

ALKOSTO TECH UTP
MANUAL DE INSTALACIÓN

Isabela Aguirre Echeverri

Kevin Alejandro Arango Chaparro

Mariana Tellez Gutierrez

Santiago Torifa Manso



UNIVERSIDAD TECNOLÓGICA DE PEREIRA
Facultad de Ingenierías – IS714 Ingeniería De Software I Gr. 102

Pereira, 2025

Índice

<u>Introducción</u>	<u>3</u>
<u>Arquitectura del sistema</u>	<u>4</u>
<u>Requisitos del sistema</u>	<u>5</u>
<u>Instalación de la base de datos (PostgreSQL Neon)</u>	<u>6</u>
<u>Instalación del backend (NestJS Koyeb)</u>	<u>10</u>
<u>Instalación del frontend (Angular Netlify)</u>	<u>21</u>

Introducción

El presente documento describe el procedimiento para instalar y configurar el sistema web Alkosto UTP, desarrollado por el equipo HamKov Engineers. El sistema está compuesto por un frontend en Angular, un backend en NestJS y una base de datos PostgreSQL. Este manual cubre la instalación del sistema en servidores gratuitos, utilizando Koyeb para el backend, Netlify para el frontend y Neón para la base de datos. También se tienen en cuenta detalles como variables de entorno necesarias para el funcionamiento y como y donde se obtienen.

Tener en cuenta seguir el instructivo al pie de la letra para hacer una instalación exitosa del aplicativo Alkosto HamKov Engineer o mas bien conocido como Alkosto UTP.

Arquitectura del sistema

El sistema sigue una arquitectura cliente-servidor con los siguientes componentes:

- Frontend (Angular) desplegado en Netlify.
- Backend (NestJS) desplegado en Koyeb.
- Base de datos (PostgreSQL) alojada en Neon.

El flujo de datos es el siguiente:

El cliente (Angular) envía solicitudes HTTP al backend (NestJS), que procesa la lógica de negocio y accede a la base de datos Neon a través de una conexión PostgreSQL segura y auto-gestionada.

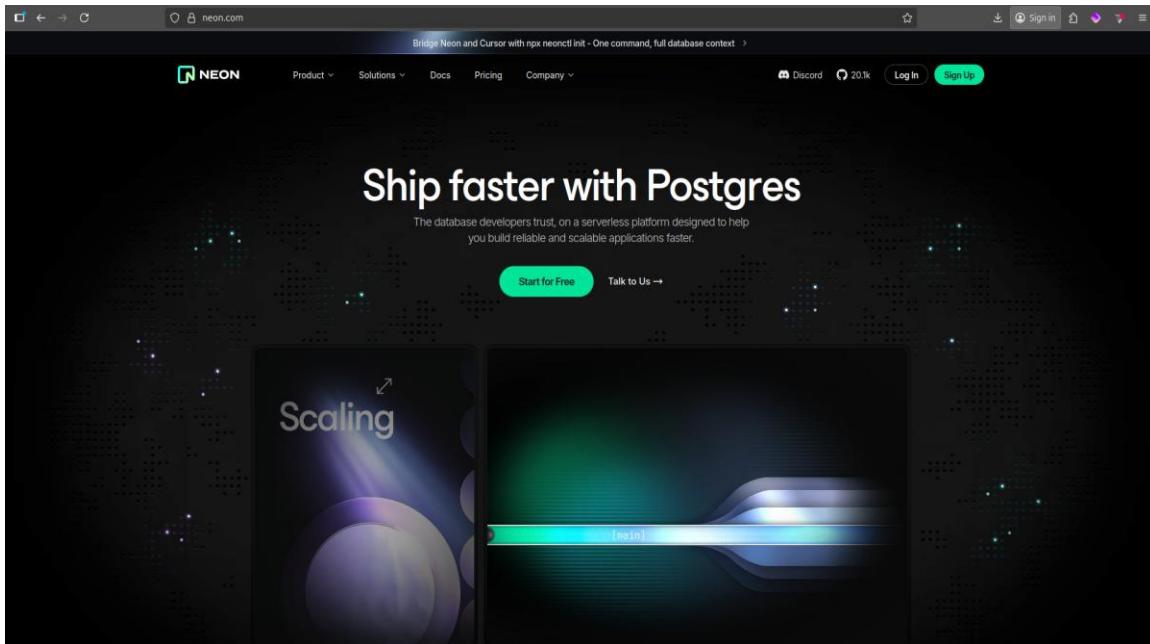
Requisitos del sistema

En este apartado se tienen los requisitos del sistema o máquina desde la que se desee realizar la instalación del aplicativo, usando los sistemas serverless auto-gestionados de Netlify, Koyeb y Neon.

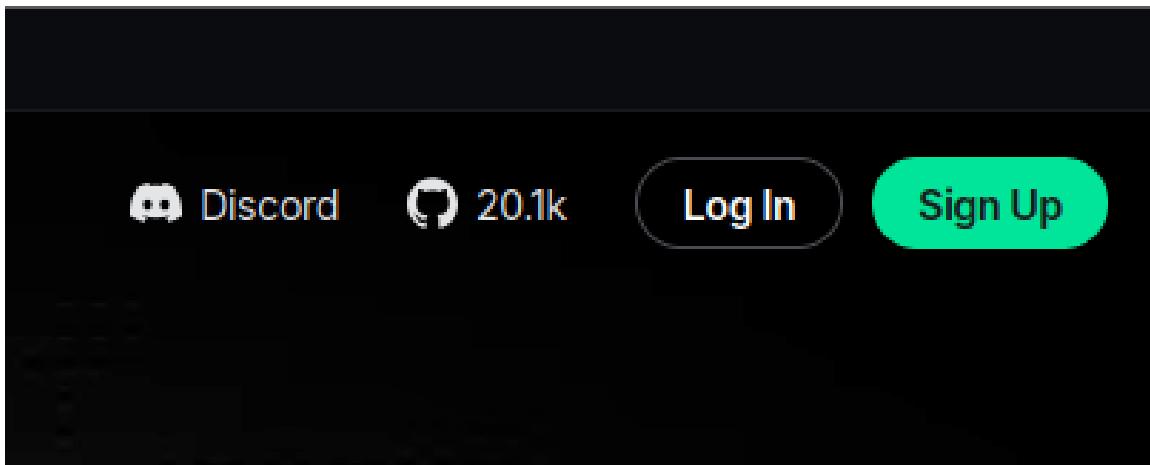
Sistema operativo	Linux, Windows o MacOS
Entorno de ejecución	Node.js 20 o superior
Gestor de paquetes	Npm versión 10 o superior
Sistema de control de versiones	Git instalado en la máquina local
Cuentas registradas en	GitHub, Netlify, Koyeb y Neon
Red	Conexión a internet estable

Instalación de la Base de Datos (PostgreSQL en Neon)

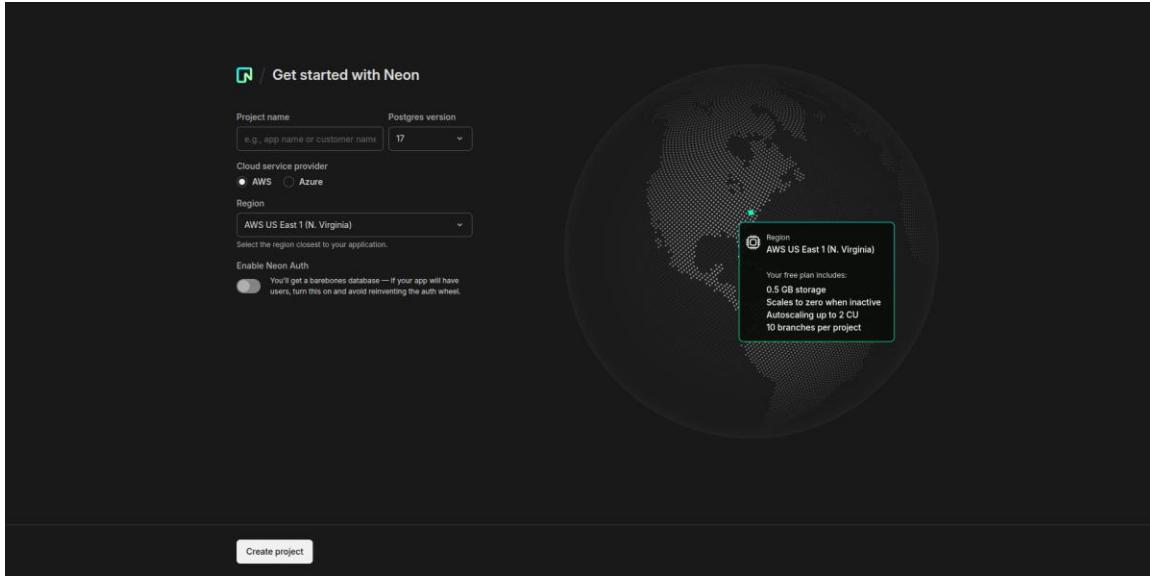
1. Lo primero que debemos hacer es ingresar al sitio web oficial de [Neon](#)



2. Una vez en la página principal, damos clic en el botón “**Log In**”, ubicado en la parte superior derecha del sitio web.

