

«Talento Tech»

Desarrollo Web 4

Clase 09





Clase N° 9 | Github y Roder

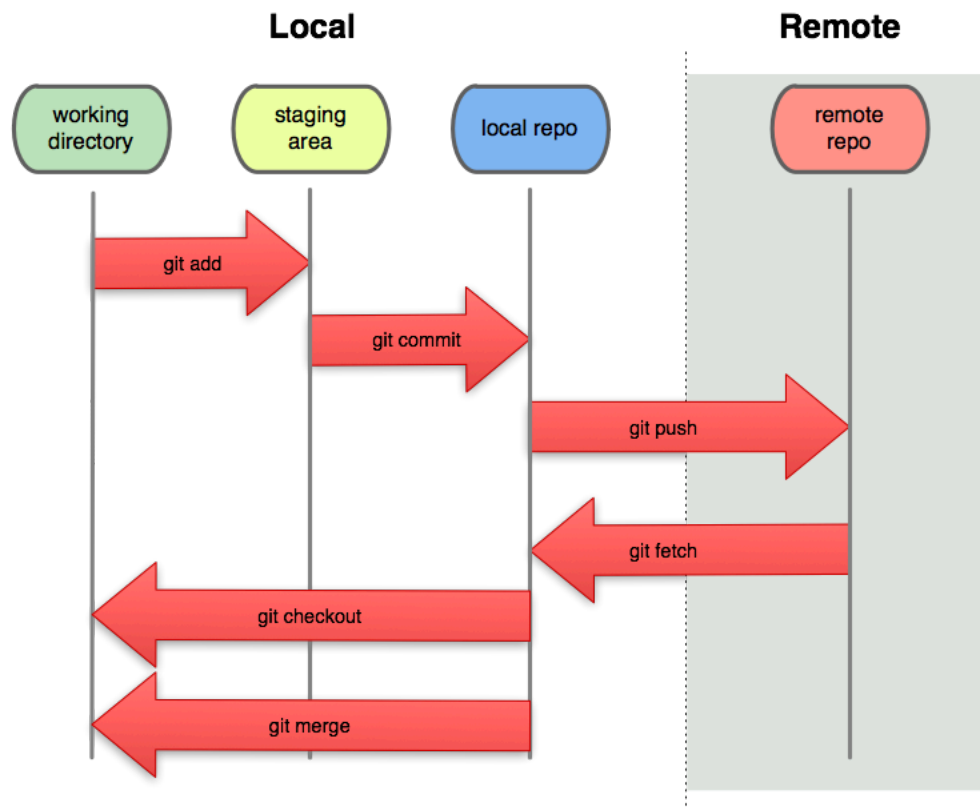
Temario:

- Definición y uso de Github
- Creación de una Cuenta en Github
- Deploy de Node.js en Render
- ¿Qué es y cómo se usa Render?



Definición y uso de Github

GitHub es una plataforma esencial para el desarrollo colaborativo de software, proporcionando herramientas poderosas para el control de versiones, seguimiento de problemas y facilitando la colaboración entre desarrolladores.



¿Cual es el Propósito de Github?

GitHub permite a los desarrolladores colaborar en proyectos de software. Facilita el seguimiento de cambios, la gestión de problemas, y proporciona herramientas para trabajar en equipo.

¿Qué beneficios me trae el uso de Github?

Control de versiones: Utiliza **Git** para rastrear y gestionar cambios en el código fuente, permitiendo la colaboración simultánea de múltiples desarrolladores.



Repositorios: Los proyectos se almacenan en **repositorios**. Cada **repositorio** contiene todo el historial de cambios, archivos y carpetas relacionadas con un proyecto.

Branches (ramas): Los desarrolladores pueden trabajar en ramas separadas del código para implementar nuevas características o solucionar problemas sin afectar la rama principal.

Pull Requests: Son propuestas de cambios que permiten revisar y discutir el código antes de fusionarlo con la rama principal. Facilitan el proceso de revisión de código.

Issues: Se utilizan para realizar seguimiento de tareas, errores o mejoras en un proyecto. Pueden etiquetarse, asignarse y comentarse para una mejor organización.

Forks: Permite a los desarrolladores copiar un repositorio existente para realizar cambios sin afectar al repositorio original. Posteriormente, pueden proponer la fusión de esos cambios mediante un Pull Request.

