

ARCHIVOS CARGADOS EN LA PLATAFORMA DE PRODUCCIÓN

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

INTRODUCCIÓN.....	4
CLÚSTERES Y ANÁLISIS DE RIESGO	5
DEFINICIÓN DE CLÚSTERES	5
CLASIFICACIÓN DE CLÚSTERES.....	5
CLÚSTER DE ALTA DISPONIBILIDAD (HA)	5
CLÚSTER DE CARGA BALANCEADA	5
CLÚSTER DE COMPUTACIÓN DE ALTO RENDIMIENTO (HPC)	5
CLÚSTER DE ALMACENAMIENTO	5
CARACTERÍSTICAS DE ARQUITECTURA DE LOS CLÚSTERES	6
REDUNDANCIA.....	6
ESCALABILIDAD	6
DISTRIBUCIÓN DE CARGA	6
TOLERANCIA A FALLOS	6
PARALELISMO	7
TIPOS DE PROCESAMIENTO DE CLÚSTERES.....	8
PROCESAMIENTO PARALELO	8
PROCESAMIENTO DISTRIBUIDO	8
PROCESAMIENTO DE CARGA DE TRABAJO	8
CONCEPTO DE ANÁLISIS DE RIESGO	9
IDENTIFICACIÓN DE RIESGOS	9
EVALUACIÓN DE RIESGOS	9
PRIORIZACIÓN DE RIESGOS	9
MITIGACIÓN DE RIESGOS.....	9
MONITOREO Y REVISIÓN CONTINUA	9
ANÁLISIS DE IMPACTO Y GESTIÓN DE PRIORIDAD EN LOS CLÚSTERES.....	10
IDENTIFICACIÓN DE ACTIVOS CRÍTICOS	10
EVALUACIÓN DE IMPACTO	10
ANÁLISIS DE ESCENARIOS.....	10
PRIORIZACIÓN DE ACCIONES DE MITIGACIÓN	10
ESTRATEGIAS DE GESTIÓN DE PRIORIDAD	11
CLASIFICACIÓN DE INCIDENTES	11
IMPLEMENTACIÓN DE UN MODELO DE PRIORIZACIÓN	11

ASIGNACIÓN DE RECURSOS.....	11
COMUNICACIÓN Y COORDINACIÓN.....	11
RECUPERACIÓN DE DESASTRES: RPO Y RTO	12
RPO (RECOVERY POINT OBJECTIVE).....	12
RTO (RECOVERY TIME OBJECTIVE)	12
RPO Y RTO EN CLÚSTERES.....	12
RPO EN CLÚSTERES	12
RTO EN CLÚSTERES	12
HERRAMIENTAS Y TÉCNICAS PARA LA RECUPERACIÓN DE DESASTRES EN CLÚSTERES	13
REPLICACIÓN DE DATOS EN TIEMPO REAL O PERIÓDICA	13
VIRTUALIZACIÓN DE SERVIDORES.....	13
CONMUTACIÓN POR ERROR AUTOMÁTICA.....	13
AUTOMATIZACIÓN DE PROCESOS DE RECUPERACIÓN	13
PRUEBAS DE RECUPERACIÓN DE DESASTRES (DR)	13
PLANES DE CONTINUIDAD	14
IMPORTANCIA DE LOS PLANES DE CONTINUIDAD DEL NEGOCIO.....	14
MANTENIMIENTO DE LA OPERATIVIDAD	14
MINIMIZACIÓN DEL TIEMPO DE INACTIVIDAD	14
PROTECCIÓN DE DATOS Y ACTIVOS.....	14
CUMPLIMIENTO REGULATORIO Y CONTRACTUAL.....	14
PASOS PARA EL DISEÑO DE PLANES DE CONTINUIDAD.....	16
IDENTIFICACIÓN DE SERVICIOS CRÍTICOS	16
ANÁLISIS DE RIESGOS.....	16
ESTABLECIMIENTO DE OBJETIVOS DE RECUPERACIÓN.....	16
DESARROLLO DE ESTRATEGIAS DE RECUPERACIÓN.....	16
IMPLEMENTACIÓN Y PRUEBA	16
MANTENIMIENTO Y ACTUALIZACIÓN.....	16
HOSTING	17
CONCLUSIONES	45
BIBLIOGRAFÍA.....	45

INTRODUCCIÓN

En el mundo de la tecnología de la información, la eficiencia, la disponibilidad y la seguridad son pilares fundamentales para el éxito de cualquier proyecto o sistema. En este contexto, los clústeres emergen como una solución poderosa para optimizar recursos, aumentar la disponibilidad y garantizar la confiabilidad de los servicios informáticos.

Este documento aborda dos aspectos en el ámbito de los clústeres: su definición y clasificación, así como el análisis de riesgo asociado a su implementación y operación.

Posteriormente, nos adentraremos en el análisis de riesgo, un proceso fundamental para identificar, evaluar y mitigar las amenazas que podrían comprometer la integridad y la continuidad de los clústeres. Desde la identificación de activos críticos hasta la implementación de medidas de seguridad y recuperación.

Finalmente, un análisis sobre el despliegue exitoso de una página web en un entorno de clústeres, destacando los procesos y las herramientas utilizadas para garantizar su disponibilidad y rendimiento tanto a nivel local como en Internet.

CLÚSTERES Y ANÁLISIS DE RIESGO

DEFINICIÓN DE CLÚSTERES

En informática y tecnología, un clúster se refiere a un conjunto de computadoras interconectadas que trabajan juntas como si fueran una única entidad. Estas máquinas, o nodos, en un clúster están diseñadas para colaborar en tareas específicas para lograr un objetivo común. Los clústeres son utilizados para mejorar el rendimiento, la disponibilidad y la escalabilidad de los sistemas informáticos.

CLASIFICACIÓN DE CLÚSTERES

CLÚSTER DE ALTA DISPONIBILIDAD (HA)

Estos clústeres están diseñados para garantizar la continuidad del servicio mediante la redundancia de hardware y software. Si un nodo falla, otros nodos en el clúster pueden asumir sus funciones para evitar interrupciones en el servicio.

CLÚSTER DE CARGA BALANCEADA

En este tipo de clúster, la carga de trabajo se distribuye entre los nodos para optimizar el rendimiento y evitar la sobrecarga de un nodo en particular.

CLÚSTER DE COMPUTACIÓN DE ALTO RENDIMIENTO (HPC)

Este tipo de clúster se utiliza para realizar cálculos intensivos, como simulaciones científicas o análisis de datos, distribuyendo la carga de trabajo entre múltiples nodos para acelerar el procesamiento.

CLÚSTER DE ALMACENAMIENTO

Estos clústeres se utilizan para proporcionar almacenamiento distribuido y redundante, mejorando tanto la capacidad como la fiabilidad del almacenamiento de datos.

CARACTERÍSTICAS DE ARQUITECTURA DE LOS CLÚSTERES

REDUNDANCIA

En arquitecturas de clústeres, la redundancia se refiere a la duplicación de componentes críticos, ya sea hardware o software, para garantizar la disponibilidad y la fiabilidad de los servicios. Por ejemplo, en un clúster de alta disponibilidad, si un nodo experimenta un fallo, otro nodo puede asumir sus funciones sin interrupción en el servicio. La redundancia puede incluir la duplicación de discos duros, fuentes de alimentación, tarjetas de red y servidores completos. Además, en el software, se pueden implementar sistemas de replicación de datos y servicios para asegurar que siempre haya una copia disponible en caso de fallo de un componente.

ESCALABILIDAD

Las arquitecturas de clústeres son altamente escalables, lo que significa que pueden crecer o disminuir en tamaño según las necesidades de la carga de trabajo. Esto se logra agregando o eliminando nodos del clúster de manera dinámica. Por ejemplo, si una aplicación experimenta un aumento repentino en la demanda, se pueden agregar más nodos al clúster para distribuir la carga y mantener un rendimiento óptimo. Del mismo modo, si la demanda disminuye, los nodos adicionales pueden ser eliminados para ahorrar recursos. Esta capacidad de escalabilidad horizontal permite a las organizaciones adaptarse rápidamente a cambios en la demanda y garantizar un rendimiento consistente.

DISTRIBUCIÓN DE CARGA

En las arquitecturas de clústeres, la distribución de carga se refiere a la forma en que las cargas de trabajo se asignan y distribuyen entre los diferentes nodos del clúster. Esto se hace para optimizar el rendimiento y evitar cuellos de botella. Por ejemplo, en un clúster de balanceo de carga, las solicitudes de los usuarios se distribuyen entre múltiples nodos para garantizar que ningún nodo esté sobrecargado mientras otros permanecen inactivos. Esto ayuda a garantizar un uso eficiente de los recursos disponibles y a mantener un rendimiento consistente incluso durante picos de demanda.

TOLERANCIA A FALLOS

Los clústeres están diseñados para ser tolerantes a fallos, lo que significa que pueden continuar funcionando incluso si uno o más nodos fallan. Esto se logra mediante la redundancia y la capacidad de conmutación por error. Por ejemplo, en un clúster de alta disponibilidad, si un nodo experimenta un fallo, los servicios pueden ser automáticamente redirigidos a otro nodo que esté en buen estado. Esta capacidad de tolerancia a fallos ayuda a garantizar la disponibilidad continua de los servicios y a minimizar el tiempo de inactividad en caso de fallos de hardware o software.

PARALELISMO

En las arquitecturas de clústeres, el paralelismo se refiere a la capacidad de realizar tareas de manera simultánea y distribuida entre múltiples nodos del clúster. Esto se logra dividiendo el trabajo en partes más pequeñas y asignándolas a diferentes nodos que trabajan de manera independiente pero coordinada. Por ejemplo, en un clúster de procesamiento de datos, las tareas de análisis pueden ser divididas en subprocesos que se ejecutan simultáneamente en diferentes nodos para acelerar el tiempo de procesamiento. El paralelismo permite aprovechar al máximo los recursos disponibles y mejorar la eficiencia en la ejecución de tareas.

TIPOS DE PROCESAMIENTO DE CLÚSTERES

PROCESAMIENTO PARALELO

En este tipo de procesamiento, múltiples nodos en el clúster trabajan en conjunto para realizar una tarea dividida en partes más pequeñas, lo que acelera el tiempo de procesamiento.

PROCESAMIENTO DISTRIBUIDO

En este enfoque, los datos y las tareas se distribuyen entre los nodos del clúster para procesarse de manera independiente y luego se combinan los resultados.

PROCESAMIENTO DE CARGA DE TRABAJO

Este tipo de procesamiento implica distribuir la carga de trabajo de manera equitativa entre los nodos del clúster para garantizar un uso eficiente de los recursos disponibles.

CONCEPTO DE ANÁLISIS DE RIESGO

El análisis de riesgo en entornos de tecnología de la información (TI) se refiere al proceso de identificar, evaluar y mitigar las amenazas y vulnerabilidades que pueden afectar la seguridad y el funcionamiento de los sistemas y datos de una organización. Cuando se aplica al contexto de los clústeres, el análisis de riesgo se centra en comprender los riesgos específicos asociados con la implementación y operación de clústeres en una infraestructura de TI.

El análisis de riesgo en el contexto de los clústeres implica los siguientes pasos:

IDENTIFICACIÓN DE RIESGOS

Consiste en identificar las posibles amenazas y vulnerabilidades que pueden afectar a los clústeres, como fallos de hardware, errores de configuración, ataques de seguridad, etc.

EVALUACIÓN DE RIESGOS

Se evalúa la probabilidad de que ocurran los riesgos identificados y el impacto potencial que tendrían en el funcionamiento del clúster y en el negocio de la organización.

PRIORIZACIÓN DE RIESGOS

Se priorizan los riesgos identificados según su gravedad y probabilidad, para centrar los recursos en mitigar los riesgos más críticos primero.

MITIGACIÓN DE RIESGOS

Se implementan medidas y controles de seguridad para reducir la probabilidad de ocurrencia de los riesgos identificados y minimizar su impacto en caso de que ocurran.

MONITOREO Y REVISIÓN CONTINUA

Se establece un proceso de monitoreo constante para detectar nuevos riesgos y evaluar la efectividad de las medidas de mitigación implementadas. Además, se revisa periódicamente el análisis de riesgo para mantenerlo actualizado y adaptarlo a los cambios en el entorno tecnológico y las amenazas emergentes.

ANÁLISIS DE IMPACTO Y GESTIÓN DE PRIORIDAD EN LOS CLÚSTERES

El análisis de impacto es crucial en la gestión de incidentes en clústeres, ya que permite evaluar las consecuencias potenciales de un incidente en el funcionamiento de los sistemas y en el negocio de la organización. Aquí hay algunos aspectos clave sobre cómo se evalúa y gestiona el impacto de posibles incidentes en los clústeres.

IDENTIFICACIÓN DE ACTIVOS CRÍTICOS

El primer paso en el análisis de impacto es identificar los activos críticos que están alojados en los clústeres, como aplicaciones empresariales, bases de datos, servicios web, entre otros. Estos activos son fundamentales para el funcionamiento del negocio y su interrupción podría tener un impacto significativo.

EVALUACIÓN DE IMPACTO

Una vez identificados los activos críticos, se evalúa el impacto potencial de un incidente en estos activos. Esto implica determinar cómo afectaría la interrupción de los servicios o la pérdida de datos a la operación del negocio, la productividad de los empleados, la satisfacción del cliente, la reputación de la empresa, entre otros aspectos.

ANÁLISIS DE ESCENARIOS

Se analizan diferentes escenarios de incidentes, desde fallas de hardware o software hasta ataques cibernéticos o desastres naturales, para comprender cómo cada uno de ellos podría afectar los clústeres y los activos críticos asociados.

PRIORIZACIÓN DE ACCIONES DE MITIGACIÓN

Con base en la evaluación de impacto, se priorizan las acciones de mitigación para reducir el impacto potencial de los incidentes. Las medidas de mitigación pueden incluir la implementación de redundancia de hardware y software, la mejora de la seguridad, la realización de copias de seguridad y la planificación de la continuidad del negocio.

ESTRATEGIAS DE GESTIÓN DE PRIORIDAD

CLASIFICACIÓN DE INCIDENTES

Los incidentes se clasifican según su gravedad y su impacto en el negocio. Se establecen criterios claros para determinar la prioridad de los incidentes, como el nivel de afectación de los servicios, la criticidad de los activos afectados y el potencial daño a la reputación de la empresa.

IMPLEMENTACIÓN DE UN MODELO DE PRIORIZACIÓN

Se establece un modelo de priorización que asigna niveles de prioridad a los incidentes en función de su gravedad y urgencia. Esto permite a los equipos de respuesta a incidentes centrarse en abordar primero los incidentes que representan un mayor riesgo para el negocio.