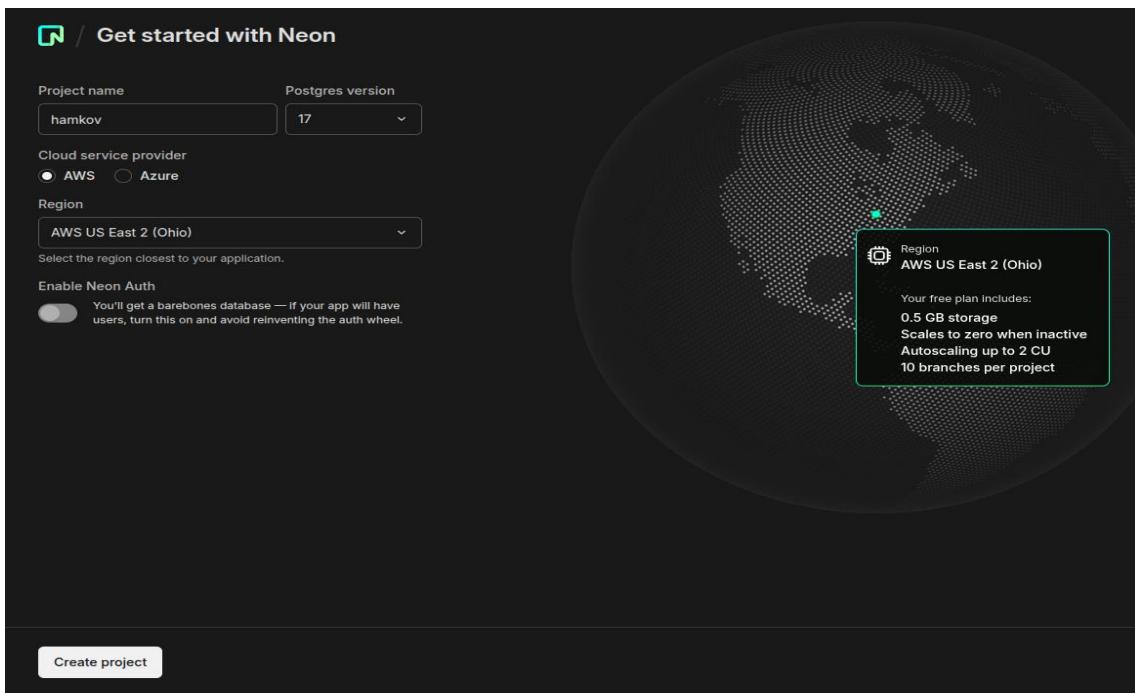


3. Al hacer clic en “**Log In**”, se desplegarán las diferentes opciones de inicio de sesión disponibles. Podremos ingresar utilizando nuestra cuenta de preferencia (por ejemplo, Google o GitHub). Una vez dentro de la plataforma, estaremos listos para comenzar con la creación de nuestra nueva base de datos.



4. A continuación, se mostrará la página principal para crear nuestro **primer proyecto de base de datos** en Neon. El primer paso será asignar un **nombre** al proyecto (puede ser cualquiera, por ejemplo, *hamkov*). Luego, seleccionamos la **versión de PostgreSQL** requerida en este caso, la versión **17**

Posteriormente, debemos escoger el **proveedor de nube** entre las opciones disponibles (**AWS** o **Azure**), según preferencia. Finalmente, seleccionamos la **región geográfica** más adecuada, idealmente aquella donde se concentre la mayor cantidad de usuarios que vayan a utilizar la aplicación *Alkosto UTP*, ya que esto influye directamente en la **velocidad y rendimiento** del sistema.



- Una vez configurado el proyecto, damos clic en el botón “Create project”. Si todos los pasos anteriores se realizaron correctamente, Neon procederá a crear y desplegar automáticamente la base de datos.

Tras unos segundos, veremos una pantalla de confirmación similar a la siguiente, donde se muestra la información general del proyecto recién creado, incluyendo el nombre de la base de datos, el proveedor de nube seleccionado y la región elegida.

The screenshot shows the Neon Project dashboard for a project named 'hamkov'. The left sidebar includes sections for Dashboard, Branches, Integrations, Auth, Settings, Overview, Data API, Monitoring, SQL Editor, Tables, Backup & Restore, and Data Masking (marked as Beta). The main area displays a 'Project dashboard' with sections for connecting to the database, importing data, connecting from an IDE, and integrating with AI tools. It also shows resource usage metrics: Branches (2/10), Compute (0 / 100 CU-hrs), Storage (0 / 0.5 GB), and Network transfer (0 / 5 GB). Below these metrics, there's a 'Monitoring' section and a 'Branches' section listing two branches: 'production' (Default, Primary compute .25 => 2 CU, Active) and 'development' (Primary compute .25 => 1 CU, Pending). A note at the bottom of the branches section says, "You don't have any preview branches yet. Improve your workflow by adding database branching to your development previews." There is also a button to "Install an integration".

Con esto confirmamos la **creación y despliegue exitoso de nuestra base de datos en la nube**, a través del servicio proporcionado por **Neon**.

Instalación del Backend (NestJS en Koyeb)

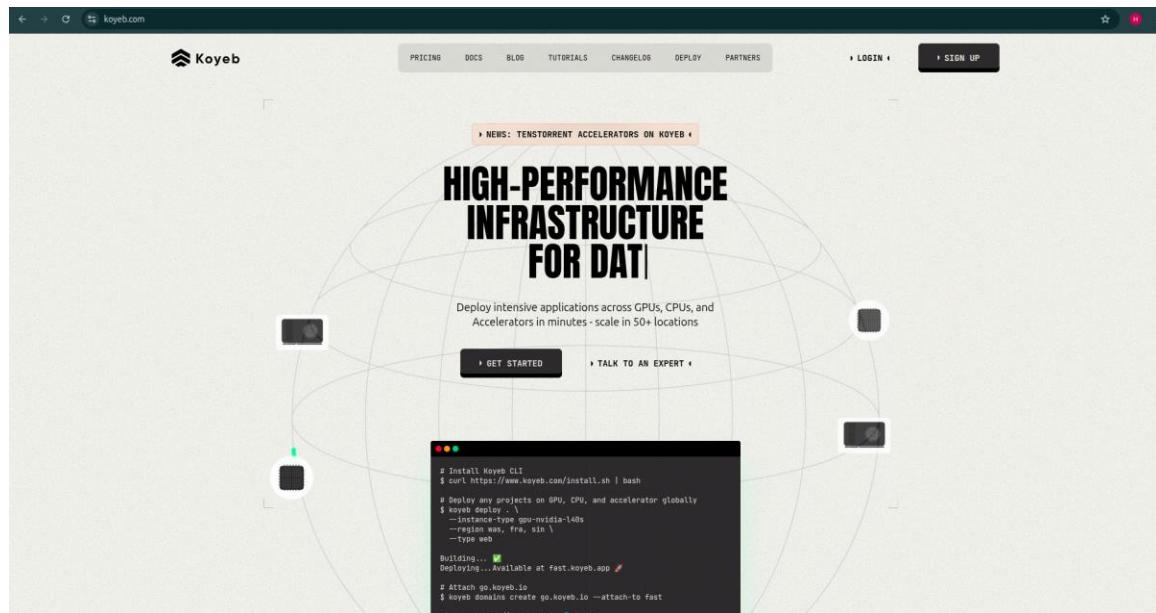
Objetivo: publicar el backend NestJS del proyecto Alkosto_UTP (carpeta [/api](#)) en Koyeb utilizando el plan gratuito, conectando el repositorio de GitHub y configurando variables de entorno para integrarlo con Neon (PostgreSQL) y los servicios de correo/Twilio.

1) Prerrequisitos

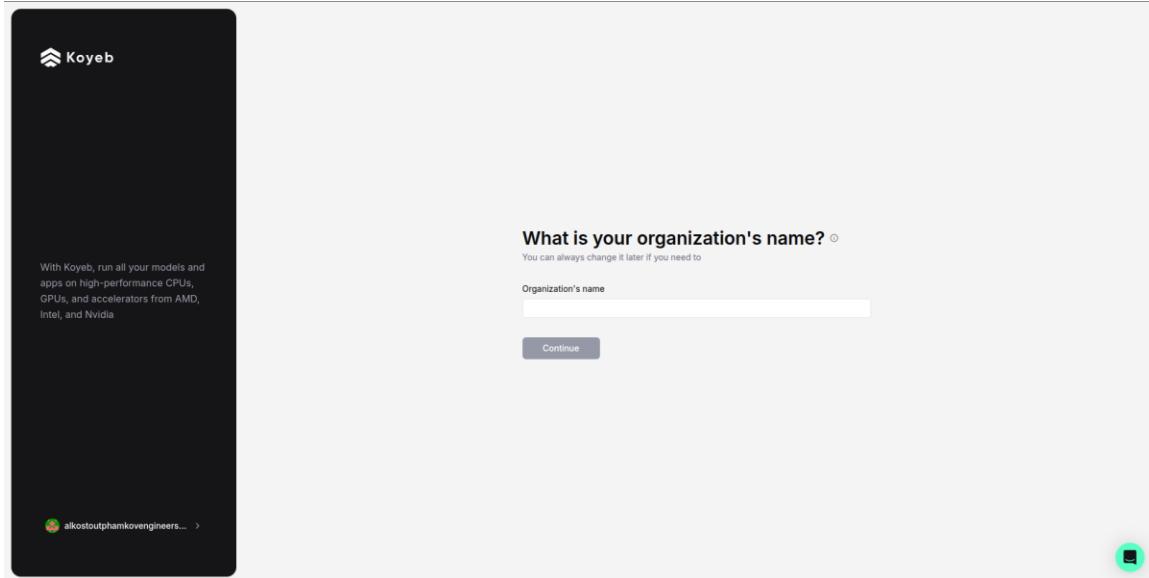
1. Contar con una cuenta en **GitHub** con acceso al repositorio [Krosbift/Alkosto_HamKov-Engineers](#).
2. Haber creado la base de datos en **Neon**.