Colaboración: GitHub facilita la colaboración entre desarrolladores, ya sea en proyectos de código abierto o en equipos privados.

GitHub Actions: Ofrece un sistema de automatización para realizar tareas como pruebas, construcciones y despliegues directamente desde el repositorio.

Integración Continua (CI) y Despliegue Continuo (CD): GitHub facilita la integración con servicios de CI/CD para automatizar pruebas y despliegues cuando se realizan cambios en el código.

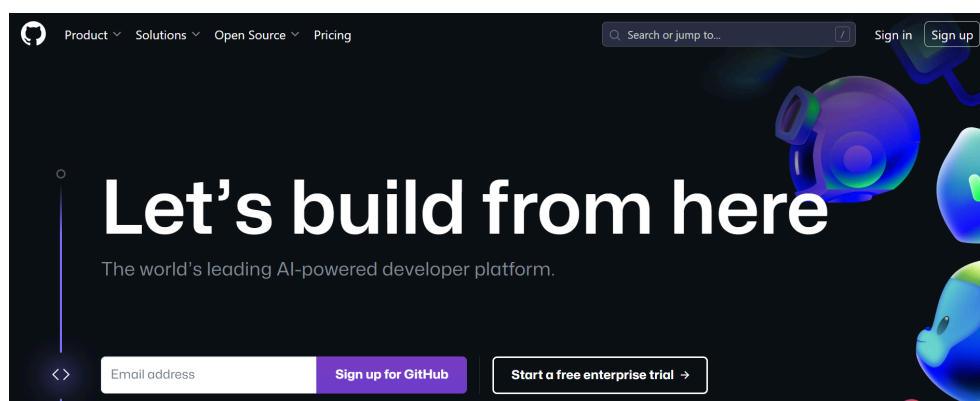
Creación de una Cuenta en Github

Ahora nos crearemos una cuenta 😊

Accede al Sitio Web de GitHub: <https://github.com/>



1. Inicia el Proceso de Registro:
Haz clic en "Sign up" (Registrarse):



Completa el Formulario:

Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ [redacted]

Create a password*
✓ [redacted]

Enter a username*
✓ [redacted]

Email preferences
☐ Receive occasional product updates and announcements.

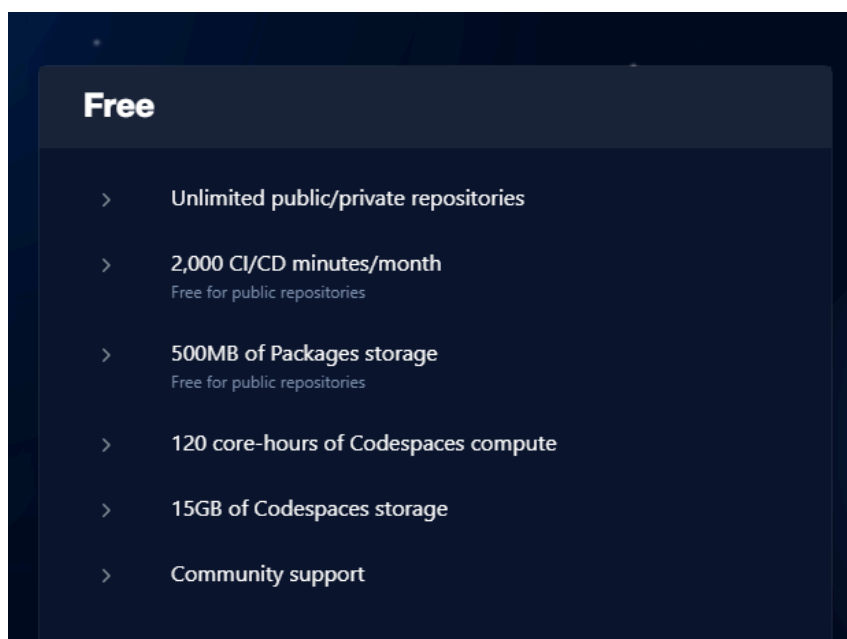
Continue



Completa el Captcha:

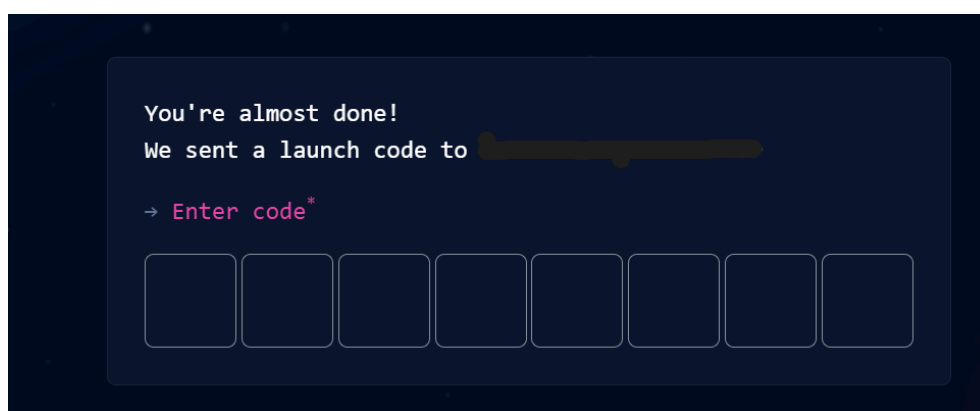


Selecciona el Plan Gratuito



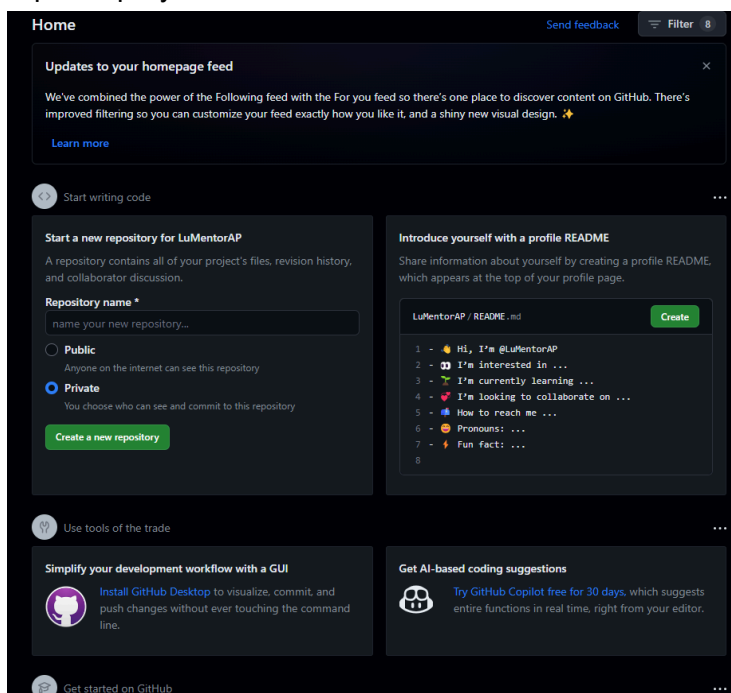


Verifica Tu Cuenta



Abre el correo electrónico que recibas y sigue las instrucciones para verificar tu cuenta.

¡Listo! Ahora tienes una cuenta en GitHub. Personaliza tu perfil y comienza a explorar proyectos.





Más adelante, en esta clase, haremos uso de esta cuenta junto con Render 😊

Deploy de Node.js en Render



Cuando realizamos el despliegue de una aplicación en **Node.js** nos referimos al proceso de hacer que esta aplicación sea accesible y funcional en un entorno de producción. Implica tomar el código fuente de la aplicación **Node.js**, configurar el entorno de producción y asegurarse de que la aplicación esté lista para manejar tráfico real.

A continuación, te dejamos unos consejos generales que realizamos a la hora de hacer un deploy con **Node.js**:

Prepara tu Código: Asegúrate de que tu código esté listo para producción. Esto puede incluir la optimización del código, eliminación de **dependencias** no utilizadas y configuración de **variables de entorno** específicas para el entorno de producción.

Gestiona tus Dependencias: Asegúrate de que todas las dependencias necesarias estén instaladas. Puedes utilizar el archivo package.json y ejecutar `npm install --production` para instalar solo las dependencias necesarias para la ejecución en producción.



Configura el Servidor: Configura el servidor que se utilizará para ejecutar la aplicación. Esto puede incluir la instalación y configuración de un servidor web como **Nginx** o **Apache** que actúe como un proxy inverso para tu aplicación Node.js.