ASIGNACIÓN DE RECURSOS

Se asignan los recursos necesarios, como personal técnico, herramientas y equipos, de acuerdo con la prioridad de los incidentes. Los recursos se distribuyen de manera eficiente para garantizar una respuesta rápida y efectiva a los incidentes más críticos.

COMUNICACIÓN Y COORDINACIÓN

Se establece un proceso claro de comunicación y coordinación entre los equipos de respuesta a incidentes, las partes interesadas internas y externas, para garantizar una gestión efectiva de la prioridad de las acciones. Esto incluye la notificación oportuna de incidentes críticos y la colaboración en la resolución de problemas.

RECUPERACIÓN DE DESASTRES: RPO Y RTO

RPO (RECOVERY POINT OBJECTIVE)

El RPO se refiere al punto en el tiempo hasta el cual una organización está dispuesta a tolerar la pérdida de datos en caso de un desastre o interrupción. En otras palabras, representa la cantidad máxima de datos que una empresa puede permitirse perder sin que esto afecte significativamente sus operaciones. Por ejemplo, si el RPO es de una hora, significa que la empresa puede tolerar perder hasta una hora de datos en caso de un incidente antes de que la pérdida sea considerada inaceptable.

RTO (RECOVERY TIME OBJECTIVE)

El RTO se refiere al tiempo máximo permitido para restaurar los servicios y la infraestructura de TI después de un desastre o interrupción. Es el tiempo que tarda una organización en recuperarse y volver a un estado operativo funcional después de un evento adverso. Por ejemplo, si el RTO es de cuatro horas, significa que la empresa debe restaurar sus sistemas y servicios dentro de ese período de tiempo para minimizar el impacto en las operaciones comerciales.

RPO Y RTO EN CLÚSTERES

RPO EN CLÚSTERES

Para minimizar la pérdida de datos en los clústeres, se implementan técnicas de replicación de datos en tiempo real o periódico entre los nodos del clúster. Esto garantiza que, en caso de un fallo, la cantidad de datos perdidos sea igual o inferior al RPO establecido.

RTO EN CLÚSTERES

Para minimizar el tiempo de inactividad en caso de un desastre, se implementan medidas de recuperación rápida, como la virtualización de servidores, la conmutación por error automática y la automatización de procesos de recuperación. Estas técnicas permiten restaurar rápidamente los servicios y la infraestructura del clúster dentro del RTO definido.

HERRAMIENTAS Y TÉCNICAS PARA LA RECUPERACIÓN DE DESASTRES EN CLÚSTERES

REPLICACIÓN DE DATOS EN TIEMPO REAL O PERIÓDICA

Las herramientas de replicación de datos permiten duplicar los datos entre los nodos del clúster de manera continua o en intervalos regulares. Esto garantiza la disponibilidad de datos actualizados en caso de un fallo en un nodo.

VIRTUALIZACIÓN DE SERVIDORES

La virtualización de servidores permite crear réplicas virtuales de los servidores físicos, lo que facilita la rápida recuperación de los servicios en un entorno de hardware alternativo en caso de un desastre.

CONMUTACIÓN POR ERROR AUTOMÁTICA

Las soluciones de conmutación por error automática detectan automáticamente los fallos en los nodos del clúster y redirigen el tráfico de manera transparente a nodos alternativos que estén operativos. Esto ayuda a minimizar el tiempo de inactividad y mantener la continuidad del servicio.

AUTOMATIZACIÓN DE PROCESOS DE RECUPERACIÓN

La automatización de procesos de recuperación permite ejecutar automáticamente las tareas necesarias para restaurar los servicios del clúster, como la restauración de copias de seguridad, la configuración de redes y la puesta en marcha de aplicaciones.

PRUEBAS DE RECUPERACIÓN DE DESASTRES (DR)

Realizar pruebas periódicas de recuperación de desastres ayuda a garantizar que las herramientas y técnicas de recuperación sean efectivas y puedan cumplir con los objetivos de RPO y RTO establecidos.

PLANES DE CONTINUIDAD

IMPORTANCIA DE LOS PLANES DE CONTINUIDAD DEL NEGOCIO

Los planes de continuidad del negocio son fundamentales para asegurar la supervivencia y la resiliencia de una organización frente a diversos tipos de interrupciones, como desastres naturales, fallos técnicos, ciberataques o emergencias sanitarias. Aquí se destaca por qué son importantes estos planes en entornos empresariales y su relación con los clústeres.

MANTENIMIENTO DE LA OPERATIVIDAD

Los planes de continuidad del negocio permiten a las organizaciones mantener la operatividad de sus servicios críticos incluso en situaciones adversas. Esto es especialmente relevante en entornos empresariales donde la interrupción de servicios puede resultar en pérdidas financieras significativas, daño a la reputación y pérdida de clientes.

MINIMIZACIÓN DEL TIEMPO DE INACTIVIDAD

La rápida recuperación de servicios es esencial para minimizar el tiempo de inactividad y reducir el impacto en las operaciones comerciales. Los planes de continuidad del negocio proporcionan procedimientos claros y recursos dedicados para restaurar los servicios de manera eficiente y efectiva.

PROTECCIÓN DE DATOS Y ACTIVOS

Los planes de continuidad del negocio incluyen medidas para proteger los datos y activos críticos de la organización durante y después de un incidente. Esto es especialmente relevante en entornos de clústeres, donde se gestionan grandes volúmenes de datos y servicios compartidos entre múltiples nodos.

CUMPLIMIENTO REGULATORIO Y CONTRACTUAL

Muchas industrias están sujetas a regulaciones y requisitos contractuales que exigen la implementación de planes de continuidad del negocio. Cumplir con estas

regulaciones y compromisos contractuales es crucial para evitar multas, sanciones y daños legales.

PASOS PARA EL DISEÑO DE PLANES DE CONTINUIDAD

IDENTIFICACIÓN DE SERVICIOS CRÍTICOS

Se identifican los servicios y aplicaciones críticas que se ejecutan en los clústeres y se evalúa su importancia para las operaciones comerciales.

ANÁLISIS DE RIESGOS

Se realiza un análisis de riesgos específico para los clústeres, identificando las amenazas y vulnerabilidades que podrían afectar su funcionamiento y los servicios asociados.

ESTABLECIMIENTO DE OBJETIVOS DE RECUPERACIÓN

Se definen los objetivos de recuperación, incluyendo el RPO y el RTO, para cada servicio crítico alojado en los clústeres.

DESARROLLO DE ESTRATEGIAS DE RECUPERACIÓN

Se diseñan estrategias de recuperación específicas para los clústeres, que incluyen medidas de mitigación de riesgos, procedimientos de respuesta a incidentes y planes de recuperación de desastres.

IMPLEMENTACIÓN Y PRUEBA

Se implementan los planes de continuidad del negocio y se realizan pruebas periódicas para asegurar su eficacia y capacidad de respuesta ante diferentes escenarios de desastre.

MANTENIMIENTO Y ACTUALIZACIÓN

Los planes de continuidad del negocio se mantienen y actualizan regularmente para reflejar los cambios en la infraestructura de clústeres, los riesgos emergentes y las mejores prácticas de la industria.

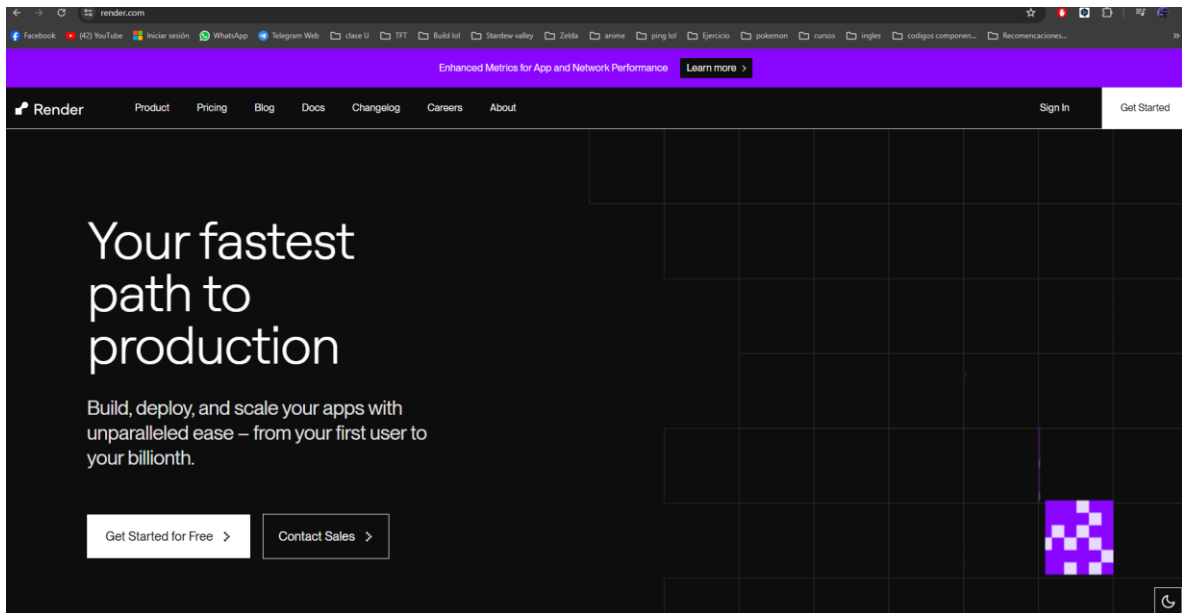
HOSTING

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.


Google

Iniciar sesión con Google




Selecciona una cuenta

para ir a [Render](#)



Brayan Alexander Puentes Martinez
alepues1@gmail.com

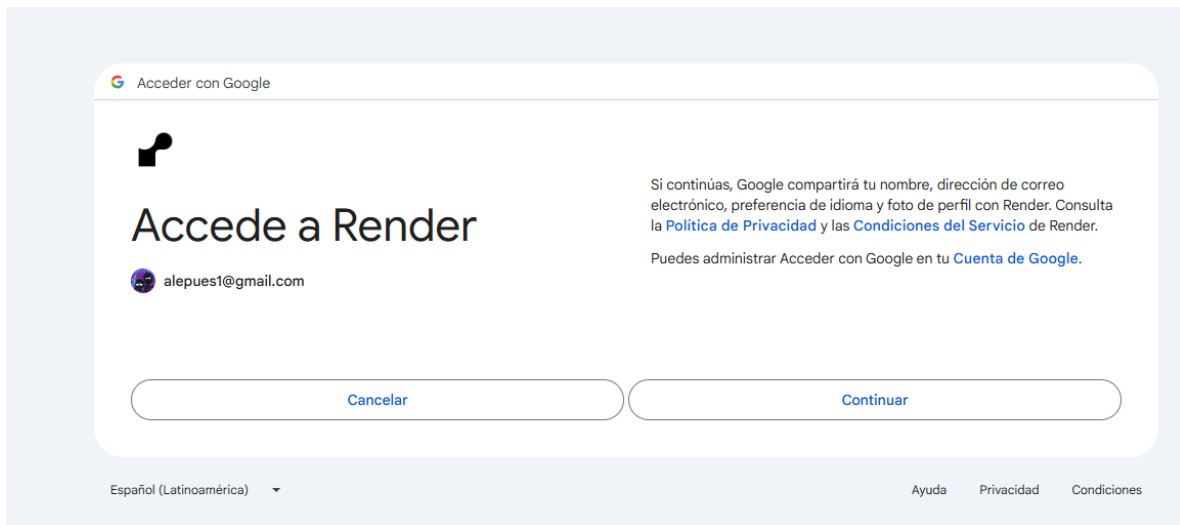
 Usar otra cuenta

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.

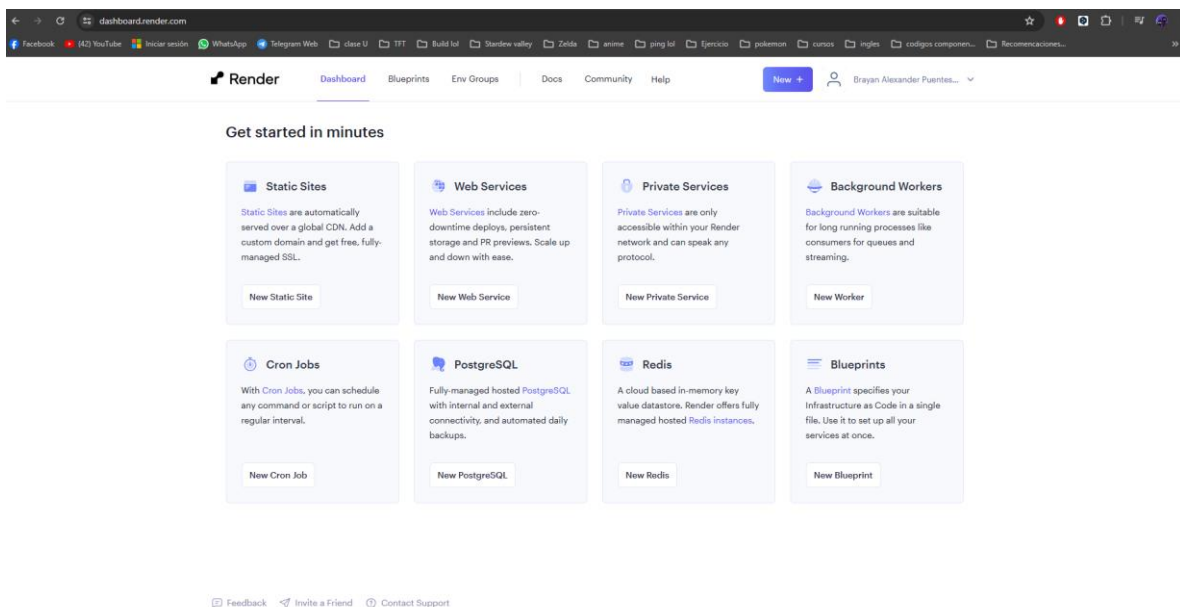
Español (España) ▼

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


Se aceptan los terminos 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.



Web Services

Web Services include zero-downtime deploys, persistent storage and PR previews. Scale up and down with ease.

New Web Service

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

Render

Panel

Planos

Grupos ambientales

Documentos

Comunidad

Ayuda

Nuevo +

Brayan Alejandro Puentes ...

Crear un nuevo servicio web

Conecte un repositorio de Git o utilice una imagen existente.


¿Cómo le gustaría implementar su servicio web ?

☒ Construya e implemente desde un repositorio Git
Conecte un repositorio de GitHub o GitLab.

☐ Implementar una imagen existente desde un registro **AVANZADO**
Extraiga una imagen pública de cualquier registro o una imagen privada de Docker Hub, GitHub o GitLab.