2) Pasos para la instalación.

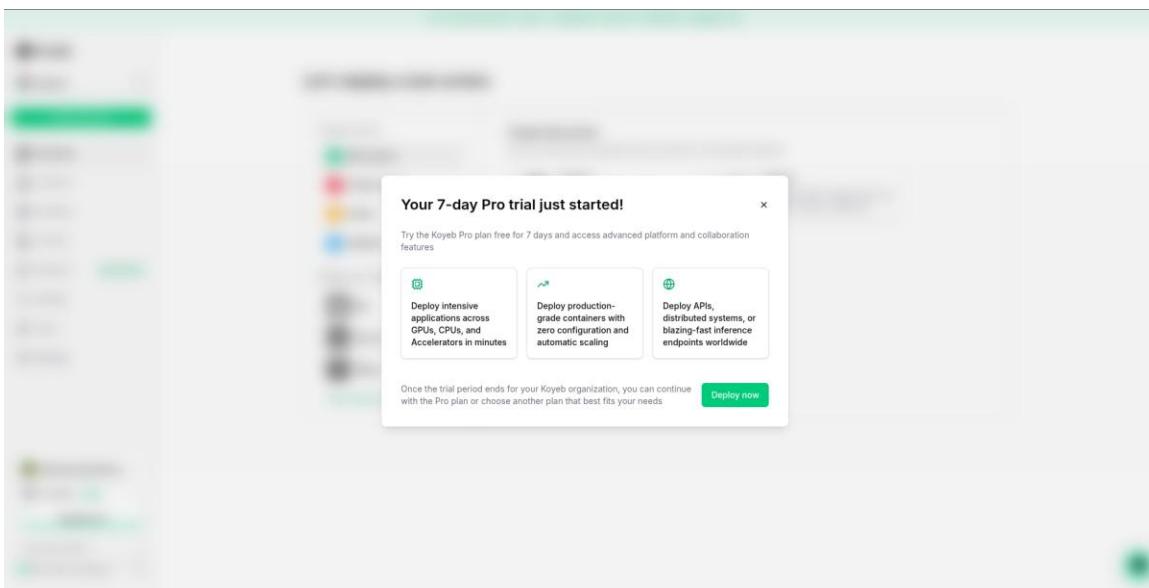
1. Lo primero que debemos hacer es ingresar al sitio web de [Koyeb](#)



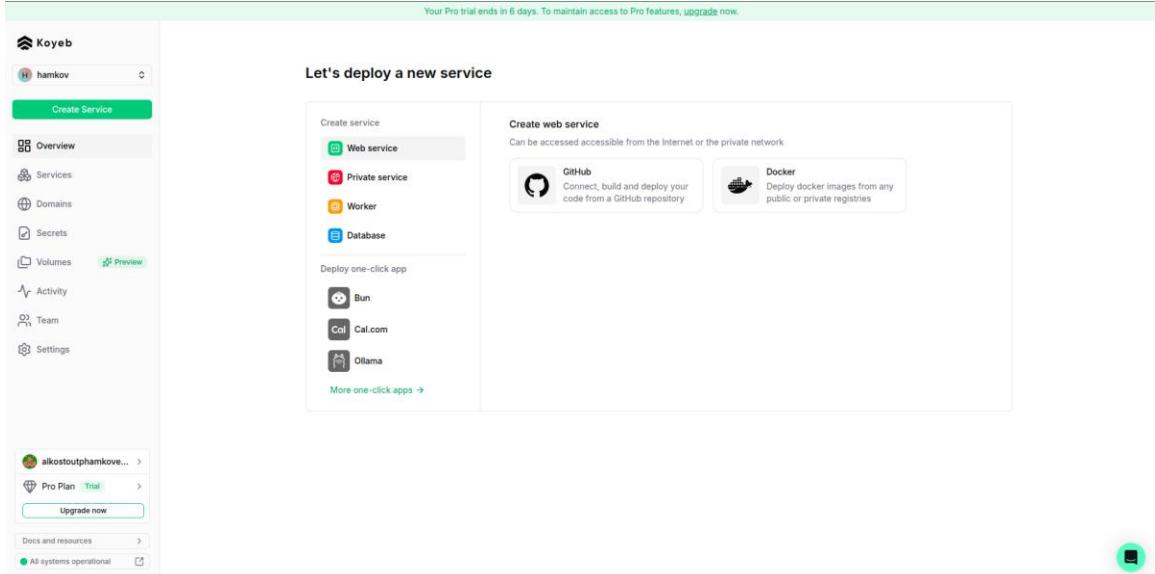
2. Una vez dentro, nos registramos y veremos la siguiente pantalla.



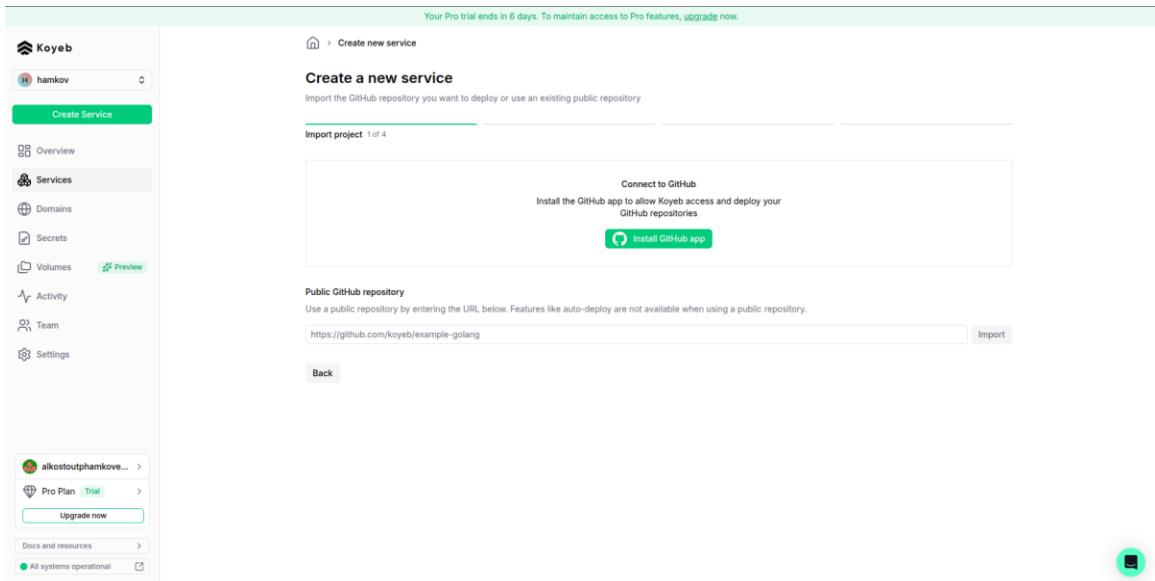
3. En este apartado se nos solicita el **nombre de nuestra organización**, el cual debemos ingresar antes de dar clic en el botón “**Continue**”. Al hacer clic en “**Continue**”, seremos dirigidos a otra sección donde Koyeb nos hace algunas preguntas básicas sobre nuestros intereses y el uso que queremos darle a la plataforma. Sin embargo, podemos simplemente hacer clic en el botón “**Skip**” para omitir esta parte y continuar hacia el panel principal. En este punto, Koyeb nos informará que tenemos **7 días de la versión PRO gratuita**, aunque para este proyecto no será necesario utilizarla.



4. Una vez en el **sitio principal**, comenzamos el proceso de despliegue. El primer paso es indicar que queremos **crear un nuevo servicio web**, y que lo haremos utilizando **GitHub** como fuente del código.



5. Al hacer clic en la opción de **GitHub**, se iniciará el asistente paso a paso para crear el servicio web mediante este método. En el campo que dice “**Public GitHub repository**”, debemos agregar la **URL pública del repositorio de GitHub** donde se encuentra el código del backend del aplicativo **Alkosto UTP**, y posteriormente damos clic en el botón “**Import**”.

