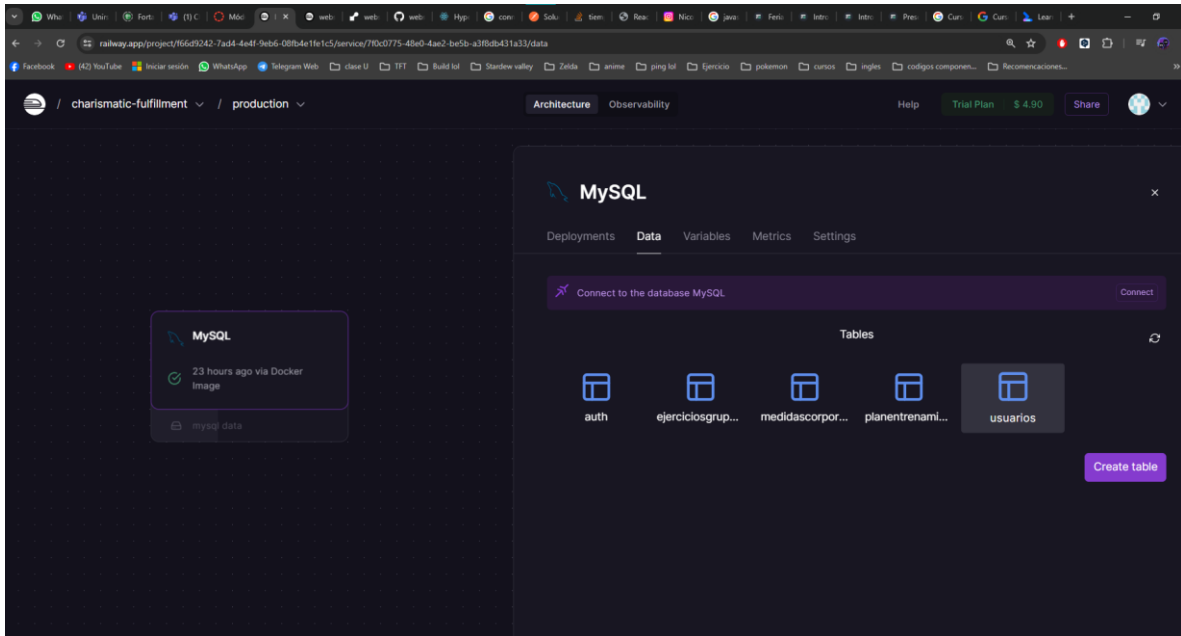
También aparece el mismo error en el postman, el error es el : “connect ETIMEDOUT”



El error que estamos enfrentando parece estar relacionado con el peso de las imágenes que hemos incluido en el frontend, lo que está causando tiempos de carga más largos. Además, existe la posibilidad de que el problema esté relacionado con el hecho de que tanto el backend como el frontend se presentan

en el mismo archivo. Sin embargo, la información que hemos encontrado en Internet no es lo suficientemente clara para resolver este error.

Aquí se evidencia el deploy exitoso mencionado de la base de datos



MySQL Workbench

webstegem x MySQL Model x EER Diagram x Hypertrophy Fitness - Miami x unconnected x EER Diagram2 x

File Edit View Query Database Server Tools Scripting Help

Query 1 SQL File 2 SQL File 4 usuarios auth

1 SELECT \* FROM railway.usuarios

Automatic context help disabled. Use the toolbar manually get help for the current caret position or toggle automatic help

Result Grid

Cedula	Nombre_Completo	Celular
1000	Carla Sofia	3183322113
1013893067	Alex Maria Luciano Reyes	3182134499
1014	alex	3143173333
1015	Nara	3143173345
1016	brayan	3183456780
1017	alexandra	3183456780
1018	gabie	3183456780
1235	Nari	312242345
0008	0008	0008

usuarios 1 x

Output

#	Time	Action	Message	Duration / Fetch
73	19:57:51	/*40101 SET CHARACTER_SET_RESULTS=BOLD_CHARACTER_SET_RESULTS */	0 rows(s) affected	0.125 sec
74	19:57:51	/*40101 SET COLLATION_CONNECTION=BOLD_COLLATION_CONNECTION */	0 rows(s) affected	0.125 sec
75	19:57:51	/*40111 SET SQL_NOTES=BOLD_SQL_NOTES */	0 rows(s) affected	0.141 sec
76	20:18:17	UNLOCK TABLES	0 rows(s) affected	0.141 sec
77	20:18:56	SELECT * FROM railway.usuarios LIMIT 0, 1000	0 rows(s) returned	0.140 sec / 0.000 sec
78	20:19:22	LOCK TABLES 'usuarios' WRITE	0 rows(s) affected	0.125 sec