Próximo

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

Render

[Panel](#)[Planos](#)[Grupos ambientales](#)[Documentos](#)[Comunidad](#)[Ayuda](#)

Nuevo +


Brayan Alejandro Puentes ...


Crear un nuevo servicio web


Conecte su repositorio Git o utilice una URL de repositorio público existente.


Conectar un repositorio

Conecte su cuenta de Render a GitHub, GitLab o Bitbucket para comenzar a usar sus repositorios existentes para nuevos servicios.


Conectar GitHub

Conectar GitLab


Conectar Bitbucket

GitHub

+ Conectarcuenta

GitLab

+ Conectarcuenta

Bitbucket

+ Conectarcuenta

Repositorio público de Git


Utilice un repositorio público ingresando la URL a continuación. Funciones como [vistas previas de PR](#) e [implementación automática](#) no están disponibles si el repositorio no se ha configurado para renderizar.


Continuar



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

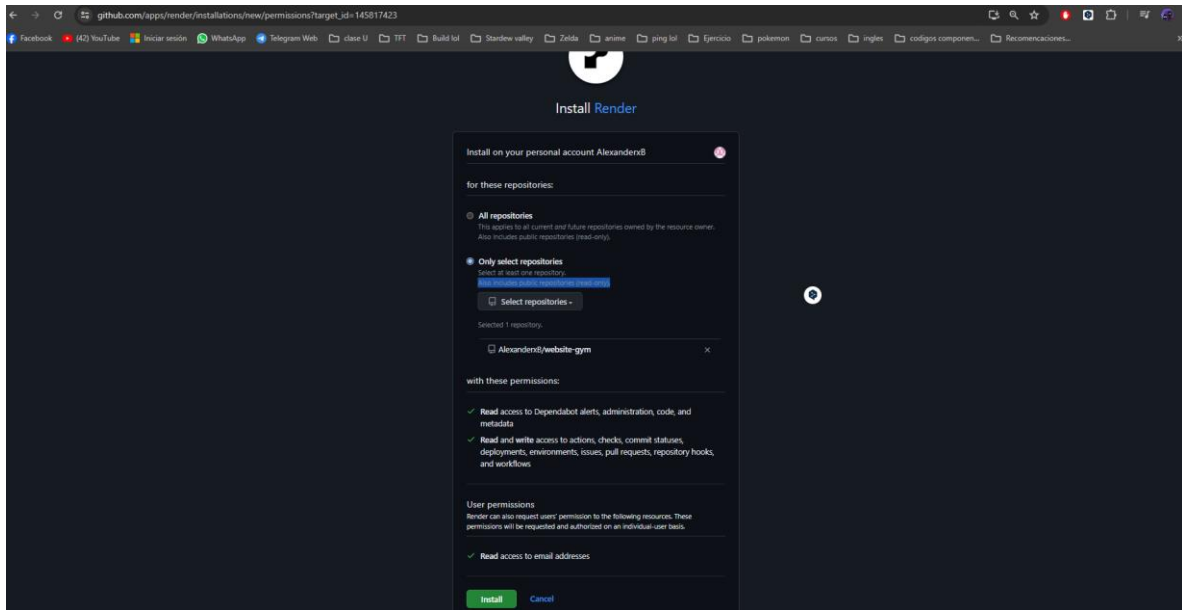
Authorizing will redirect to
<https://dashboard.render.com>

 Not owned or
operated by GitHub

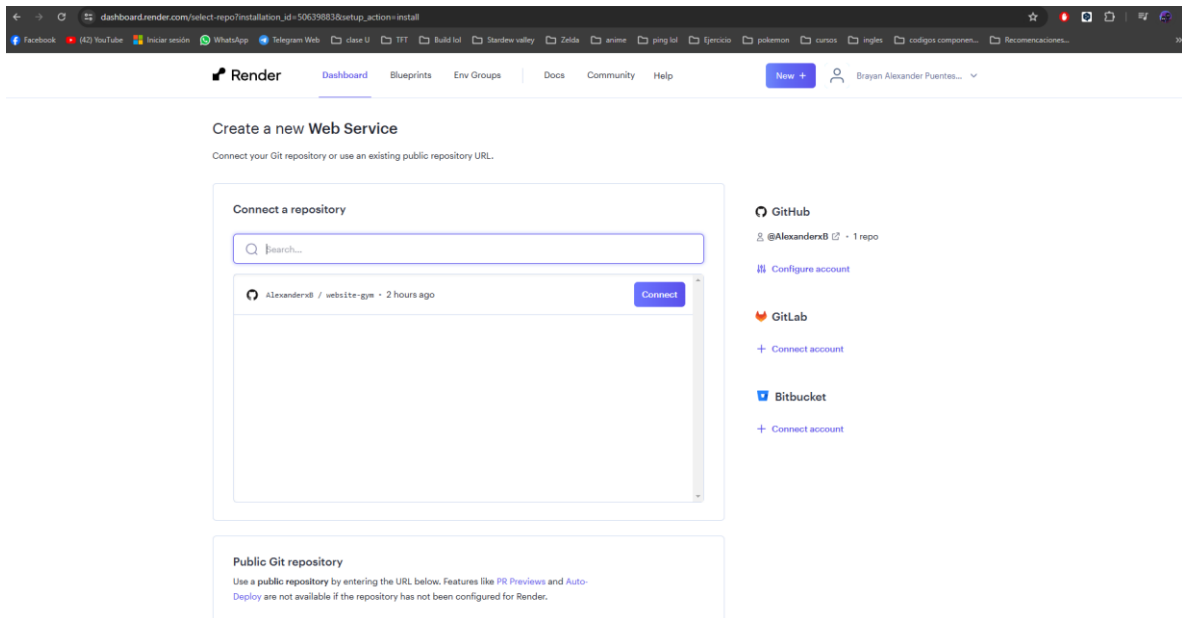
 Created 6 years ago

 More than 1K
GitHub users

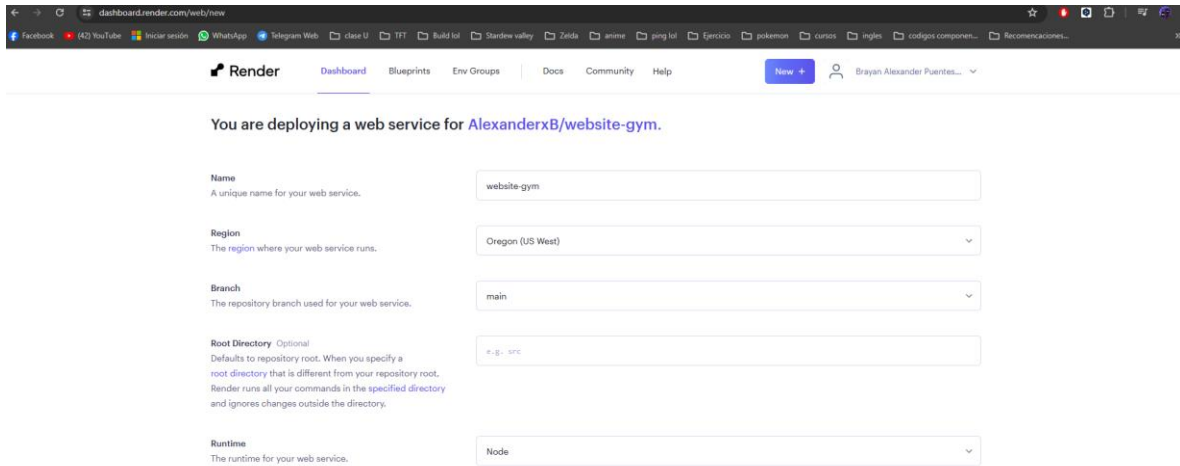
Se le da permiso únicamente para que acceda render al repositorio que dice website-gym



Se ha creado con éxito la creación de un web service



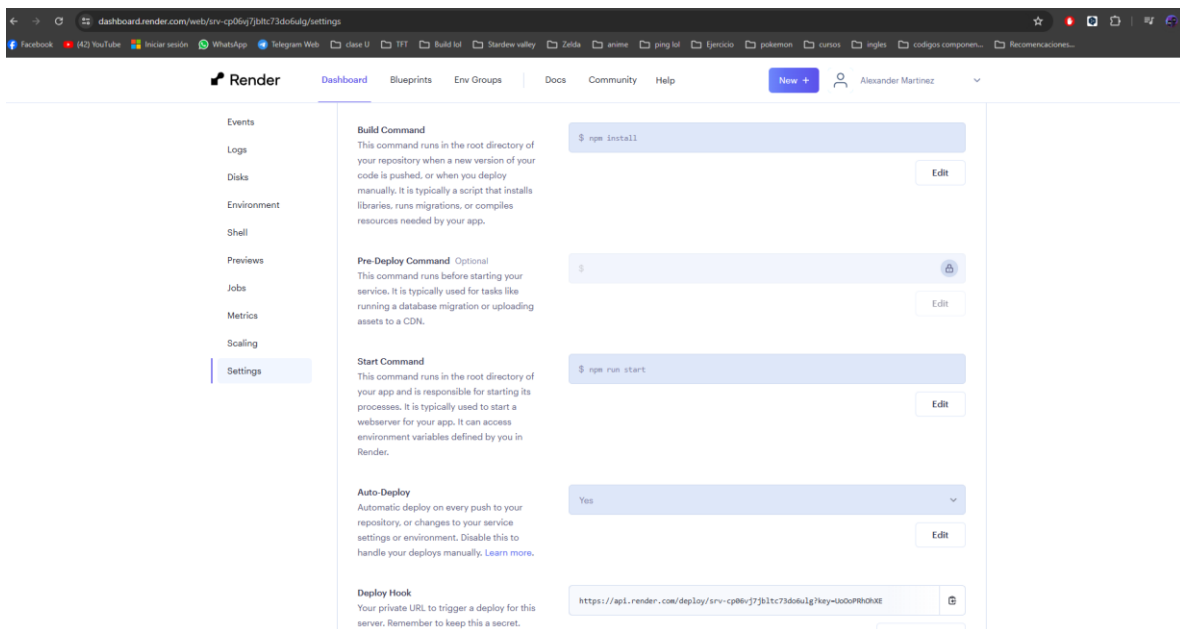
Al dar click al web service creado, se empieza a realizar la configuración de este para poder desplegar el front end del proyecto “Hypertrophy Fitness”



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, defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.
- 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, This command runs before starting your service. It is typically used for tasks like running a database migration or uploading assets to a CDN.
- Start Command:** \$ npm run start
- Auto-Deploy:** Yes
- Deploy Hook:** https://api.render.com/deploy/srv-cp86v37jbtcc73dolu1g7hey-4a0p8h0k8e

Instance Type

For hobby projects

Free
\$0 / month

512 MB (RAM)
0.1 CPU

Upgrade to enable more features
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.

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

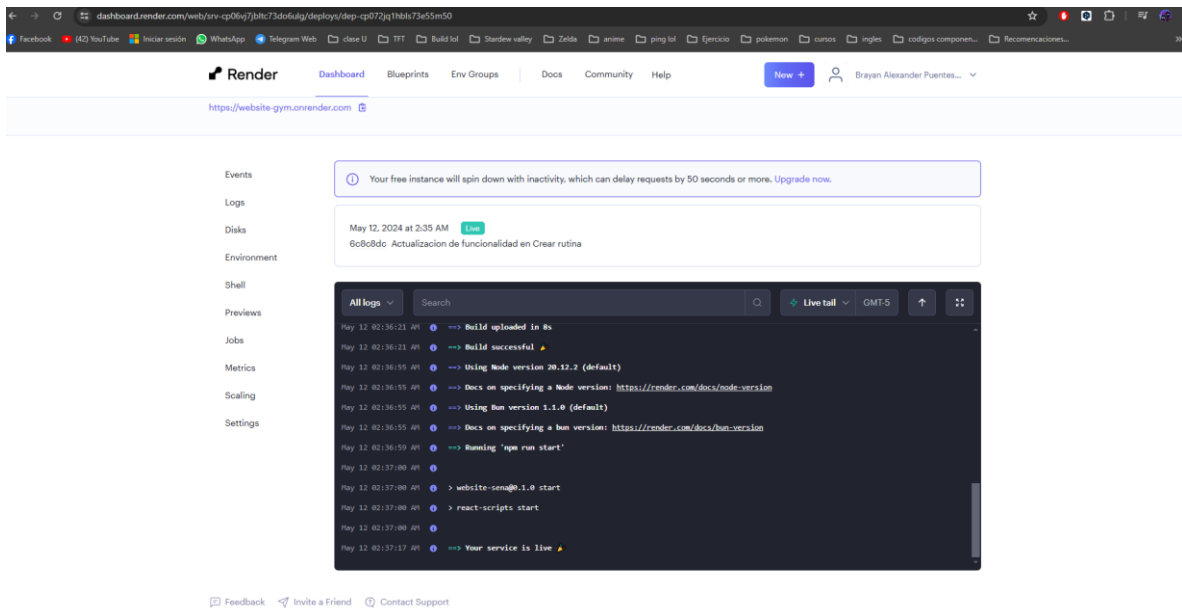
Starter \$7 / month	512 MB (RAM) 0.5 CPU	Standard \$25 / month	2 GB (RAM) 1 CPU
Pro \$85 / month	4 GB (RAM) 2 CPU	Pro Plus \$175 / month	8 GB (RAM) 4 CPU
Pro Max \$225 / month	16 GB (RAM) 4 CPU	Pro Ultra \$450 / month	32 GB (RAM) 8 CPU

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

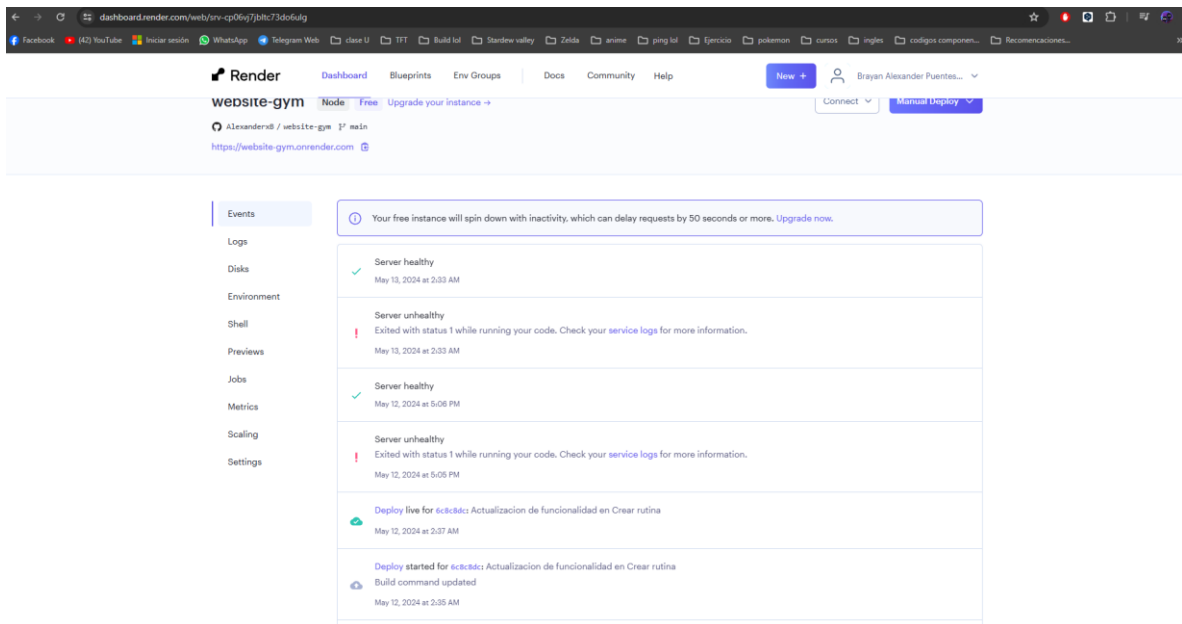
Finalmente, se da click sobre el boton que dice “Create web service”

The screenshot shows the Render dashboard interface. At the top, there's a navigation bar with 'Render' logo and links for Dashboard, Blueprints, Env Groups, Docs, Community, and Help. A 'Now + New' button is visible. Below the navigation bar, the 'Instance Type' section is displayed, showing various pricing tiers from Free to Pro Ultra. Below this, the 'Environment Variables' section is visible, with a table for adding variables. At the bottom, there's an 'Advanced' dropdown and a prominent blue 'Create Web Service' button. The footer includes links for Feedback, Invite a Friend, and Contact Support.