Entorno de Producción: Ajusta la configuración de la aplicación para el entorno de producción. Esto incluye configuraciones como puertos, direcciones de base de datos, ajustes de seguridad y otros parámetros específicos para el entorno de producción.

Gestión de Procesos: Implementa una solución para gestionar el proceso de tu aplicación en producción. Herramientas como **PM2** o **systemd** son comunes para garantizar que la aplicación se reinicie automáticamente en caso de fallos y que se ejecute de manera continua.

Configuración de Variables de Entorno: Utiliza variables de entorno para almacenar información **sensible**, como **claves** de API o **credenciales de bases de datos**, de manera segura en el entorno de producción.

Despliegue Continuo (Opcional): Considera implementar un sistema de despliegue continuo para facilitar actualizaciones automáticas y rápidas de tu aplicación en producción.

Monitoreo y Registro: Implementa herramientas de monitoreo y registro para realizar un seguimiento del rendimiento de tu aplicación en producción y diagnosticar problemas si es necesario.

El despliegue de una aplicación **Node.js** puede variar según la **infraestructura** y las **necesidades** específicas del proyecto, pero estos son pasos generales que suelen aplicarse en muchos casos.

¿Qué es y cómo se usa Render?

Render es como una plataforma súper útil para los equipos que crean programas. Les ayuda a lanzar sus creaciones de manera fácil y rápida, sin importar si son pequeños experimentos o aplicaciones gigantes con un montón de cosas funcionando. Lo más importante de todo esto es que se asegura de que todo funcione bien todo el tiempo.



En comparación con otros lugares donde se pueden guardar y hacer funcionar programas, como **Heroku o Railway**, **Render** tiene algunas ventajas. Por ejemplo, se enfocan mucho en que lo que guardas en su plataforma siempre esté disponible y funcionando. Aunque otros sitios también ofrecen servicios similares, Render se destaca por ser robusto y eficiente.

Imagina que **Render** es como tu mejor amigo asegurándose de que tus programas estén siempre listos para ser usados, sin importar lo grande o pequeño que sea.

Vamos a usar Render junto a Git con nuestra cuenta ya creada 🚀

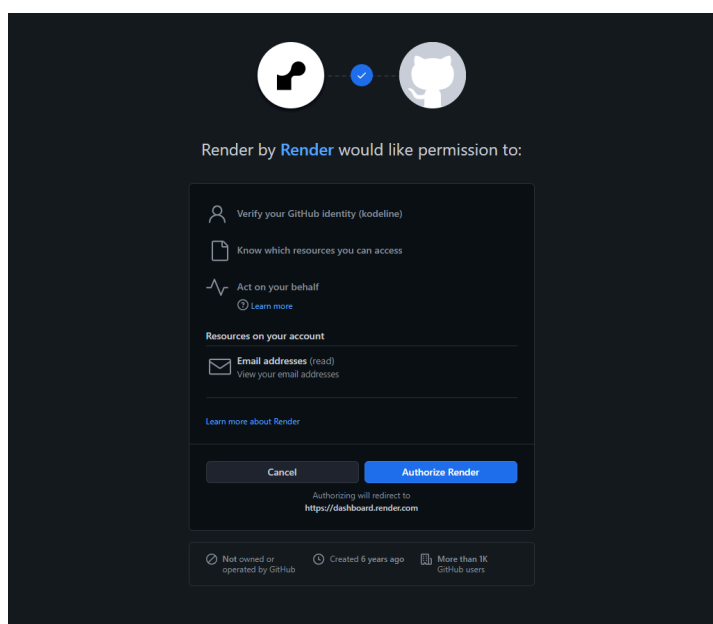
Paso 1

Crearse una cuenta en render.com, se recomienda crearse la cuenta con tu cuenta de Github así evitas más configuraciones.