Table: auth

Columns:

- Cedula: varchar(20)
- Nombre\_Completo: varchar(200)
- Contraseña: varchar(255)

LINK DESPLIEGUE BASE DE DATOS: [website-gym-back-end-production.up.railway.app](https://website-gym-back-end-production.up.railway.app)

# **PLAN PARA LA ELIMINACIÓN DE LA PÁGINA WEB**

## **PASOS ASEGUIR PARA LA ELIMINACIÓN DE LA PAGINA WEB**

### **1. EVALUACIÓN DE LA RAZÓN DE LA ELIMINACIÓN**

La página web será eliminada debido a la reestructuración de la empresa y el cambio de enfoque en sus estrategias de marketing y comunicación.

### **2. COMUNICACIÓN CON LAS PARTES INTERESADAS**

Se notificará a todas las partes interesadas, incluidos los usuarios registrados, el equipo interno y los proveedores de servicios, sobre la decisión de eliminar la página web.

### **3. RESPALDO DE DATOS IMPORTANTES**

Se realizará un respaldo completo de todos los datos importantes, incluidos los perfiles de usuario, el contenido del sitio y cualquier información relevante almacenada en la base de datos.

### **4. SUSPENSIÓN DE NUEVAS SUSCRIPCIONES Y ACTUALIZACIONES**

Se suspenderá la capacidad de registrar nuevas cuentas de usuario y se detendrá cualquier actualización o modificación de contenido en la página web existente.

### **5. REDIRECCIONAMIENTO DE TRÁFICO**

Se establecerán redireccionamientos adecuados para guiar a los visitantes de la página web hacia otras plataformas o recursos relevantes, como redes sociales, blogs u otras páginas de la empresa.

### **6. CANCELACIÓN DE SERVICIOS Y CONTRATOS**



Se cancelarán todos los servicios y contratos asociados con la página web, como el alojamiento web, el registro de dominio y cualquier suscripción a servicios adicionales.

## **7. ELIMINACIÓN DE LA INFRAESTRUCTURA**

Se procederá a la eliminación de la infraestructura de alojamiento y cualquier otro componente técnico asociado con la página web, siguiendo los procedimientos recomendados por los proveedores de servicios.

## **8. NOTIFICACIÓN DE ELIMINACIÓN**

Se enviará una notificación final a todos los usuarios registrados y suscriptores de la página web, informándoles sobre la eliminación inminente y proporcionando orientación sobre cómo acceder a recursos alternativos si es necesario.

## **9. VERIFICACIÓN Y CONFIRMACIÓN DE LA ELIMINACIÓN**

Se realizará una verificación final para confirmar que todos los datos y componentes asociados con la página web han sido eliminados de manera segura y completa.

## **10. ANÁLISIS Y DOCUMENTACIÓN**

Se realizará un análisis post-eliminación para evaluar el proceso y documentar cualquier lección aprendida o recomendaciones para futuras eliminaciones de páginas web.

## **CONCLUSIONES**

La documentación del plan de migración y el respaldo de los datos son componentes fundamentales en cualquier proyecto de desarrollo de software. Proporcionan una guía detallada para llevar a cabo la migración de manera eficiente y efectiva, minimizando los riesgos y asegurando la continuidad del negocio.

Al centrarse en estos aspectos clave, los equipos de desarrollo pueden mitigar los riesgos asociados con la migración de datos y garantizar una transición suave entre diferentes plataformas o versiones del software. Además, una documentación exhaustiva facilita la colaboración entre los miembros del equipo y proporciona un punto de referencia claro para futuras actualizaciones o modificaciones del sistema.