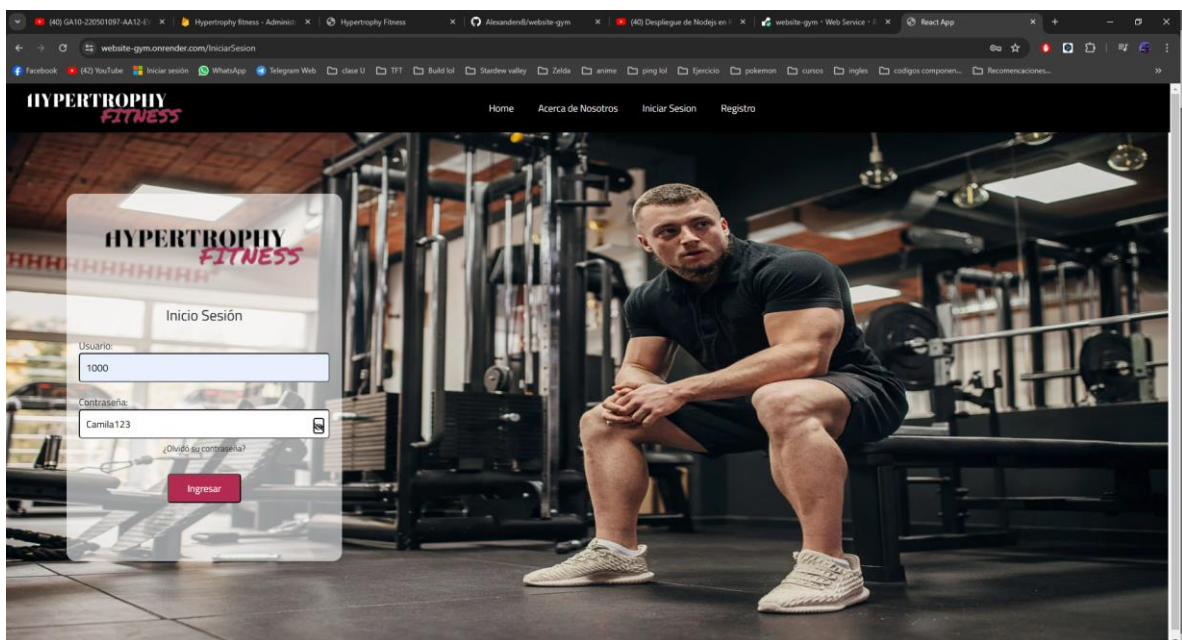
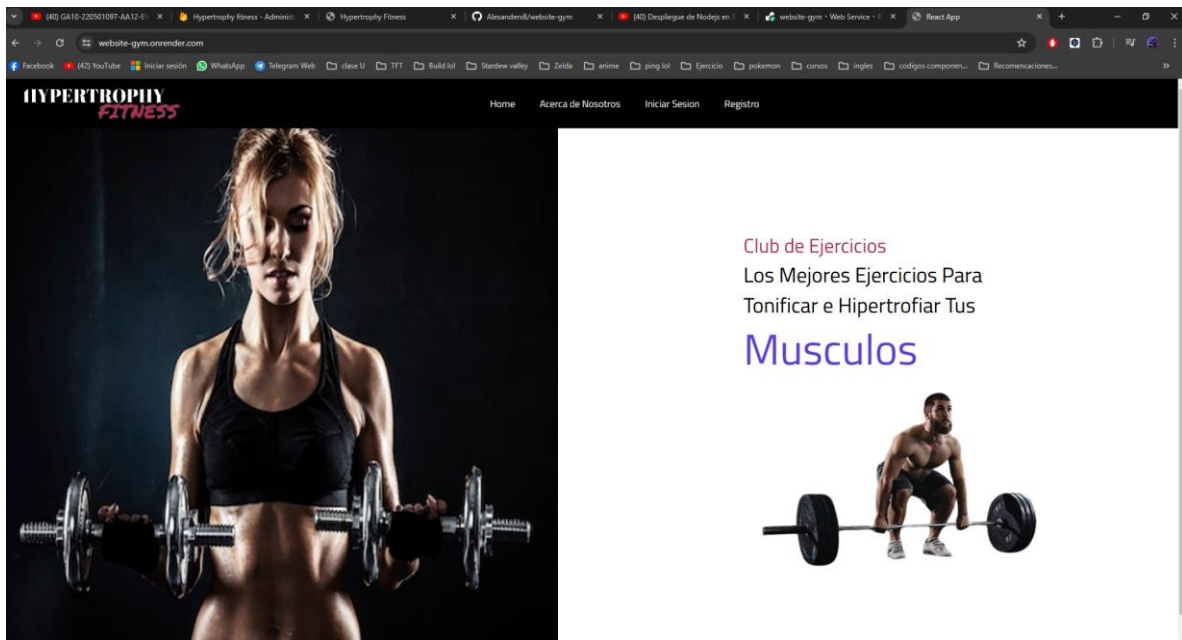
En el lado izquierdo, se selecciona la opción "Logs". Aquí se puede observar el proceso de build, el despliegue, posibles errores y otros detalles relevantes.

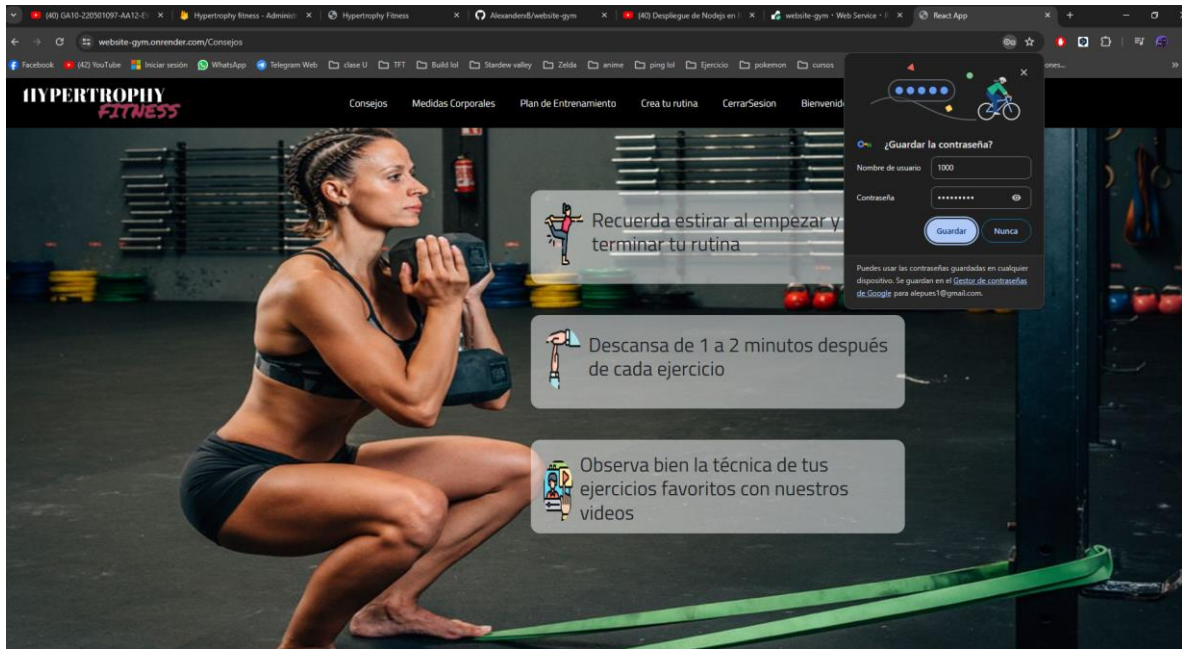


En esta imagen se evidencia con el icono de una nube verde que el despliegue salió exitoso.

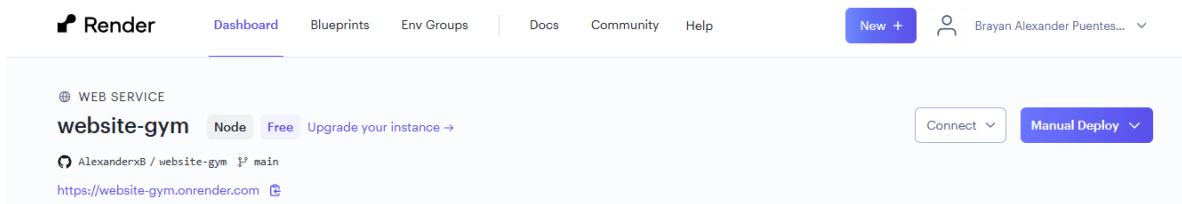


En las imágenes se muestra que al usar la URL generada por Render, se puede verificar que el despliegue del front end fue exitoso.





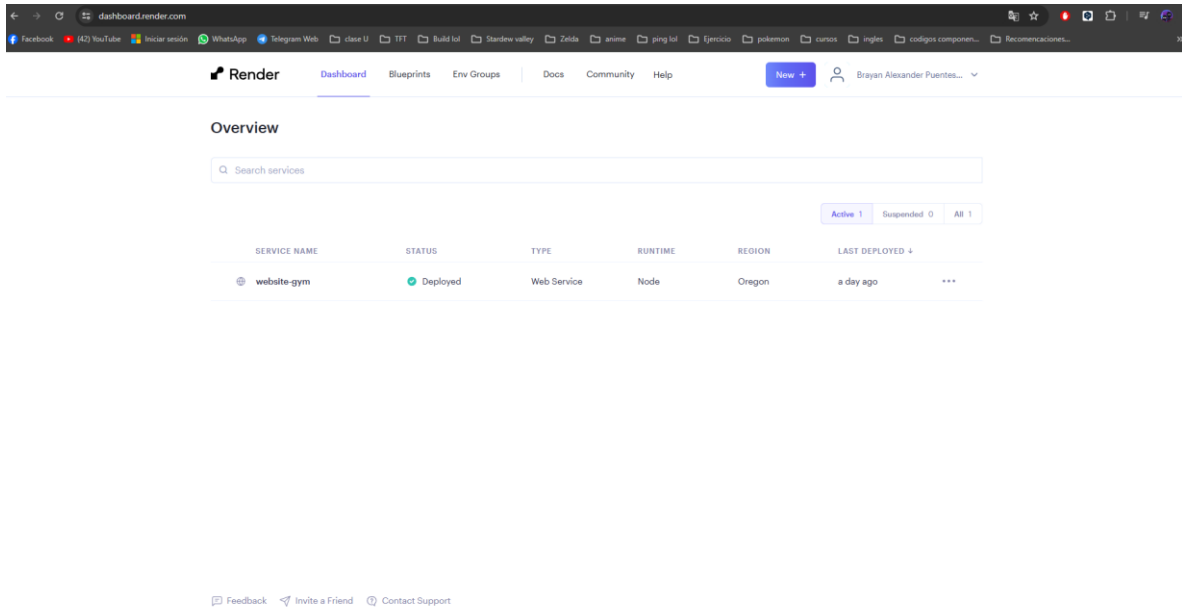
En esta imagen se evidencia la URL que Render genera



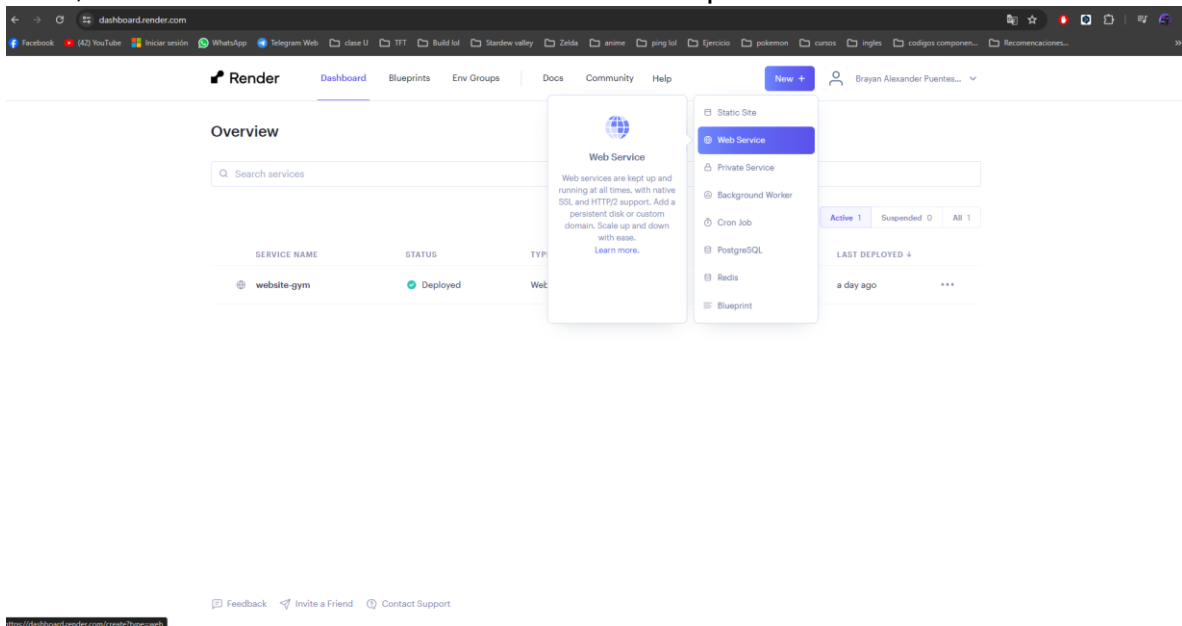
URL DESPLIEGUE FRONT END: <https://website-gym.onrender.com>