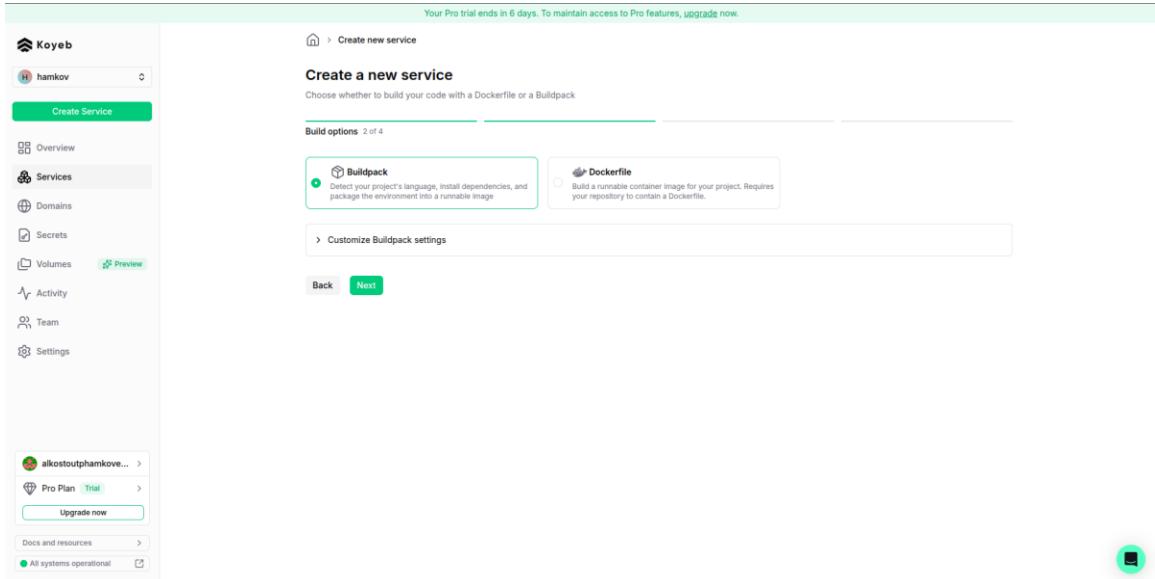


6. Después de esto, aparecerá una nueva sección que nos ofrecerá dos opciones para crear el servicio:

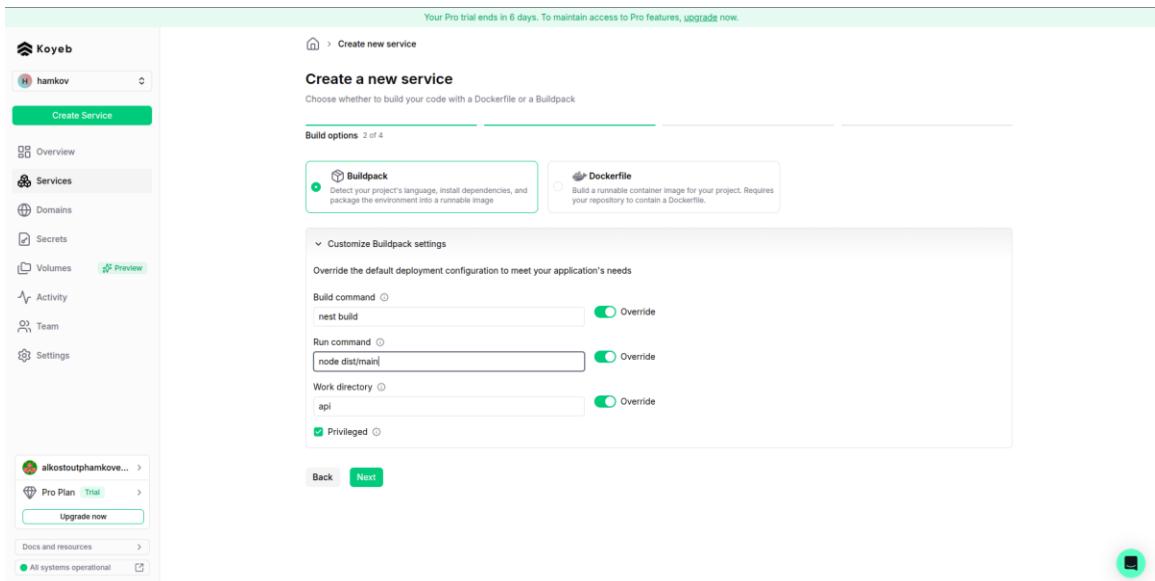
- La primera, utilizando un archivo **Dockerfile** dentro del repositorio.
- La segunda, mediante la herramienta **Buildpack** que trae Koyeb, la cual detecta automáticamente las dependencias, las instala y configura el

servicio.

Nosotros utilizaremos la **segunda opción (Buildpack)**, y además agregaremos algunas configuraciones adicionales.

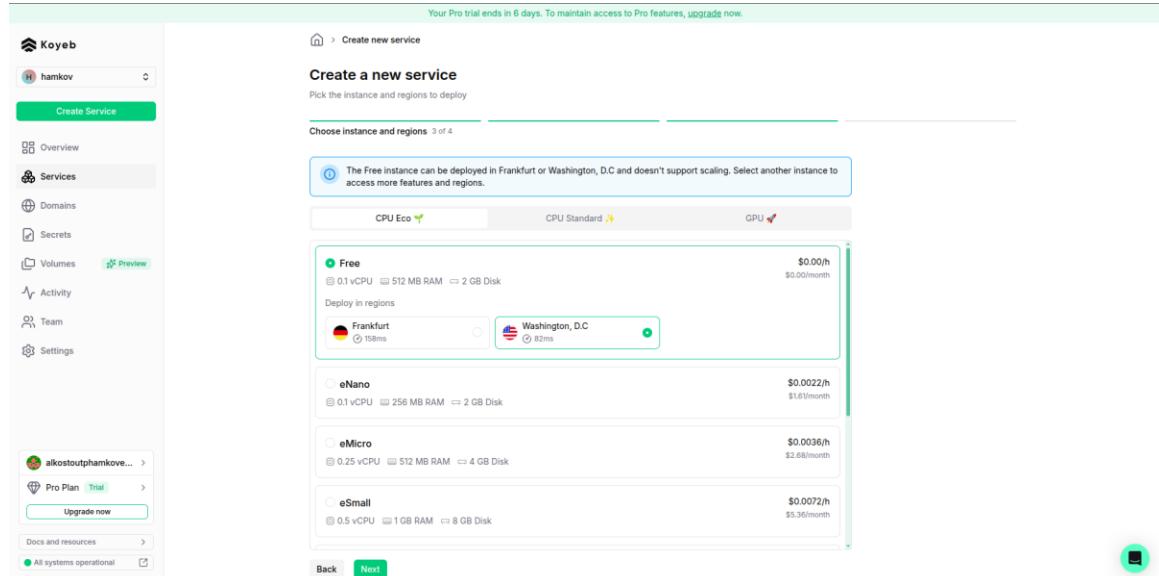


7. A continuación, damos clic en “Customize Buildpack settings” para agregar los valores necesarios que permitirán que el despliegue se realice correctamente. Luego, hacemos clic en el botón “Next”. Los valores de configuración del servicio web se muestran en la siguiente imagen.

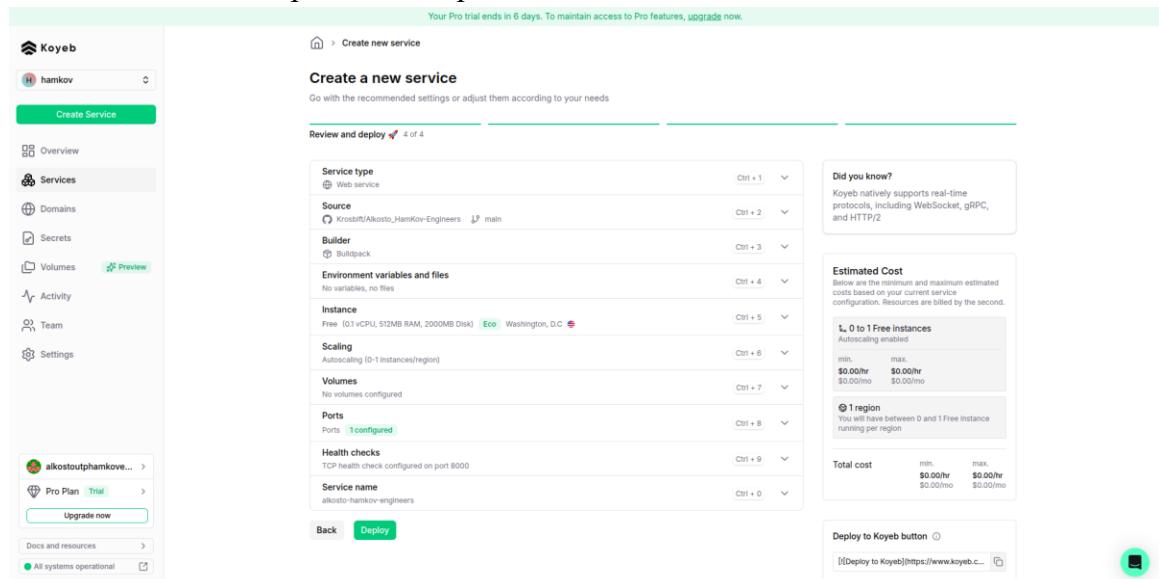


8. Posteriormente, el sistema nos solicitará que seleccionemos las **características del servicio web**, mostrando distintas opciones con precios diferentes. En nuestro

caso, debemos dar clic en la pestaña “CPU Eco” y elegir la **primera opción**, que corresponde al **plan gratuito (Free Plan)**. Luego, hacemos clic en “Next” para continuar.



9. A continuación, se mostrará una pestaña de resumen con todos los valores seleccionados previamente para la inicialización del servicio web.



10. El siguiente paso consiste en **agregar las variables de entorno** necesarias para que el backend (NestJS) funcione correctamente. Estas variables permiten que el servicio sepa con qué cuenta enviará los correos electrónicos, los números que usará para enviar los códigos por **SMS** y **WhatsApp**, así como la **cadena de conexión a la base de datos**. También debemos incluir las variables básicas de inicialización de NestJS, como el **HOST** y el **PORT**.

Environment variables and files

Edit variables and files

Ctrl + 4 ⌘

Environment variables Files

Configure manually or drag and drop file with multiple variables

Name	Value ⓘ
HOST	0.0.0.0
PORT	8000

+ Add another Raw editor

11. En la imagen anterior ya se observan configuradas las variables **HOST** y **PORT** con los valores correspondientes que debemos asignarles. Ahora procederemos a crear la variable que servirá como **cadena de conexión a la base de datos**. Para obtener este valor, debemos dirigirnos al servicio de **Neon**, donde tenemos desplegada nuestra base de datos de **PostgreSQL versión 17**.