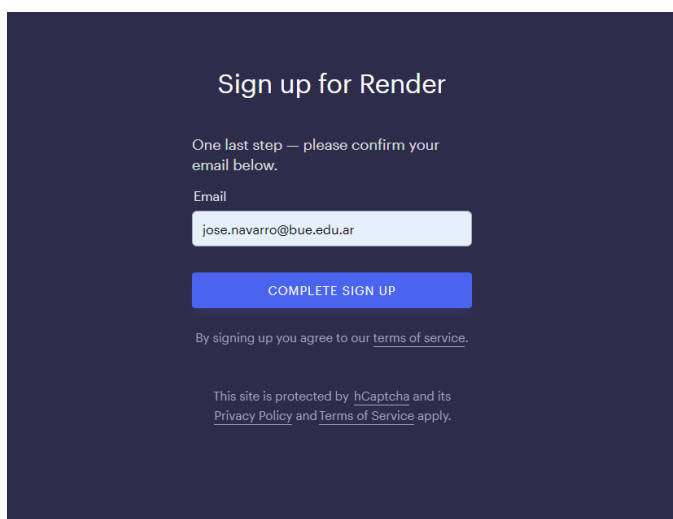
Paso 2

Autorizar a Render para que pueda tener acceso a su cuenta de Github.



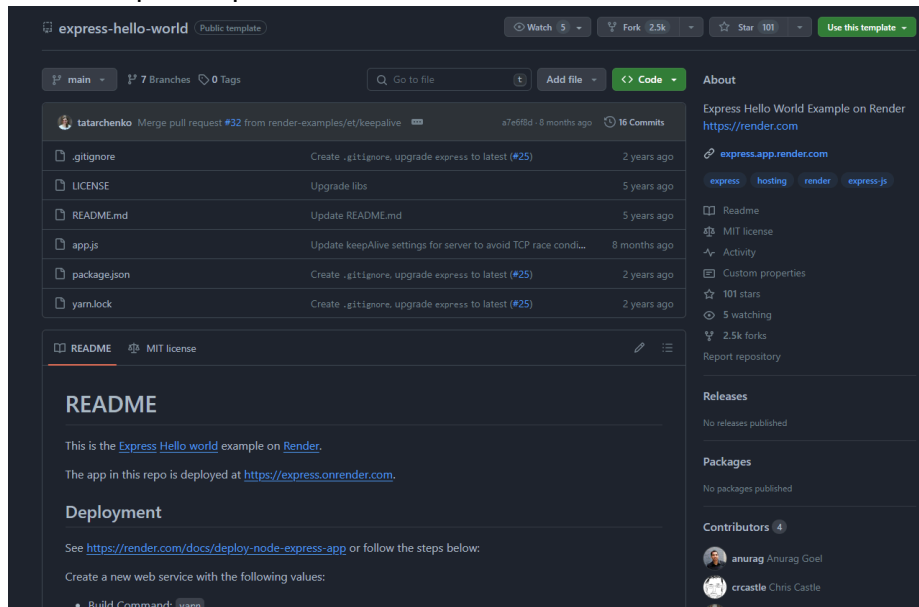
Paso 3

Confirmá tu cuenta de Render, verificando la casilla de mensajes de tu correo electrónico.



Paso 4

Vamos a 'Forkear' un repositorio que nos da Render para usar de prueba entrando a este link: <https://github.com/render-examples/express-hello-world> y clickeamos en Fork continuando con los pasos que ofrece Git.



The screenshot shows the GitHub repository page for 'express-hello-world' by 'tatarchenko'. The repository is a public template with 5 watchers, 2.5k forks, and 101 stars. It includes a 'Use this template' button. The repository contains files like .gitignore, LICENSE, README.md, app.js, package.json, and yarn.lock. The README section is visible, stating that this is the Express Hello world example on Render, and the app is deployed at <https://express.onrender.com>. The deployment instructions are also shown.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *


express-hello-world is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ Copy the `main` branch only

Contribute back to render-examples/express-hello-world by adding your own branch. [Learn more.](#)