BACK END

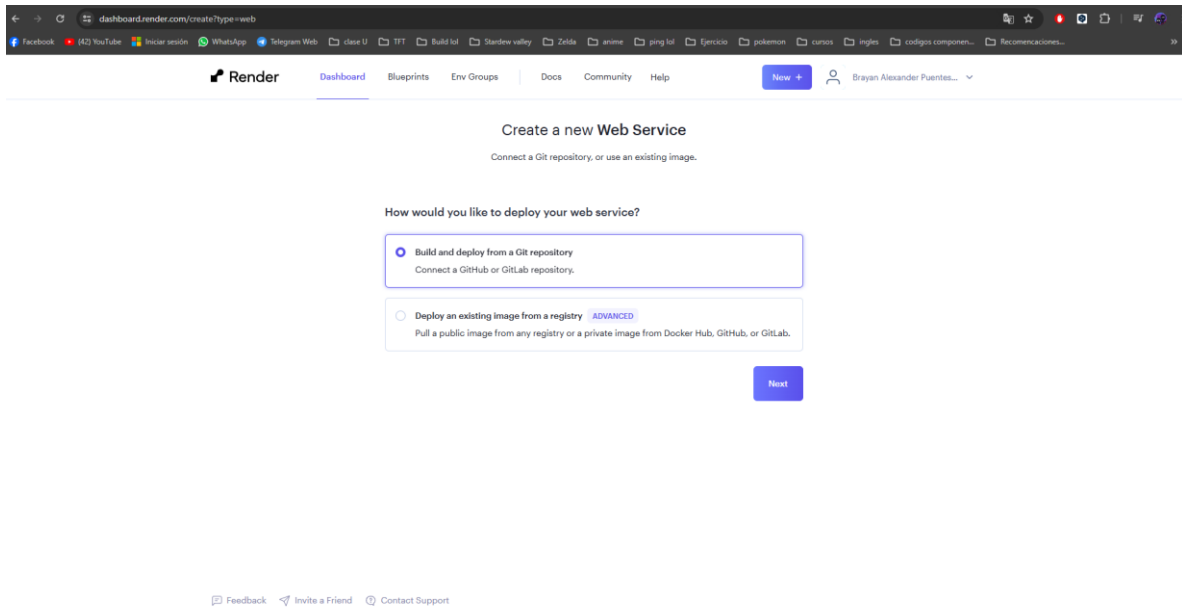
Ahora, para realizar el despliegue del back end, se sigue el mismo procedimiento descrito anteriormente. En esta imagen se evidencia el despliegue exitoso del front end.



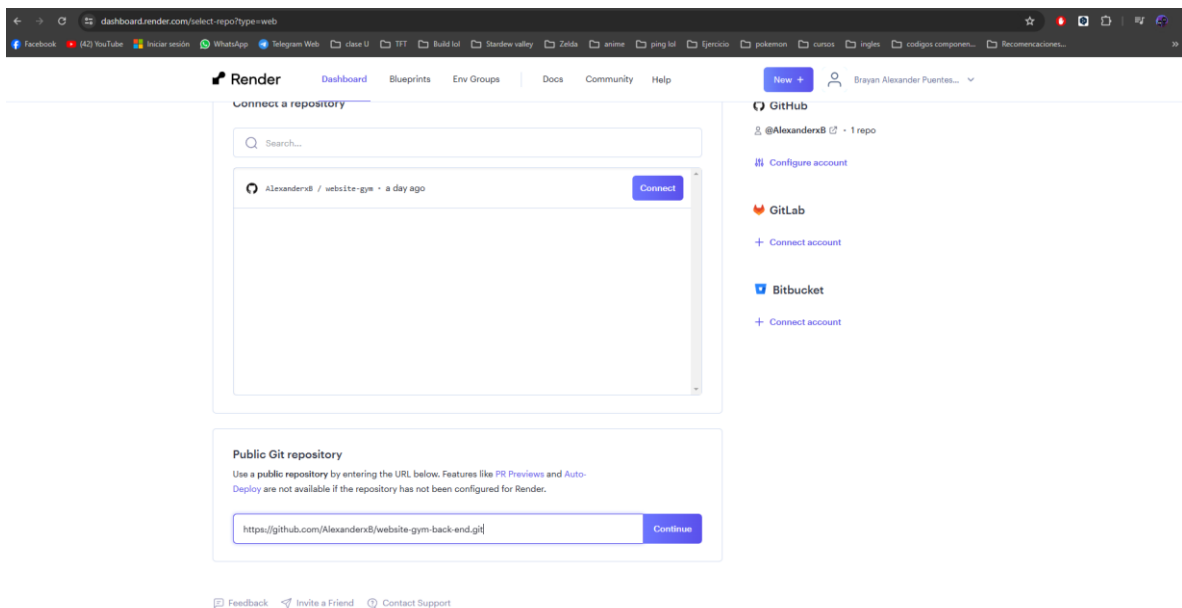
Al hacer clic en el botón que dice "New", se despliegan varias opciones. En este caso, se selecciona la opción "Web Service".



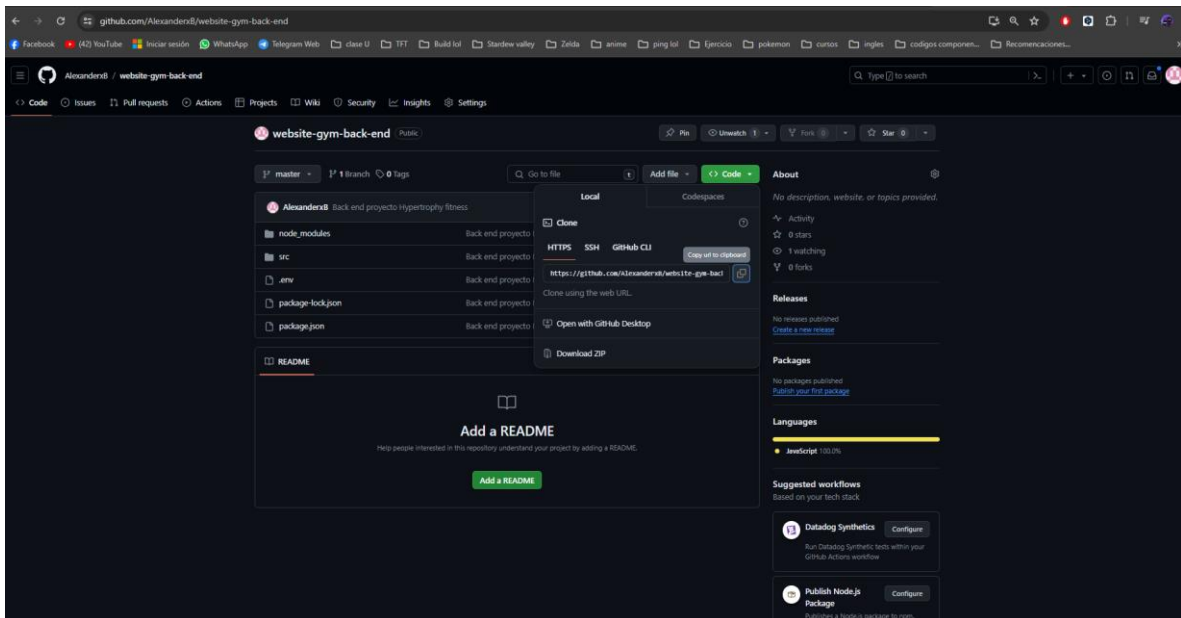
Se vuelve a crear el web service para realizar el despliegue del back, conectando el GitHub



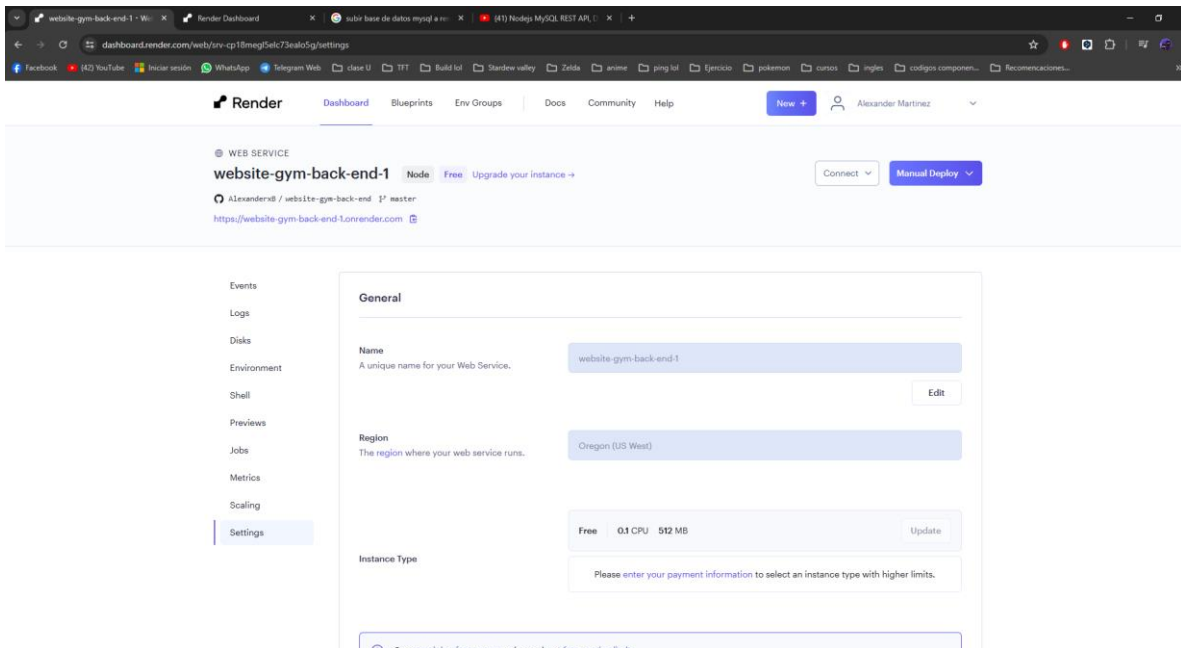
En la sección de "Public Git repository", se introduce la URL del repositorio de GitHub.

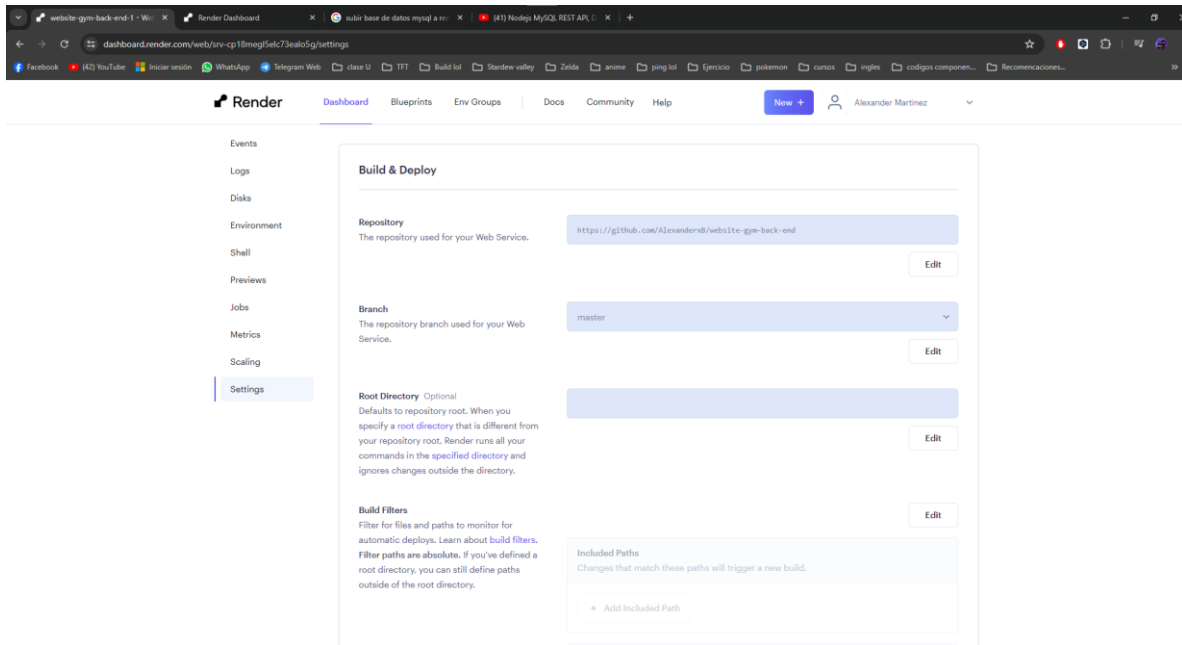


De aquí se genera la URL mencionada en el paso anterior.