HamKov Engineer Free hamkov

All ok ⚡ Upgrade H

PROJECT

- Dashboard
- Branches
- Integrations
- Auth
- Settings
- OVERVIEW
- Data API
- Monitoring
- SQL Editor
- Tables
- Backup & Restore
- Data Masking Beta
- Feedback
- Collapse menu

hamkov

Project dashboard

Get started with your new database

Connect to your database Get connection details your app needs to connect to your database Connect

Import your data Automatically import your data directly into your Neon database Import data

Connect from your IDE Use the Neon Local connect IDE extension to connect to your database Get extension

Integrate Neon with your AI tools Connect to MCP clients like Cursor, Claude Desktop, Cline, Zed, and Wardsurfer. Install MCP Server Go

Branches 2 / 10 Compute 0 / 100 CU-hrs Storage 0 / 0.5 GB Network transfer 0 / 5 GB

Usage since Nov 9, 2025. Metrics may be delayed by an hour and are not updated for inactive projects. Learn more.

Monitoring View all metrics

Branch production Default Compute Primary • idle Refresh

2 Branches View all

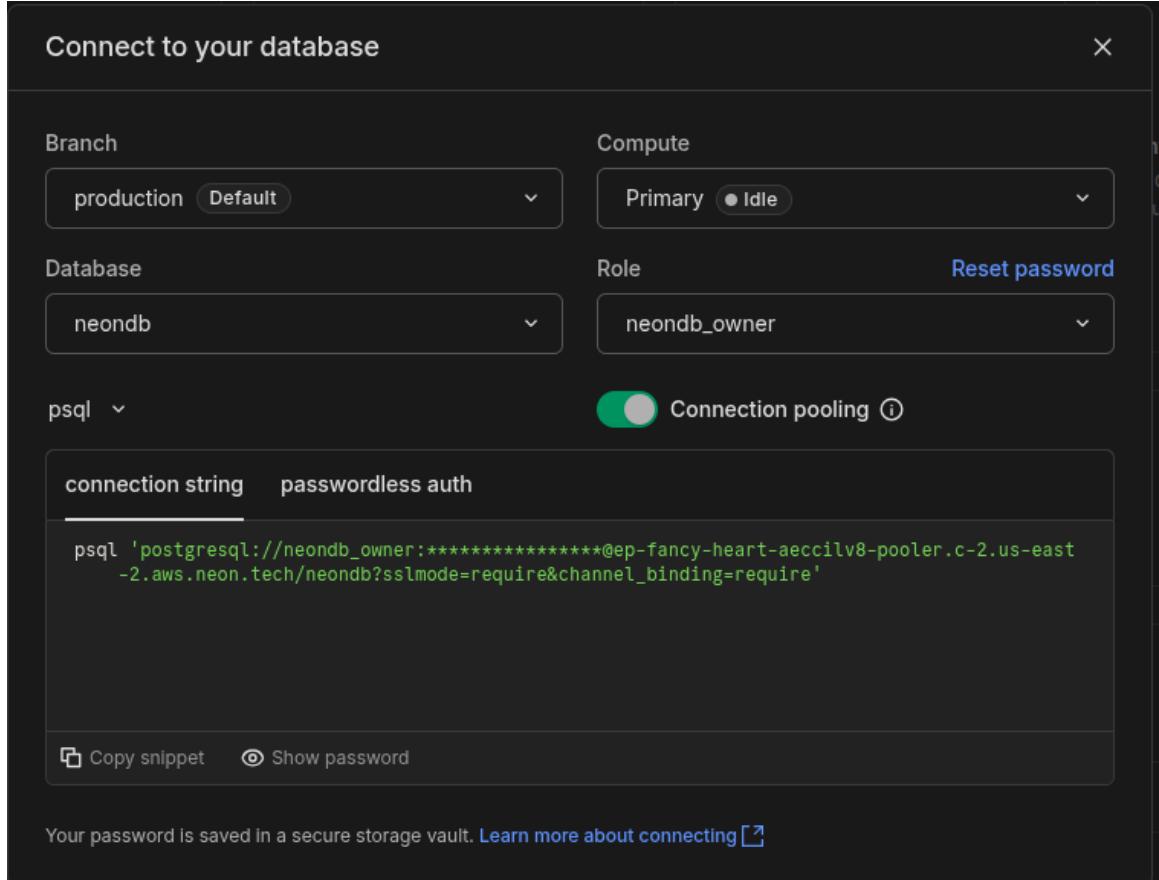
Name	Primary compute	Created by
production	.25 => 2 CU • idle	
development	.25 => 1 CU • idle	

Preview Workflow

You don't have any preview branches yet. Improve your workflow by adding database branching to your development previews.

Install an integration

12. Una vez dentro de la plataforma de Neon, hacemos clic en el botón ubicado en la sección “Connect to your database”. Al ingresar, se abrirá una ventana que mostrará las diferentes opciones de conexión disponibles para la base de datos. En esta ventana, hacemos clic en el botón “Copy snippet” para copiar la cadena de conexión completa.



13. Despues de copiar la cadena, volvemos a **Koyeb** e insertamos este valor como una **nueva variable de entorno**, siguiendo el formato que se muestra en la imagen siguiente.

14. A continuación, ingresamos al sitio web de **Twilio**. Allí crearemos una cuenta y seleccionaremos un **plan temporal gratuito** (Free Trial).

Una vez creada la cuenta, obtendremos las **credenciales necesarias**:

- **API Key**
- **Session SID**
- **Número para envío de SMS**
- **Número para envío de mensajes por WhatsApp**

Con estas credenciales, agregamos las variables de entorno correspondientes en **Koyeb**, tal como se muestra en la imagen siguiente.

TWILIO_SID	ACa65a18d42b03d19b86df02c892260293	
TWILIO_TOKEN	1e228dd3c94b75a63b8a41cde3cf87d	
TWILIO_SMS_NUMBER	+19787238685	
TWILIO_WHATSAPP_NUMBER	+14155238886	

[+ Add another](#) [Raw editor](#)

15. Por último, configuraremos el **servicio de correo electrónico**. Para ello, accederemos al correo institucional, copiaremos su dirección y crearemos una **contraseña de aplicación de Gmail**. Es importante tener activada la **autenticación en dos pasos**, ya que este requisito es necesario para generar la contraseña de aplicación. Una vez creada, copiamos la contraseña y la añadimos junto con la dirección de correo a las variables de entorno, de la siguiente manera.

EMAIL_USER	tucorreo@gmail.com	
EMAIL_PASSWORD	a x v j z g h j s d w a a n s q	

[+ Add another](#) [Raw editor](#)

16. Finalmente, todas nuestras variables de entorno deberían verse como se muestra en la siguiente imagen. (*Nota: los datos visibles en la imagen son únicamente de prueba y ya fueron deshabilitados, por lo que no tienen validez actual.*)