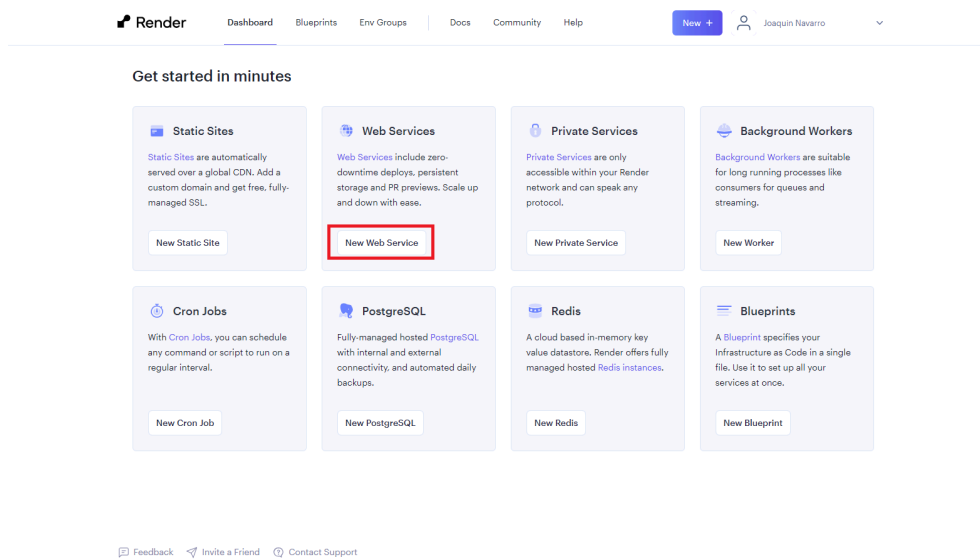
 You are creating a fork in your personal account.

[Create fork](#)



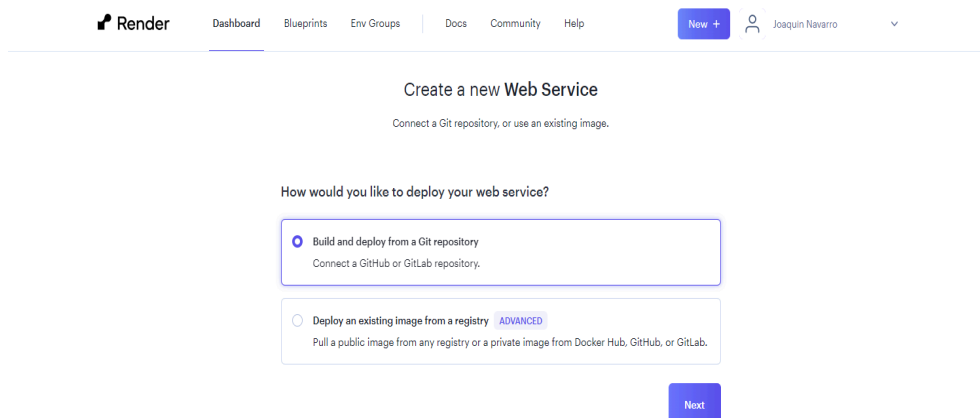
Paso 5

Después, devuelta en Render, podremos ver el dashboard principal y todos los servicios que nos brinda render, nosotros usaremos Web Services



Paso 6

Utilizamos la opción predeterminada **Build and deploy from a Git repository**, en este caso vamos a hacer el deploy de nuestra app de Node.js desde su repositorio en Github.

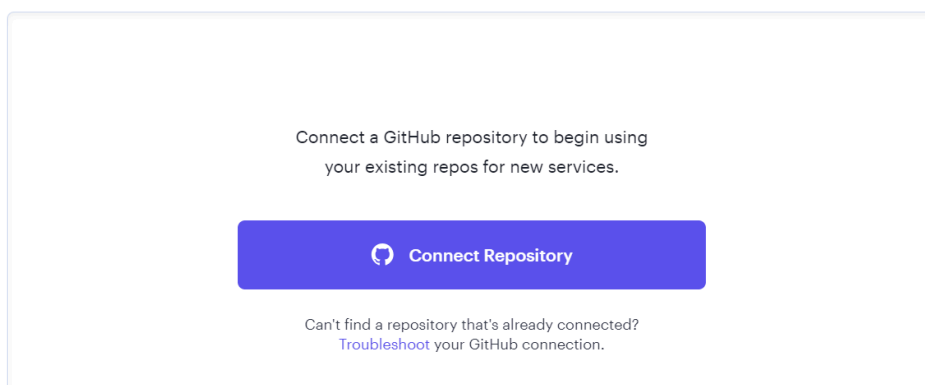




Paso 7

Conectamos con nuestro repositorio de GIT.

Connect a repository



Paso 8

Podemos elegir qué Render vea todos nuestros repositorios o solo uno. Este paso es libre.



☒ **All repositories**

This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

☐ **Only select repositories**

Select at least one repository.
Also includes public repositories (read-only).

with these permissions:

- ✓ **Read** access to Dependabot alerts, administration, code, and metadata
- ✓ **Read and write** access to actions, checks, commit statuses, deployments, environments, issues, pull requests, repository hooks, and workflows

User permissions

Render can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

- ✓ **Read** access to email addresses

Install

Cancel



Next: you'll be directed to the GitHub App's site to complete setup.