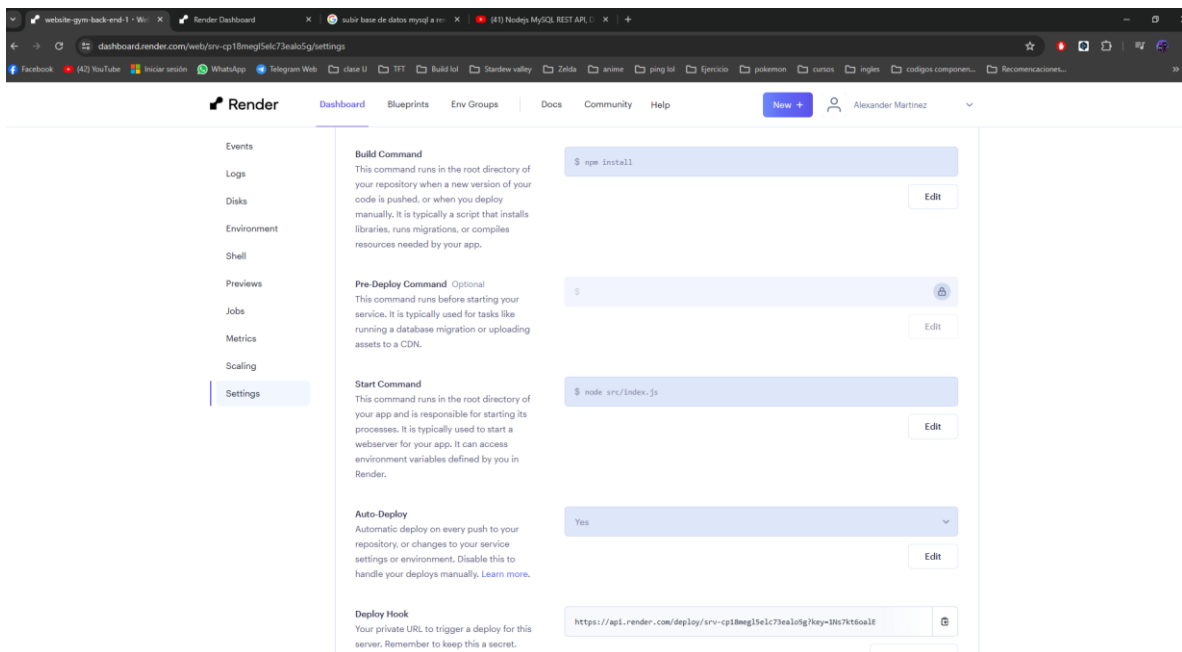


Se vuelve a realizar la configuración para realizar el despliegue y el build

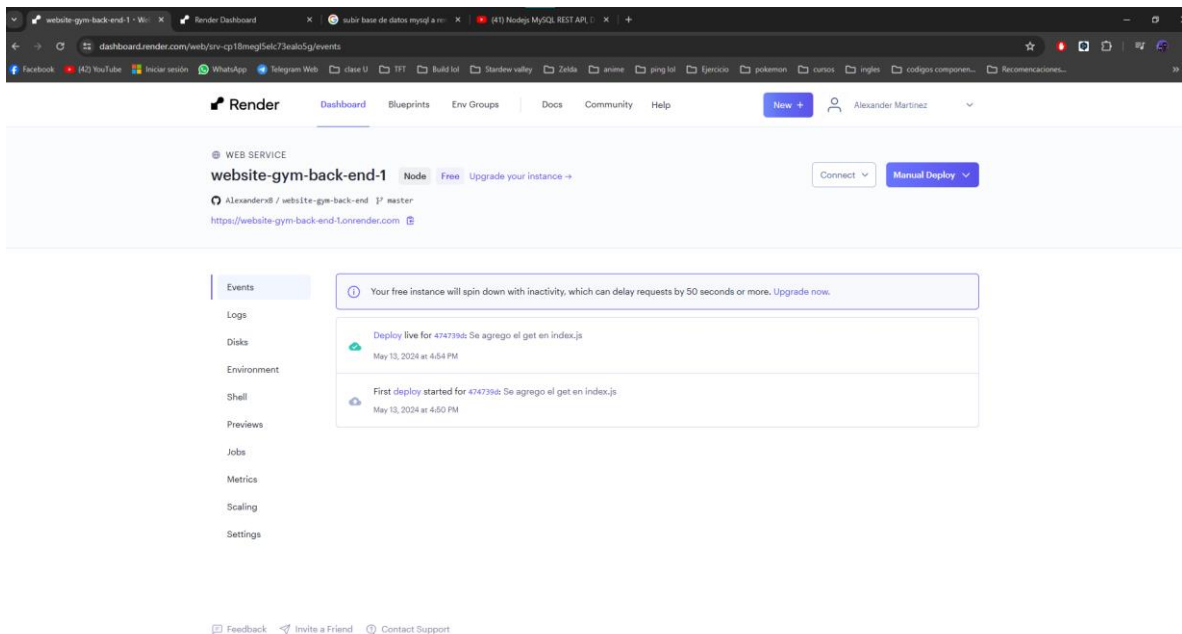




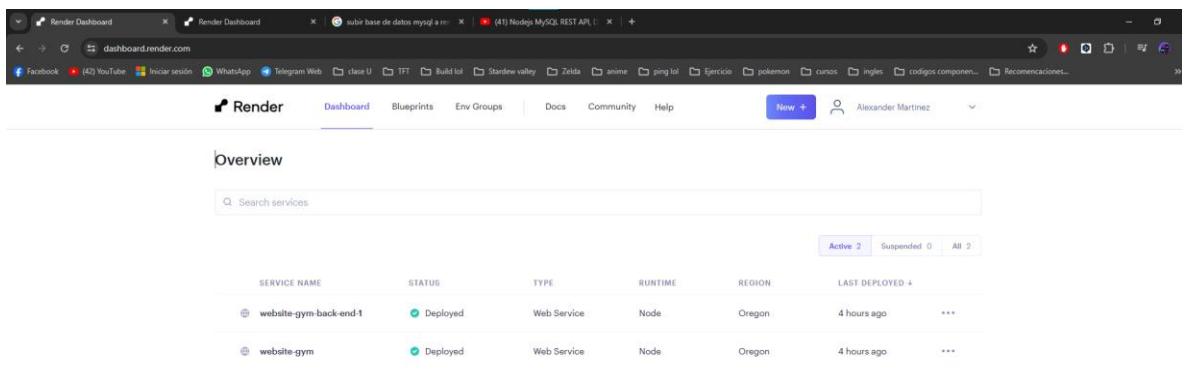
En la sección del build se ingresa el comando npm install y en el apartado de start comando se ingresa el comando node src/index.js



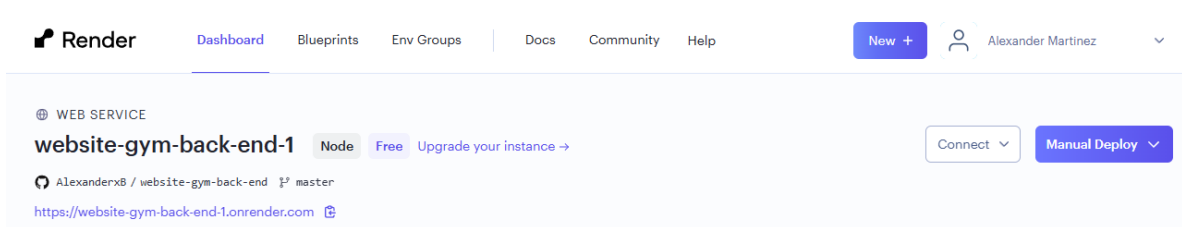
Se evidencia en Events que el despliegue se realizo de forma correcta



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

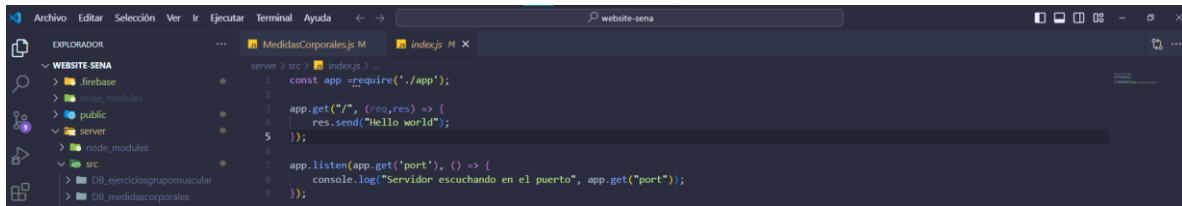


Aquí se puede evidenciar la URL del back end



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

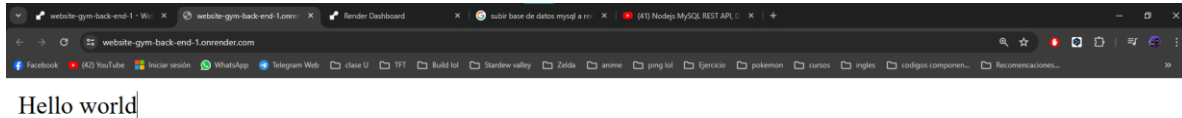
correctamente.



The screenshot shows a VS Code editor with a project named 'website-sena'. The Explorer sidebar on the left shows the file structure: 'website-sena' (root), 'libreria', 'node_modules', 'public', 'server', and 'src'. The 'server' folder is expanded, showing 'index.js'. The main editor displays the code in 'index.js':

```
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 });
```

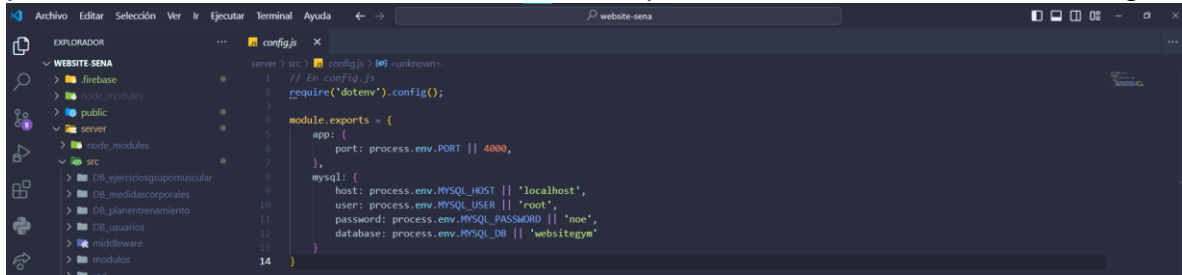
En esta imagen se evidencia el mensaje hello world, es decir, el despliegue esta funcionando correctamente.



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.

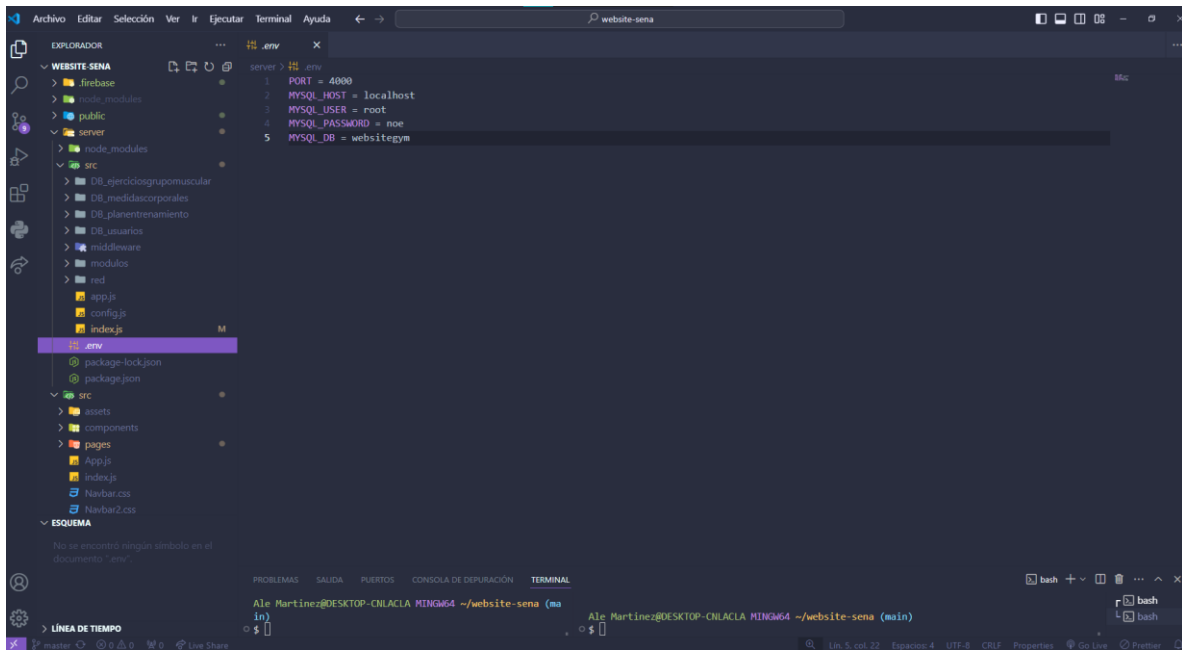


```
server > src > config.js (x) <unknown>
1 // En config.js
2 require("dotenv").config();
3
4 module.exports = {
5   app: {
6     port: process.env.PORT || 4000,
7   },
8   mysql: {
9     host: process.env.MYSQL_HOST || "localhost",
10    user: process.env.MYSQL_USER || "root",
11    password: process.env.MYSQL_PASSWORD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

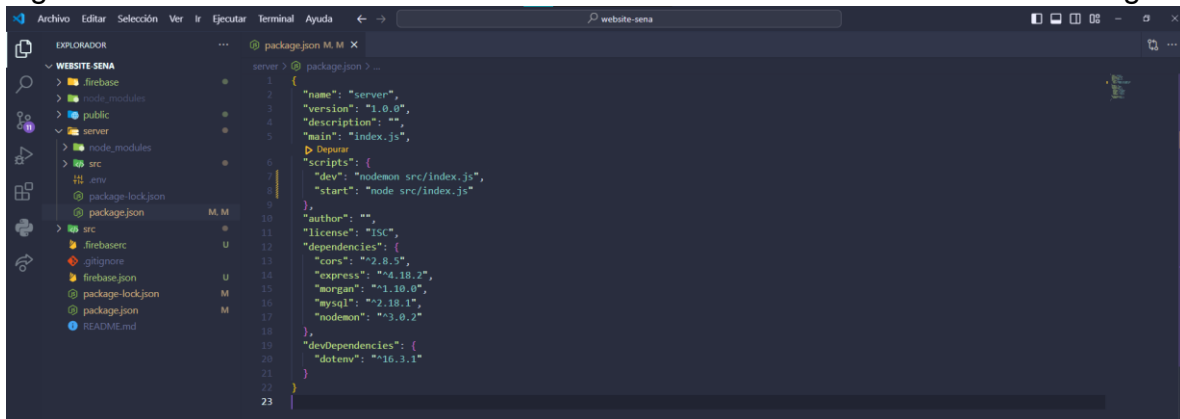
Primero, necesitamos instalar un módulo en la terminal de Visual Studio Code. Podemos hacerlo con el comando `npm i dotenv`.

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



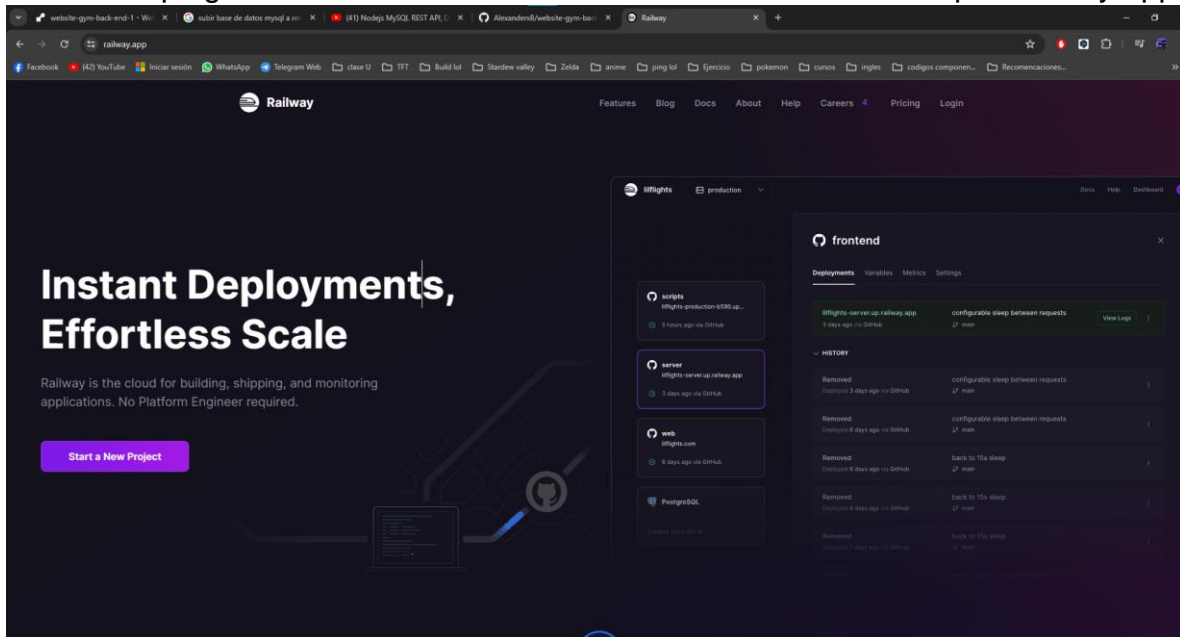
```
server > .env
1 PORT = 4000
2 MYSQL_HOST = localhost
3 MYSQL_USER = root
4 MYSQL_PASSWORD = noe
5 MYSQL_DB = websitegm
```

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:

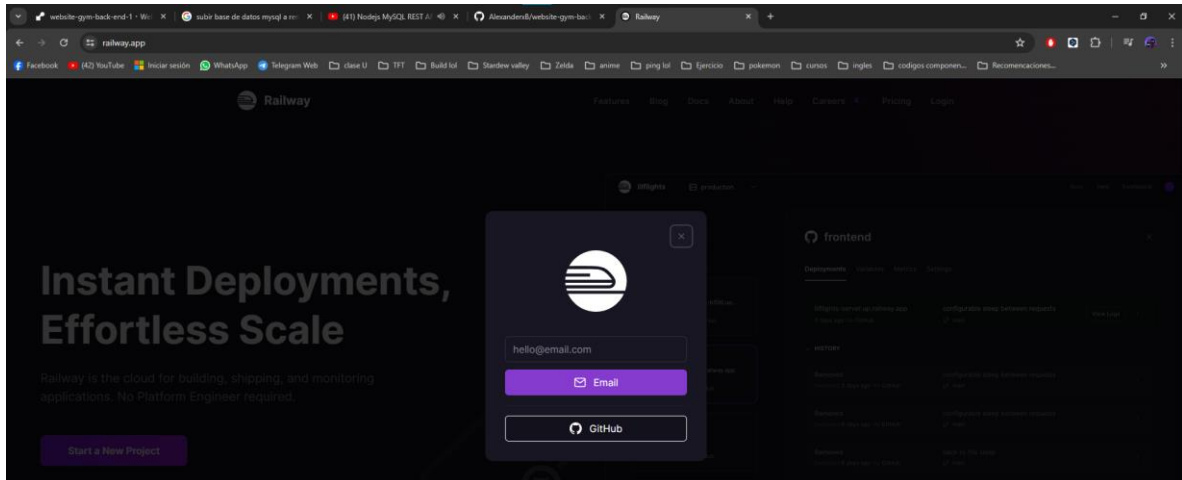


```
1 {
2   "name": "server",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "dev": "nodemon src/index.js",
8     "start": "node src/index.js"
9   },
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "cors": "^2.8.5",
14    "express": "^4.18.2",
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
```

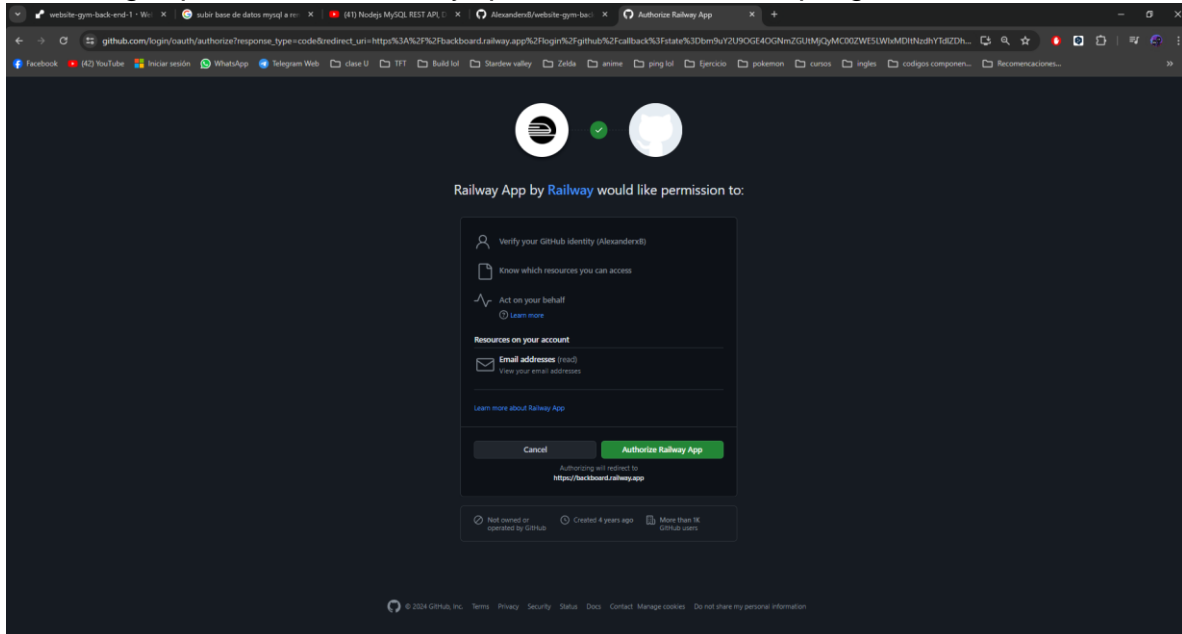
Para desplegar la base de datos se va a usar <https://railway.app/>



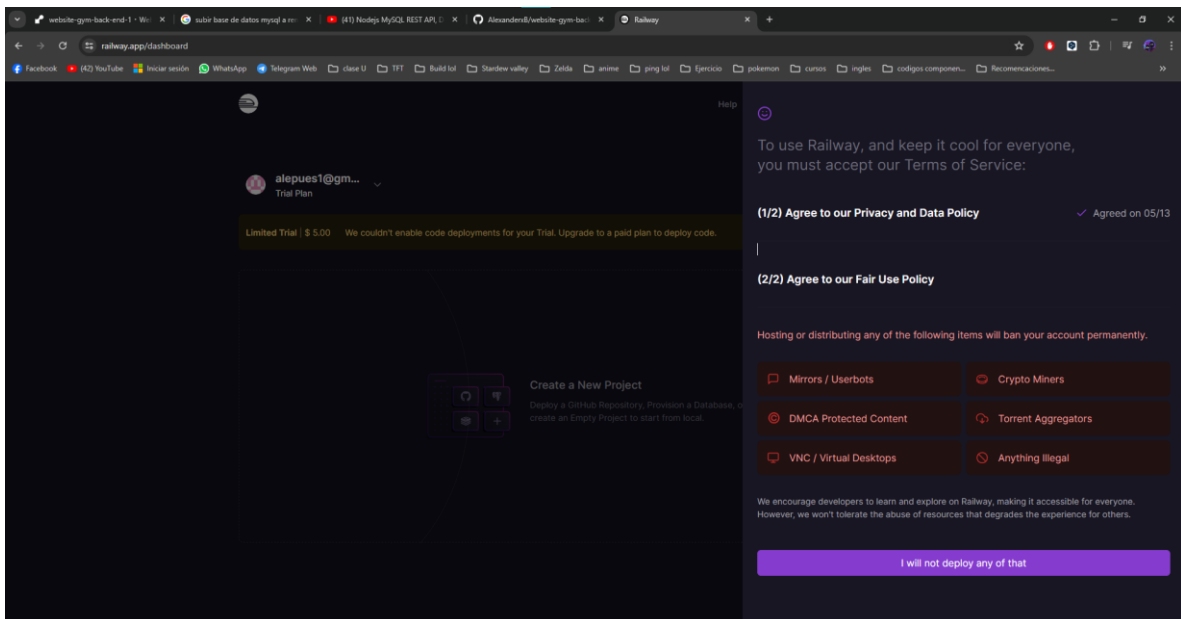
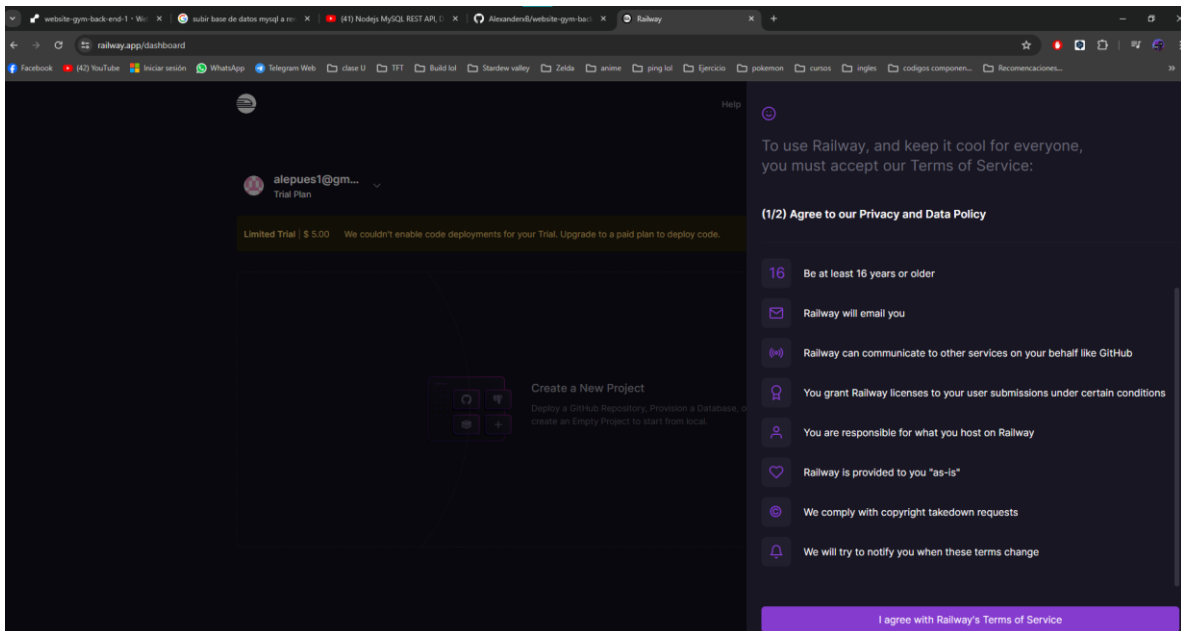
Para utilizar Railway, primero debes iniciar sesión. En este caso, lo haremos a través de GitHub.



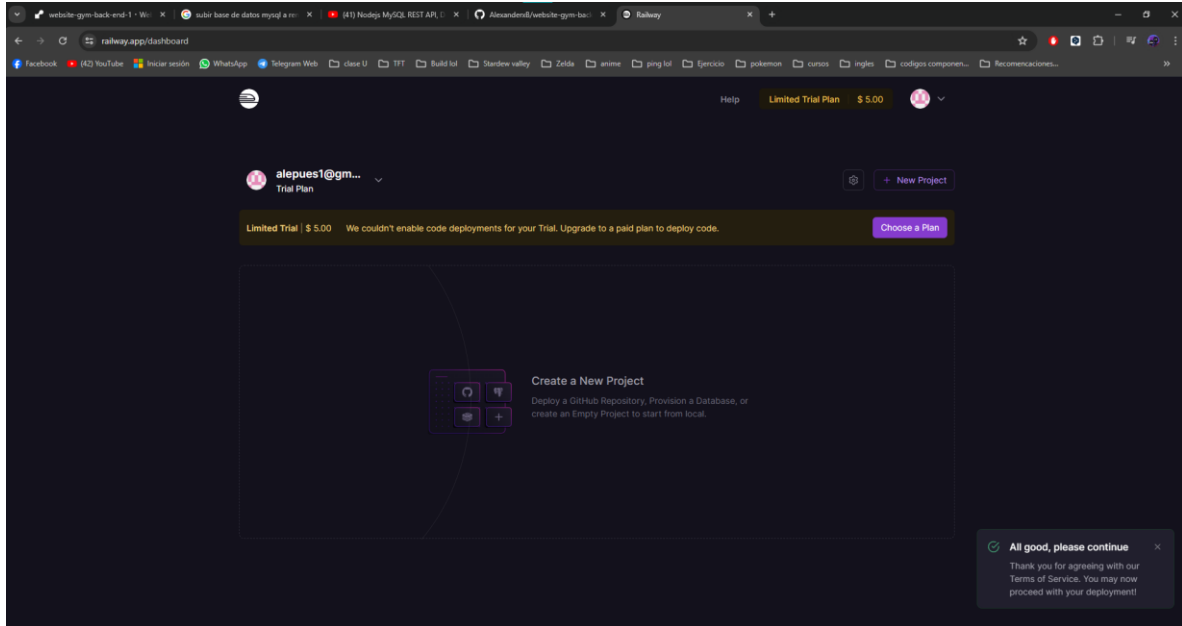
Se otorga permiso a Railway para realizar el despliegue mediante GitHub.



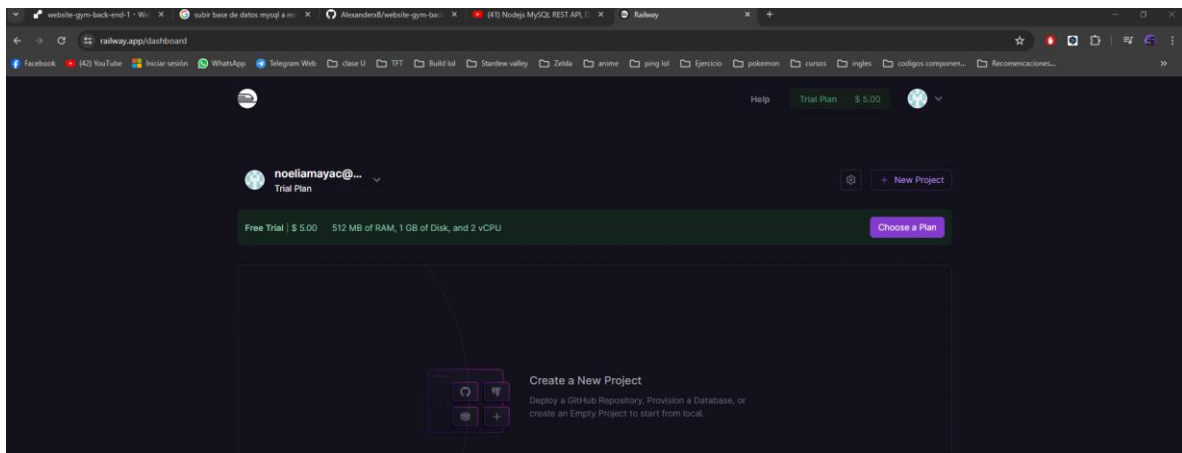
Se aceptan los términos y condiciones para poder usar railway



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

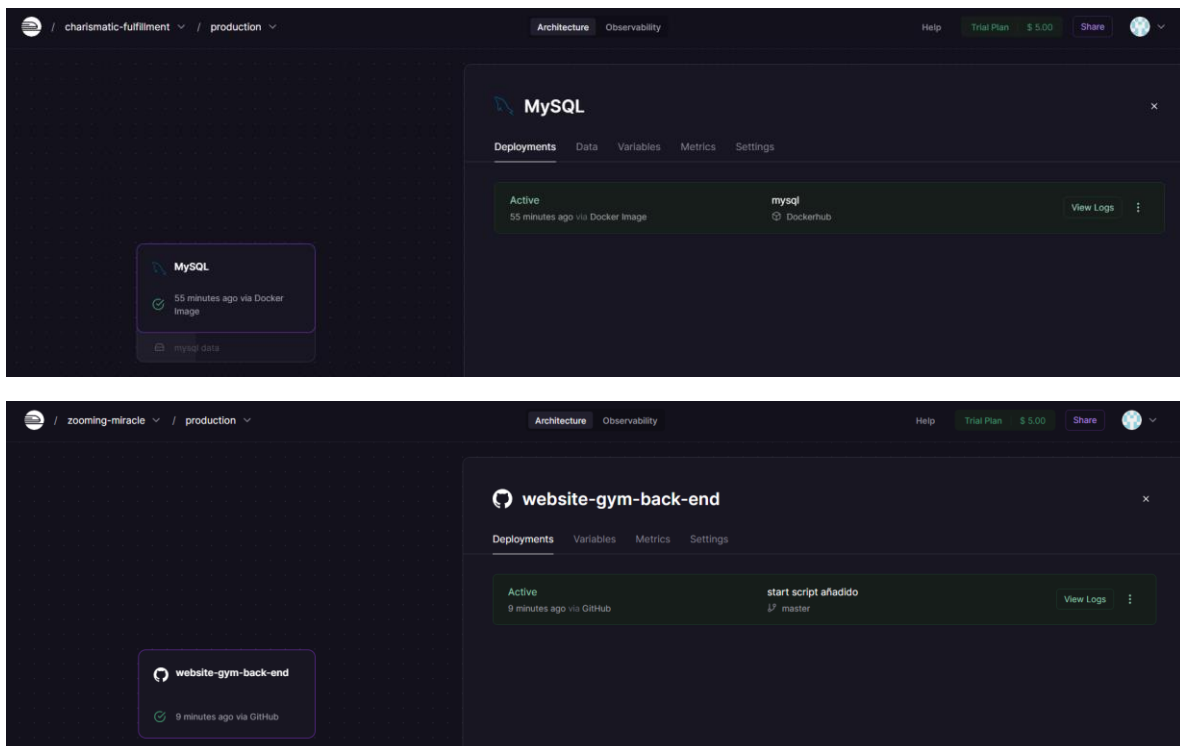


A continuación, se detallará el proceso para utilizar Railway:

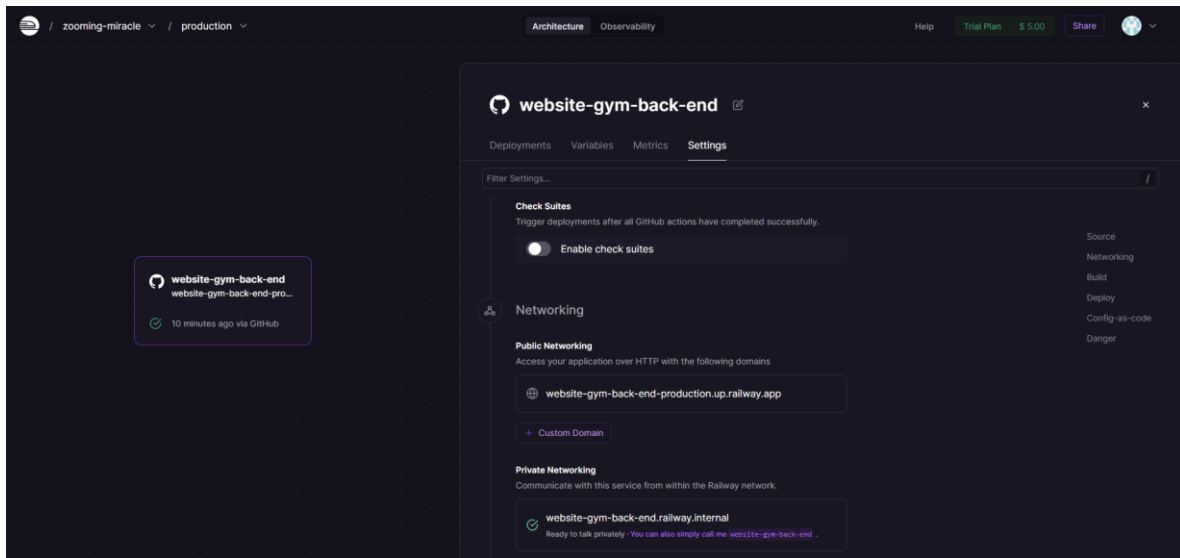
Agregar una base de datos: Una vez dentro de tu proyecto, busca la opción para agregar un nuevo servicio o recurso. Allí encontrarás la opción para añadir una base de datos. Railway es compatible con varias bases de datos, como PostgreSQL, MySQL, MongoDB, entre otras. Selecciona la base de datos que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

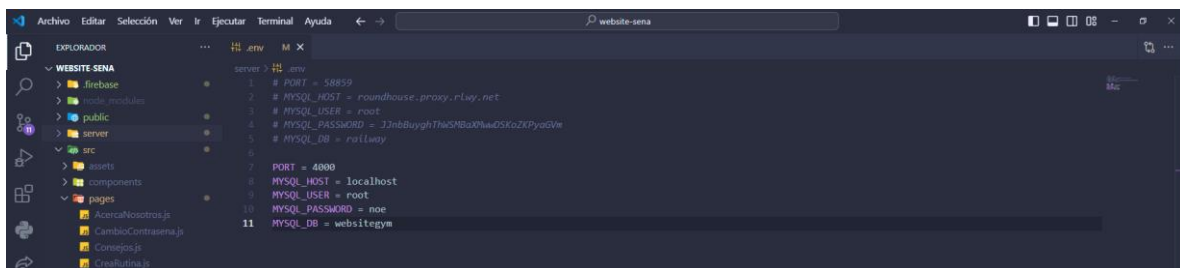
Estos pasos se ilustran en la siguiente imagen:



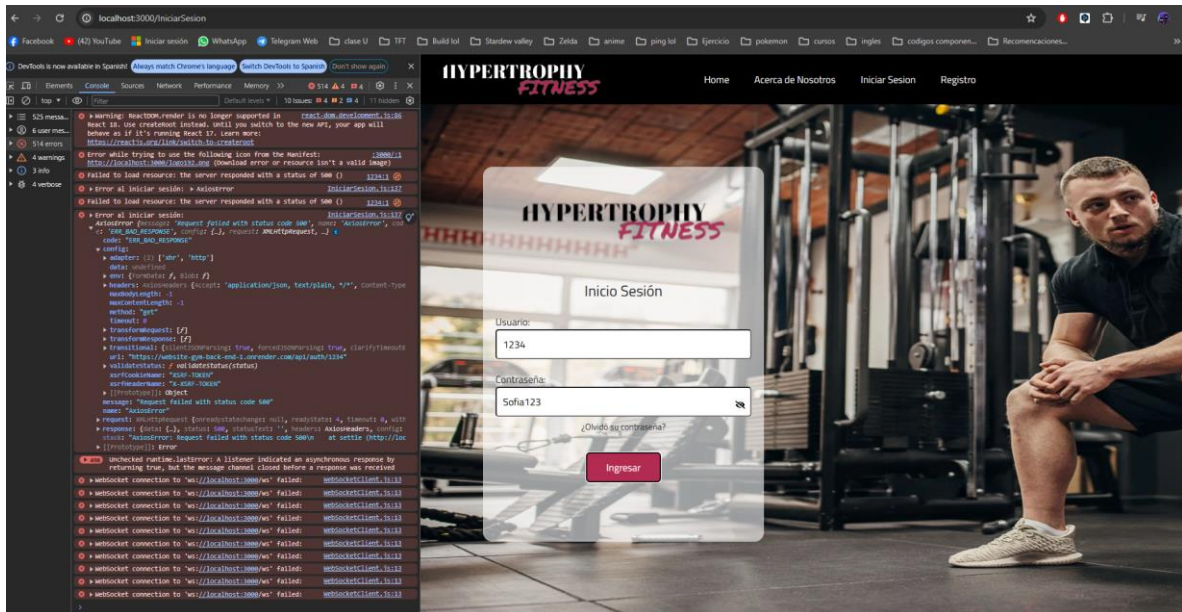
Después de seleccionar el tipo de base de datos, Railway te guiará a través de los pasos necesarios para configurarla. Esto puede incluir la selección de la región del servidor, el plan de precios (si corresponde), y la configuración de credenciales y otras opciones específicas de la base de datos.



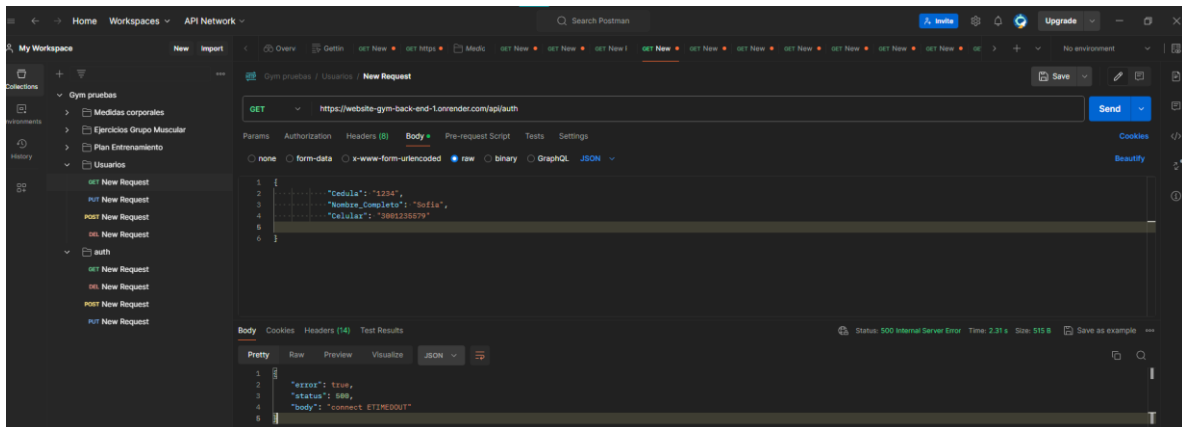
En esta imagen se evidencian las variables de entorno para hacer la conexión con la base de datos, por el momento se están usando los datos para acceder a la pagina web de forma local porque a pesar de que la base de datos tuvo un deploy exitoso, y se hicieron las configuraciones respectivas en el back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

23 hours ago via Docker Image

mysql data

MySQL

Deployments

Data

Variables

Metrics

Settings

Connect to the database MySQL

Connect

Tables

auth

ejerciciosgrup...

medidascorpor...

planetentrenami...

usuarios

Create table

MySQL

23 hours ago via Docker Image

mysql data

MySQL

Deployments

Data

Variables

Metrics

Settings

usuarios

Add Column

Cedula	Nombre_Completo	Celular
1000	Camila Sofia	3143175333
1013693067	Ana Maria Lozano Reyes	3182134459
1014	alex	3143175333
1015	Mara	31431752345
1016	brayan	3183456780
1017	alejandra	3183456780
1018	pablo	3183456780
1234	Sofia	3143175333
1235	Nani	312342345

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 1 SQL File 2 usuarios auth

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	3103222113
1013893067	Alex Maria Luciano Reyes	3102134499
1014	alex	3143173333
1015	Nara	3143173345
1016	brayan	3103456780
1017	alexandra	3103456780
1018	gabie	3103456780
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

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

CONCLUSIONES

En resumen, el proceso de desarrollo y despliegue de la página web "Hypertrophy Fitness" fue exitoso tanto a nivel local como en el entorno de producción en Internet. Utilizando tecnologías como Node.js y Express, se logró crear y hospedar la página web en un servidor local mediante localhost. Además, mediante el despliegue en un servicio de hosting con un dominio personalizado gratuito, se hizo accesible la página web para usuarios de Internet.

El despliegue local permitió realizar pruebas y ajustes en un entorno controlado antes de lanzar la página web al público. Esto garantizó su funcionamiento adecuado y su rendimiento óptimo. Por otro lado, el despliegue en Internet con un dominio personalizado proporcionó una presencia profesional en línea, permitiendo a los usuarios acceder fácilmente a la página web desde cualquier lugar.

BIBLIOGRAFÍA

Cluster: Arquitectura de computación escalable. (s/f). Hostgator.mx. Recuperado el 8 de mayo de 2024, de <https://www.hostgator.mx/blog/cluster-computacion-escalable/>

¿Qué es un cluster? (s/f). Unam.mx. Recuperado el 8 de mayo de 2024, de <https://www.revista.unam.mx/vol.4/num2/art3/cluster.htm>

¿Qué es un cluster? ¿Cuál es su función? ¿Y sus objetivos? (2021, agosto 13). Zendesk. <https://www.zendesk.com.mx/blog/cluster-que-es/>