Environment variables and files

Edit variables and files Ctrl + 4 ^

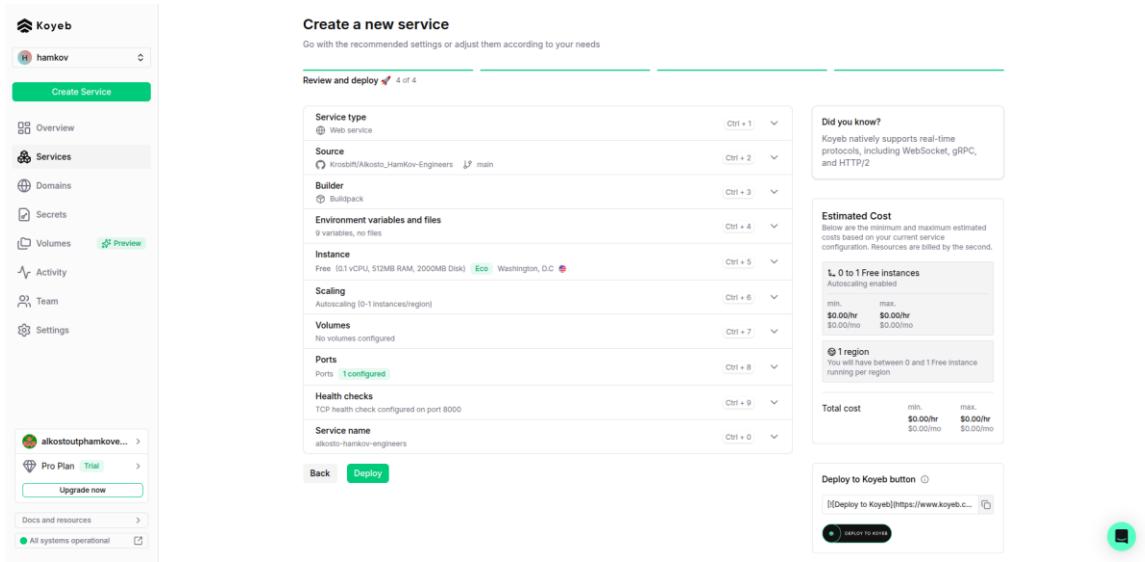
Environment variables Files

Configure manually or drag and drop file with multiple variables

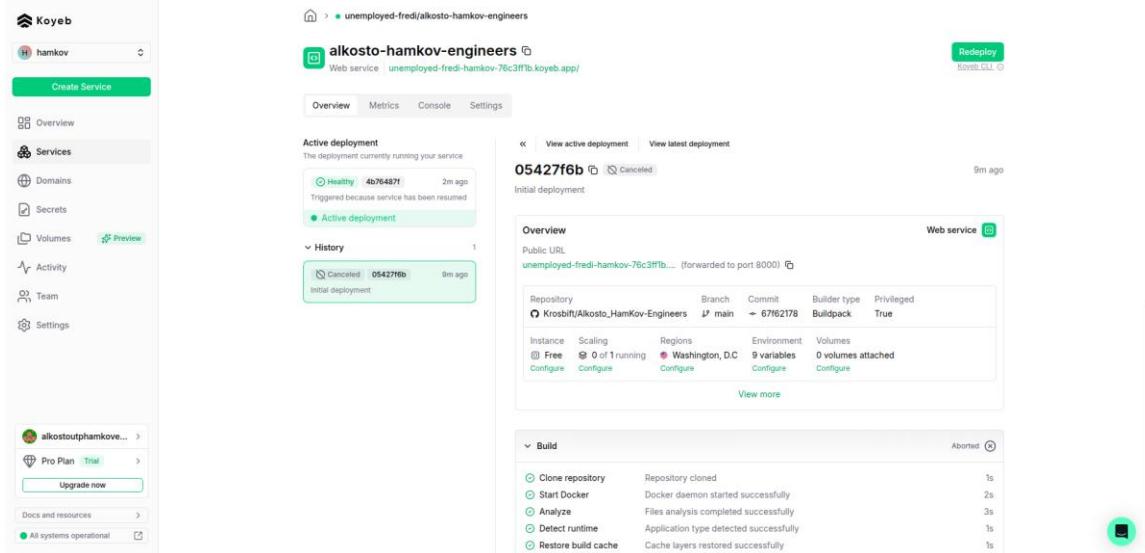
Name	Value <small> ⓘ</small>	
HOST	0.0.0.0	
PORT	8000	
DATABASE_CONNECTION_STRING	postgresql://neondb_owner:npg_mWr2lkBhS... <small>Copy</small>	
TWILIO_SID	ACa65a18d42b03d19b86df02c892260293 <small>Copy</small>	
TWILIO_TOKEN	1e228dd3c94b75a63b8a41cde3cfa87d <small>Copy</small>	
TWILIO_SMS_NUMBER	+19787238685 <small>Copy</small>	
TWILIO_WHATSAPP_NUMBER	+14155238886 <small>Copy</small>	
EMAIL_USER	alkosto.utp.hamkov.engineers@gmail.com <small>Copy</small>	
EMAIL_PASSWORD	a u v j x g h j s m w a a n s j l <small>Copy</small>	

[+ Add another](#) [Raw editor](#)

17. Viendo la página web como se muestra en la siguiente imagen, ya solo queda hacer clic en el botón “**Deploy**” y esperar a que el proceso de despliegue se complete. A continuación, se mostrará una nueva página con el **progreso de la implementación**, donde podremos observar en tiempo real las diferentes etapas del proceso, como la instalación de dependencias, la compilación del proyecto y la inicialización del servicio. Este procedimiento puede tardar algunos minutos, por lo que se recomienda esperar hasta que el estado cambie a “**Deployment successful**”, indicando que el backend ha sido desplegado correctamente en **Koyeb** y se encuentra en funcionamiento.



18. Si todo se ejecuta correctamente terminaremos viendo la siguiente pantalla.



19. Ahora solo nos queda probar un endpoint para terminar de comprobar si está todo funcional o en su defecto abrir el front en local y cambiar la url base de la api para probarlo así también para termina viendo algo como lo siguiente.

Screenshot of the ALKOSTO website showing a grid of mobile phones for sale, followed by a screenshot of the Network tab in the developer tools showing a request for a product image.

ALKOSTO Website Screenshot:

- Header:** Includes the ALKOSTO logo, navigation links (Tecnología, Electrodomésticos, Llantas y vehículos, Muebles y colchones, Hogar, Juguetes, Deportes, Pines, Ropa Géneros, Mercado), a search bar ("¿Qué buscas hoy?"), and user account links (Mi cuenta, Mi carrito).
- Product Grid:** Displays eight products in a 2x4 grid:
 - Top Row: iPhone 13 128GB Blanco Estrella (COP 2,579,010)
 - Top Row: iPhone 17 Pro Max 256GB 5G Azul Oscuro (COP 6,999,010)
 - Top Row: iPhone 13 128GB Azul Medianoche (COP 2,579,010)
 - Top Row: iPhone 17 Pro 256GB 5G Azul Oscuro (COP 6,439,010)
 - Bottom Row: iPhone 16e 128GB Negro (COP 2,579,010)
 - Bottom Row: Celular SAMSUNG Galaxy A36 256GB 5G Negro (COP 6,439,010)
 - Bottom Row: Celular SAMSUNG Galaxy A16 128GB LTE 4G Negro (COP 6,439,010)
 - Bottom Row: Celular SAMSUNG Galaxy A26 256GB 5G Negro (COP 6,439,010)

Developer Tools Network Tab Screenshot:

- Network Tab Headers:** Shows the URL (https://unemployed-fredi-hamkov-76c3ff1b.koyeb.app/products/filtered?idCategoria=2&idMarca=1), Request Method (GET), Status Code (200 OK), and Response Headers (Content-Type: application/json; charset=utf-8, Date: Sun, 09 Nov 2025 22:38:28 GMT, etc.).
- Request Headers:** Shows the client's request headers including Authority, Method, Path, Scheme, Accept, Accept-Encoding, Accept-Language, Origin, Priority, and Referer.
- Timing Bar:** A horizontal timeline showing the duration of each request from 10 ms to 130 ms.
- Table:** A detailed table of requests, showing columns for Name, Headers, Payload, Preview, Response, Initiator, and Timing.

Instalación del Frontend (Angular en Netlify)

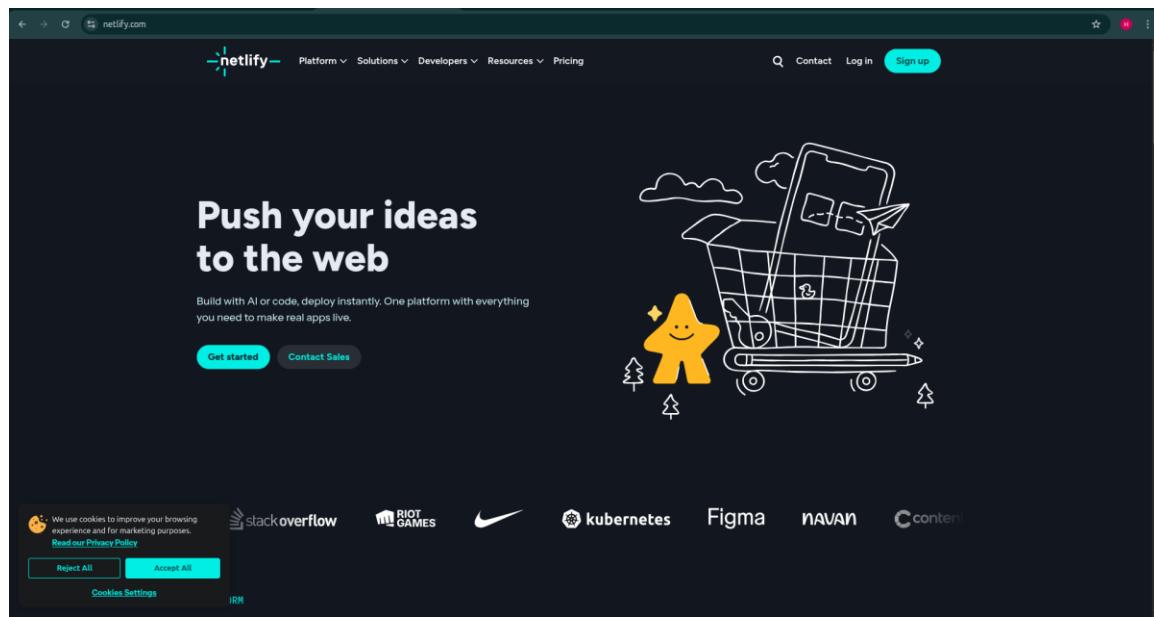
Objetivo: publicar el frontend Angular del proyecto Alkosto_UTP (carpeta `/app`) en Netlify utilizando el plan gratuito, conectando el repositorio de GitHub y configurando la URL base de la API para apuntar al backend desplegado en Koyeb.

1) Prerrequisitos

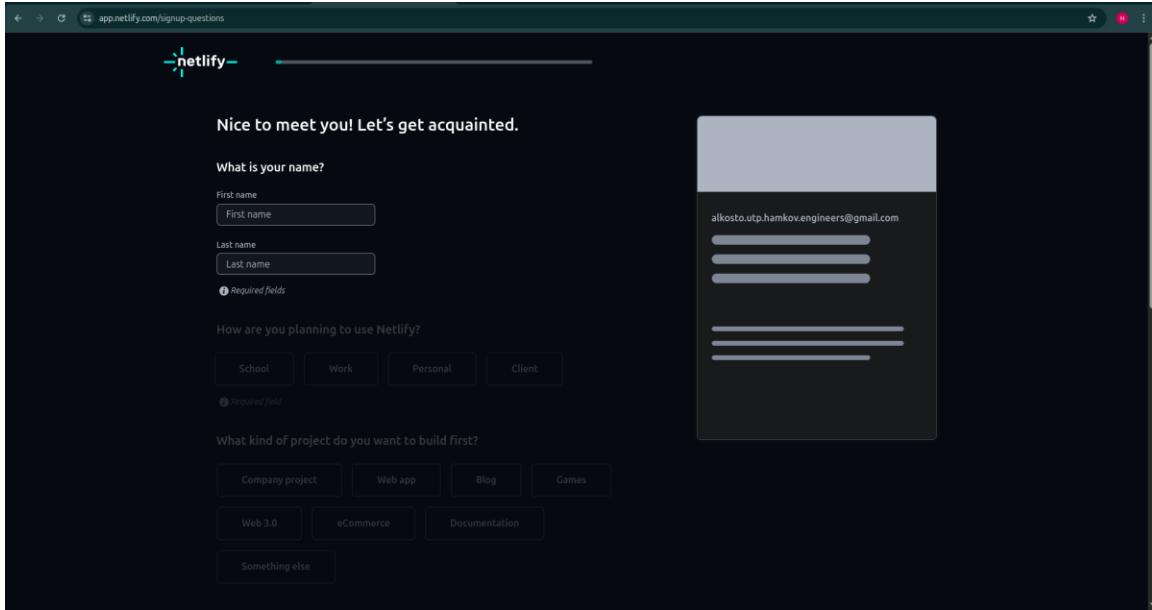
- Cuenta en GitHub con acceso al repositorio `Krosbift/Alkosto_HamKov-Engineers`.
- Haber creado la base de datos en **Neon**.
- Backend **NestJS** ya desplegado en Koyeb y URL pública disponible (por ejemplo: <https://alkosto-utp-api-xxxx.koyeb.app>).
- Node.js 20+ y Angular CLI instalados solo si vas a construir localmente (en Netlify no es necesario).

2) Pasos para la instalación.

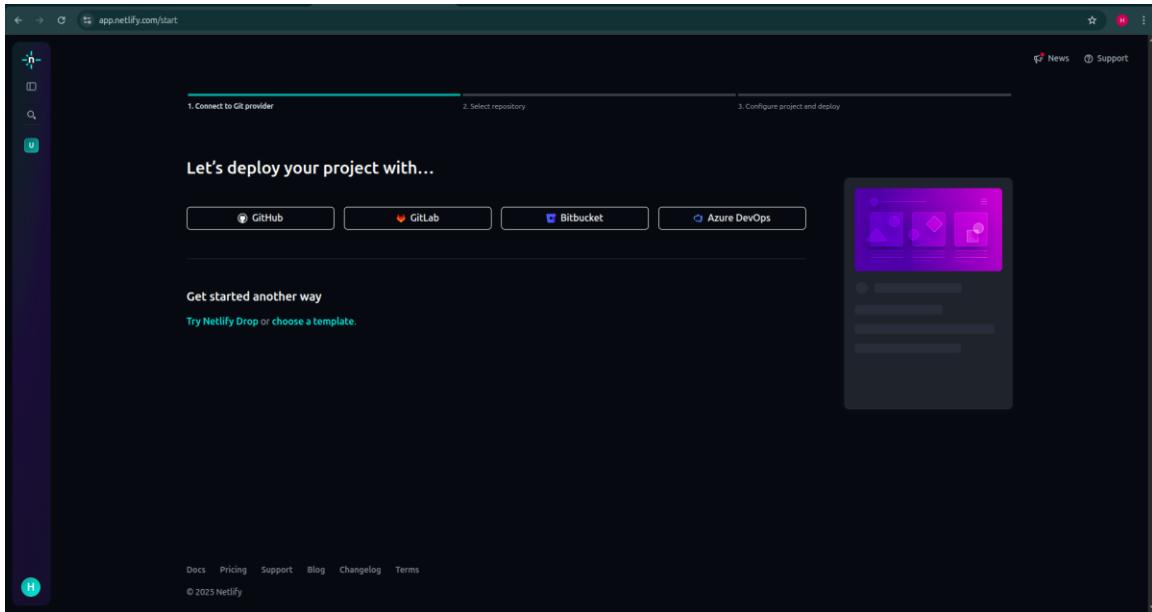
1. Lo primero que debemos hacer es ingresar al sitio web de [Netlify](#)



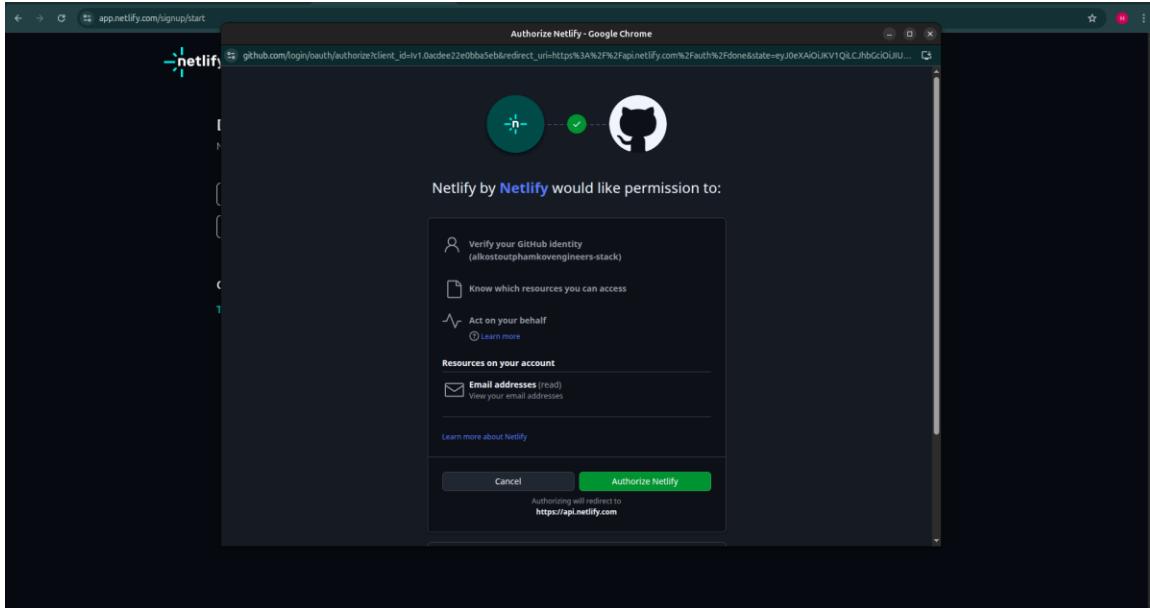
2. Una vez dentro, nos registramos y veremos la siguiente pantalla.



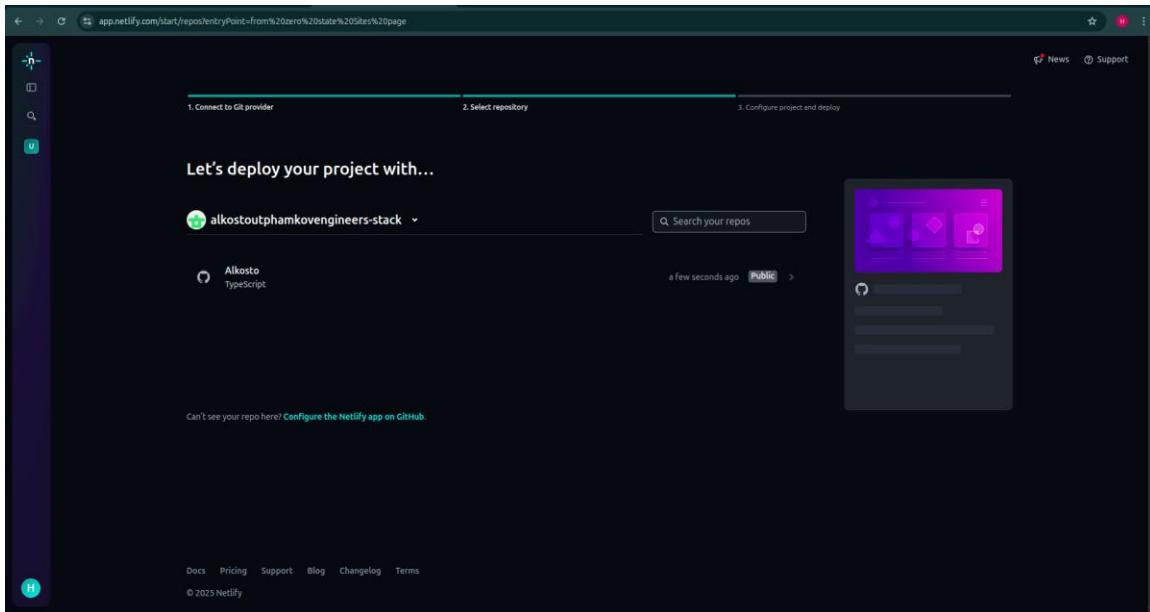
3. Luego pasará algo similar que en Koyeb, nos pedirán datos sobre quienes somos, simplemente los llenamos con normalidad y pasamos a la página siguiente donde nos diran que despleguemos nuestro primer proyecto pero por ahora lo omitimos.



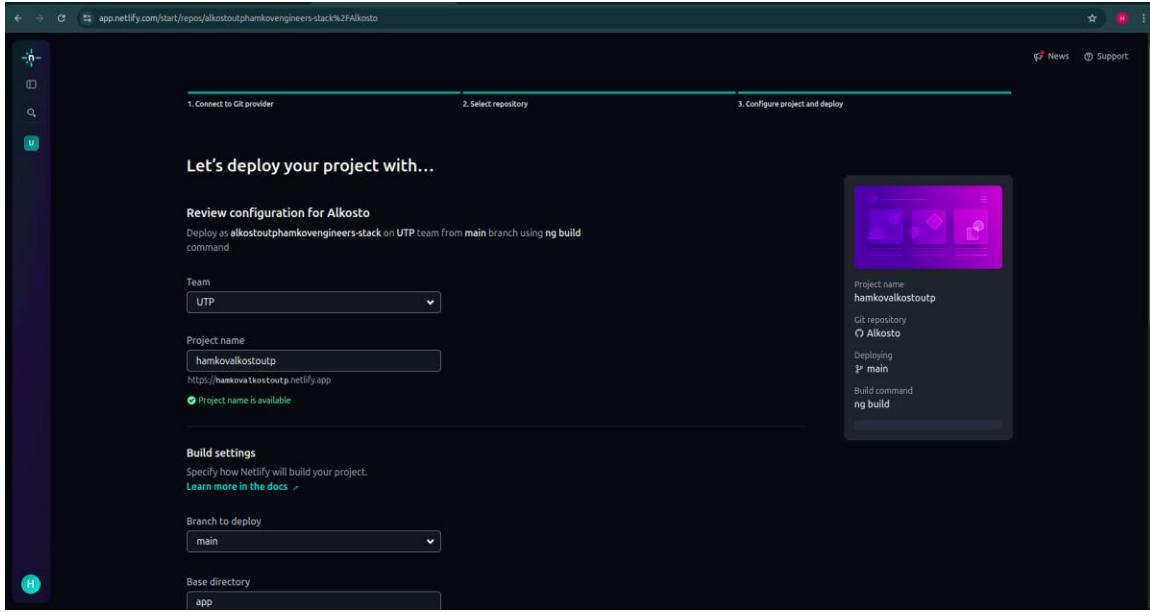
4. En esta página nos dirán a través de que servicio queremos desplegar, nosotros elegiremos GitHub como ya lo hicimos anteriormente.



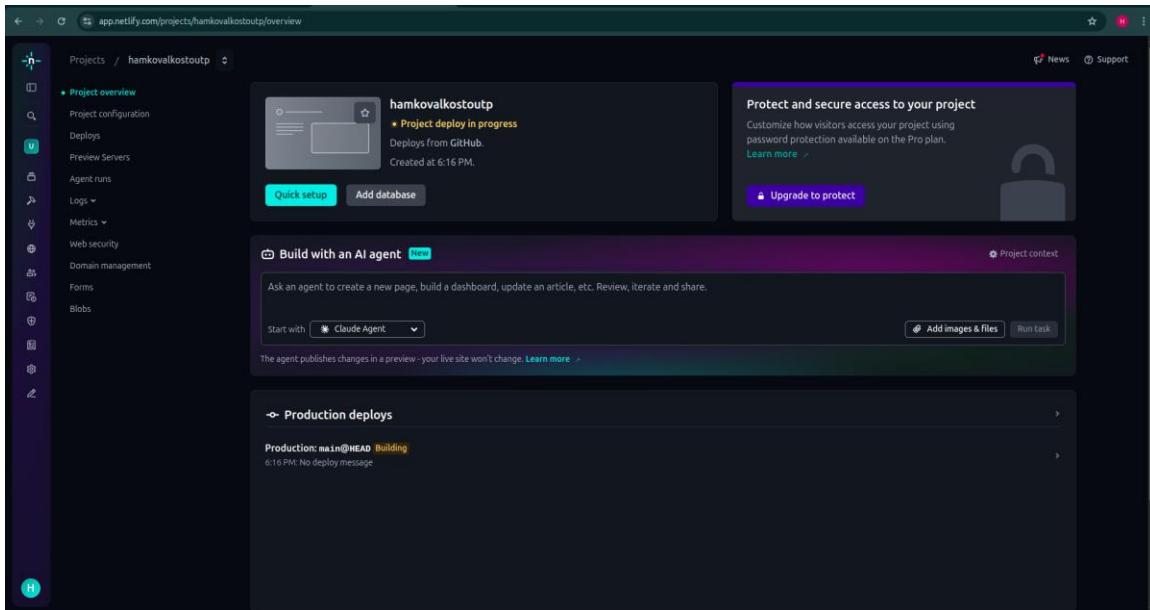
5. Esto nos abrirá una ventana para darle acceso a Netlify con GitHub, le diremos que sí y continuaremos. Luego nos saldrá una nueva ventana donde nos diran que elijamos el proyecto, pero este debe de estar si o si como un repositorio personal, entonces deberemos de clonar el repositorio original de alkosto y subirlo a nuestro repositorio personal para que nos aparezca y ahí si lo seleccionamos.



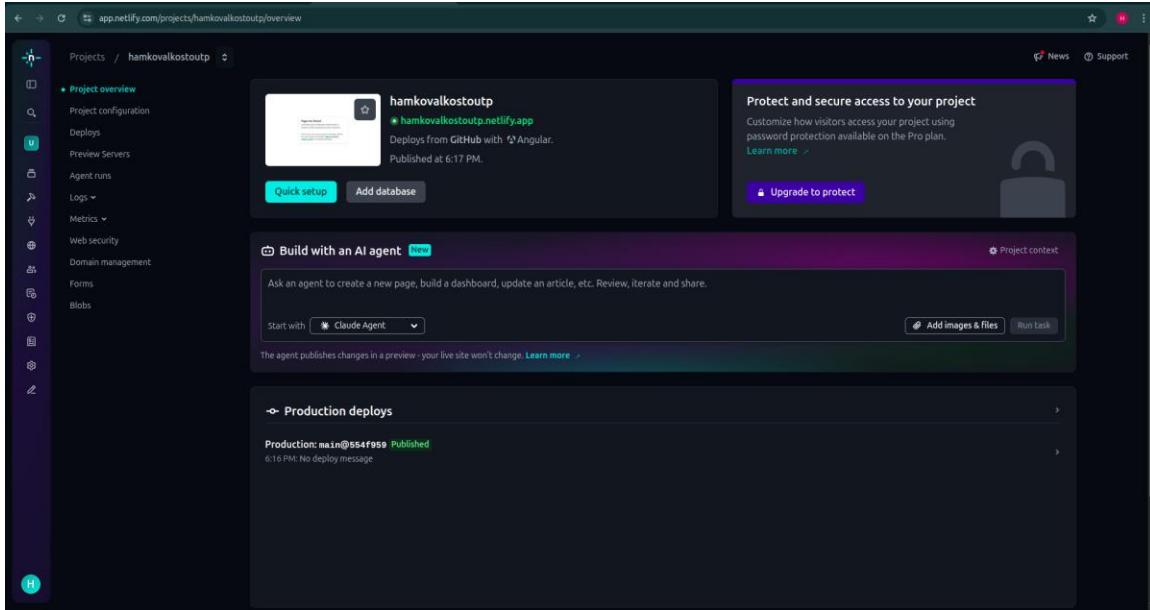
6. Daremos click en el repositorio de Alkosto que clonamos desde el sitio original.



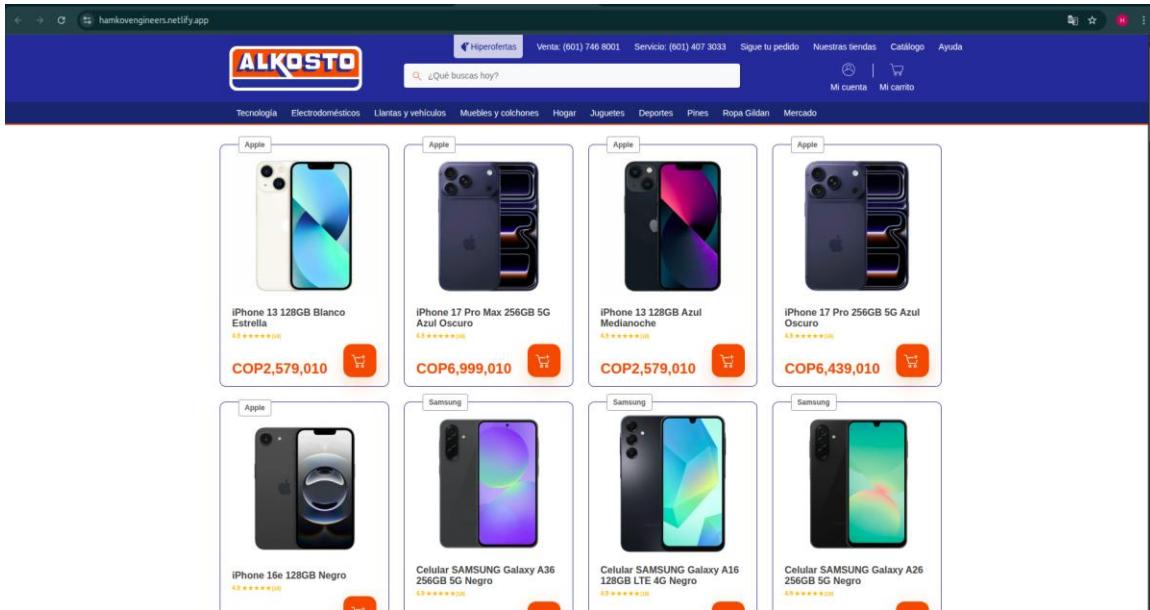
7. Luego ingresamos todos los datos entre ellos el nombre del proyecto, seleccionamos la rama principal, no olvidar agregar la carpeta de trabajo que es app y el comando de compilación de Angular que es ng build. Luego damos click en Deploy.



8. Ahora estará construyendo la aplicación y solo debemos esperar hasta que esta termine para poder validar si funcionó correctamente.



9. Ahora que nos aparece lo de la imagen anterior ya sabemos que la aplicación se desplegó correctamente, ahora solo nos queda validar que el sitio web cargue correctamente.



Como vemos todo se desplegó de forma correcta y sin ninguna complicación.