Paso 9

Elegimos el repositorio que forkeamos previamente.

Connect a repository

  / express-hello-world • 16 days ago

Connect

Paso 10

Usamos los siguientes valores siendo estos los únicos que nos importan por el momento para esta prueba

Runtime	Node
Build Command	yarn
Start Command	node app.js

Paso 11

Elegimos la opción FREE y clickeamos “Create Web Server”

Environment Variables Optional
Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

NAME_OF_VARIABLE

value

Generate

🗑

+ Add Environment Variable

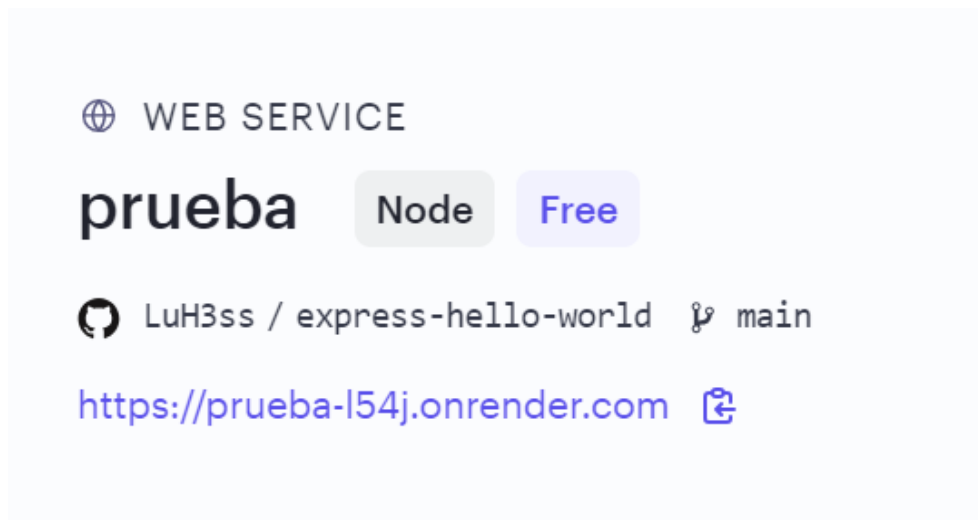
Advanced ▾

Create Web Service



¡Último paso!

Esperamos a que termine el deploy y cliqueamos en el link que aparece en pantalla.



Listo. 😊


Hello from Render!

Desafío #4:

1. Crear una cuenta en Render:

- Registrarse en Render mediante su página web y seguir el proceso de creación de una cuenta.

2. Preparar el proyecto para el despliegue:

- Asegurarse de que el proyecto esté correctamente configurado y funciona localmente.
- Verificar que todas las dependencias estén listadas en el archivo `package.json` y que se incluyan en el repositorio.

3. Configurar el despliegue en Render:

- Acceder al panel de Render y crear un nuevo servicio.
- Seleccionar las opciones adecuadas para el tipo de servicio que se está desplegando (por ejemplo, un servicio de Node.js para una aplicación Express).
- Configurar las variables de entorno necesarias, como las credenciales de la base de datos y cualquier otra configuración específica del entorno de producción.

4. Desplegar el proyecto en Render:

- Conectar el servicio de Render con el repositorio de GitHub o GitLab donde se encuentra el proyecto.
- Configurar la rama y la configuración de despliegue automático si es necesario.
- Iniciar el despliegue del proyecto en Render y verificar que se complete correctamente.

5. Verificar la API REST desplegada:

- Una vez que el despliegue se haya completado, acceder a la URL proporcionada por Render para verificar que la API REST esté accesible y funcionando correctamente en el entorno de producción.

6. Bonus Extra (COMPLEMENTARIO):

- Realizar pruebas adicionales en la API REST desplegada para asegurarse de que todos los endpoints funcionen como se esperaba.
- Documentar cualquier problema encontrado durante el proceso de despliegue y cómo se resolvieron.
- Tomar capturas de pantalla o grabar un video del proceso de despliegue y compartirlo como evidencia adicional.



Recuerda seguir cuidadosamente cada paso y verificar que la API REST esté funcionando correctamente en el entorno de Render una vez que se haya completado el despliegue.





Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad