



# MANUAL DE INSTALACIÓN PLATAFORMA NLP

Autor

**Javier Luis Sepúlveda Hernandez**

Fecha

**August 14, 2023**

## TABLA DE CONTENIDOS

TABLA DE CONTENIDOS .....	2
revisión de historial.....	2
1    Introducción .....	3
2    Base de datos .....	4
2.1    Instalación de PostgreSQL.....	4
2.2    Configuración inicial del software. ....	9
2.3    Carga de la base de datos.....	12
3    BACKEND .....	15
3.1    Requerimientos de software .....	15
3.2    Microservicio de gestión de contenido .....	16
3.3    Configuración de cuenta de servicio Google Cloud .....	21
3.4    Configuración de modelo en plataforma Vertex AI .....	21
3.5    Microservicio de procesamiento NLP .....	24

## REVISIÓN DE HISTORIAL

Versión	Fecha	Descripción	Autor
1	14/08/2023	Versión inicial	Javier Sepúlveda

# 1 INTRODUCCIÓN

El presente documento servirá como guía de instalación de la plataforma NLP. Cabe destacar que este manual está diseñado para el sistema operativo Windows 10/11, considerando a lo menos 2 GB de memoria RAM disponible a la hora de su ejecución.

La aplicación permite el ingreso de documentos PDF para el análisis de preguntas abiertas mediante herramientas de procesamiento de lenguaje natural, el uso de estas mediante solicitudes HTTP, así como la visualización de un gran volumen de datos, a través del uso de gráficas y métricas, pensando en el uso para la facultad de ingeniería de la Universidad de Santiago de Chile.

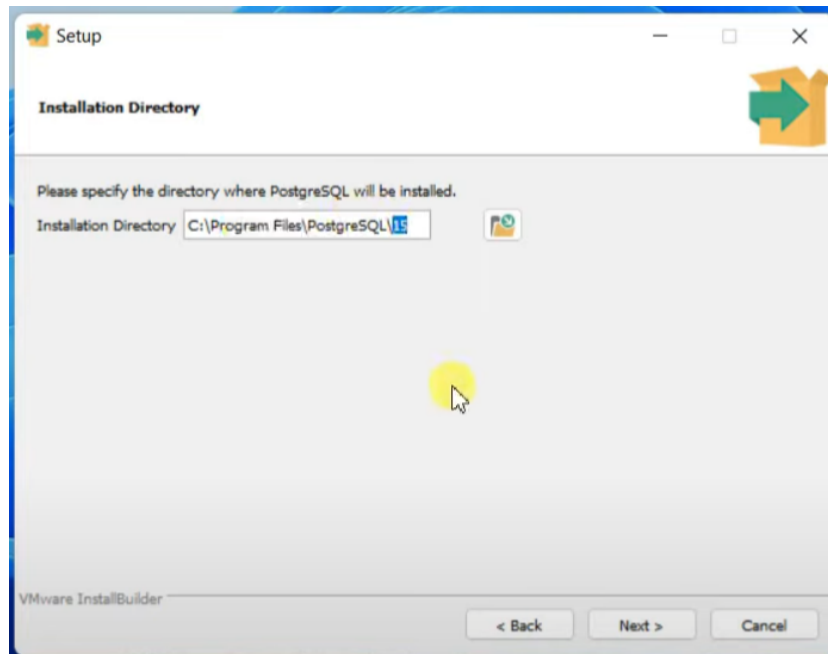
## 2 BASE DE DATOS

Esta etapa se abordará la instalación del software de persistencia de datos llamado PostgreSQL, y la carga de datos necesarios del proyecto en forma de backup.

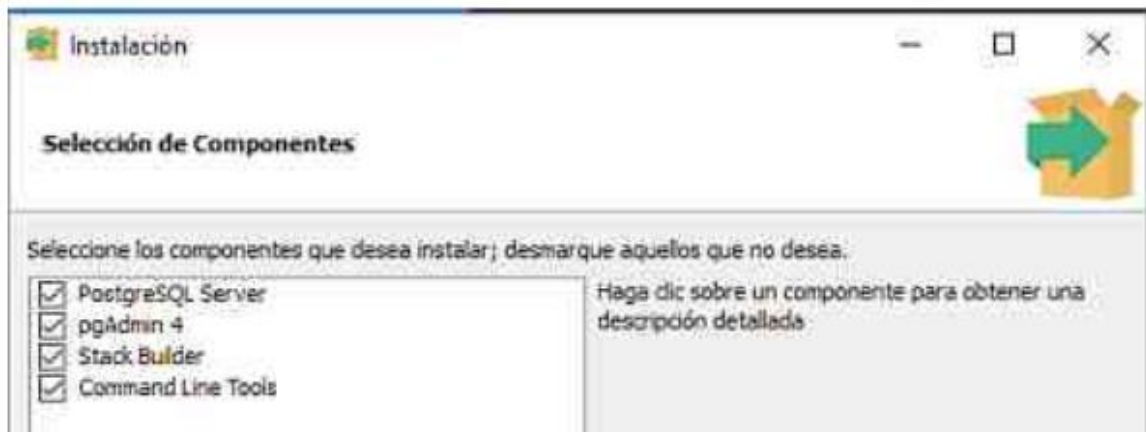
### 2.1 Instalación de PostgreSQL

\* Saltar este paso si es que ya se encuentra instalado en el sistema.

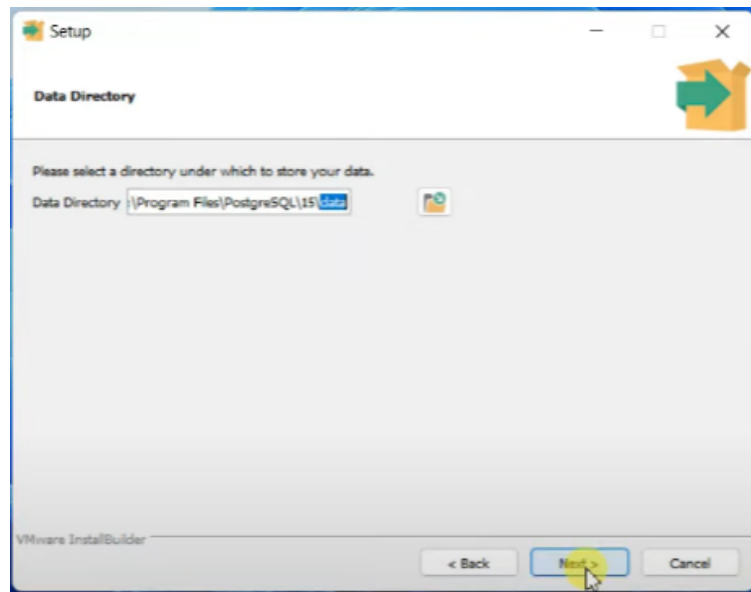
1. Descargar la versión correspondiente según el sistema operativo que posea siguiendo el siguiente enlace. La versión 15 y pgAdmin 4. [PostgreSQL: Downloads](#)
2. Abrir archivo ejecutable que se encuentre en las descargas
3. Definir donde se desea instalar y dar clic en **NEXT**.



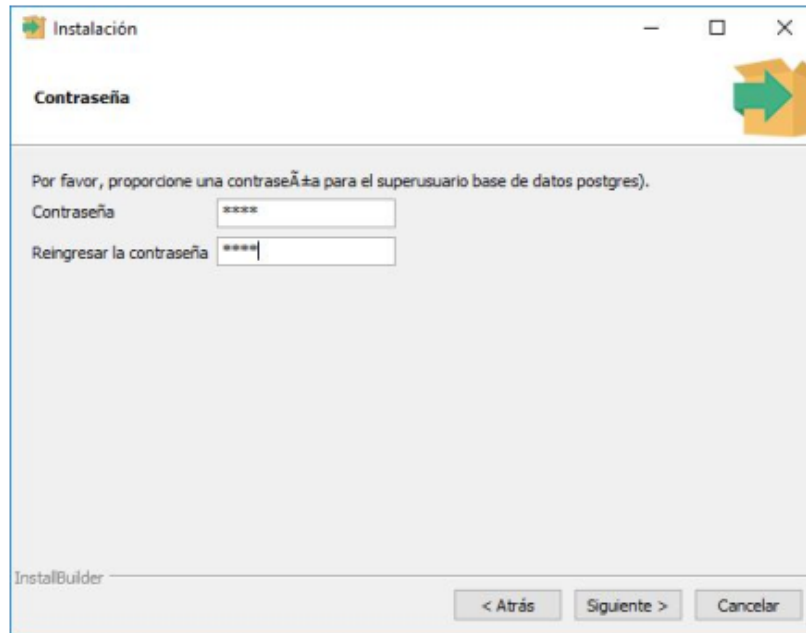
4. De los siguientes componentes sólo son necesarios, **postgreSQL Server**, **pgAdmin 4** y **commandLineTools**, stack builder no es necesario para el proyecto, y dar clic en **NEXT**.



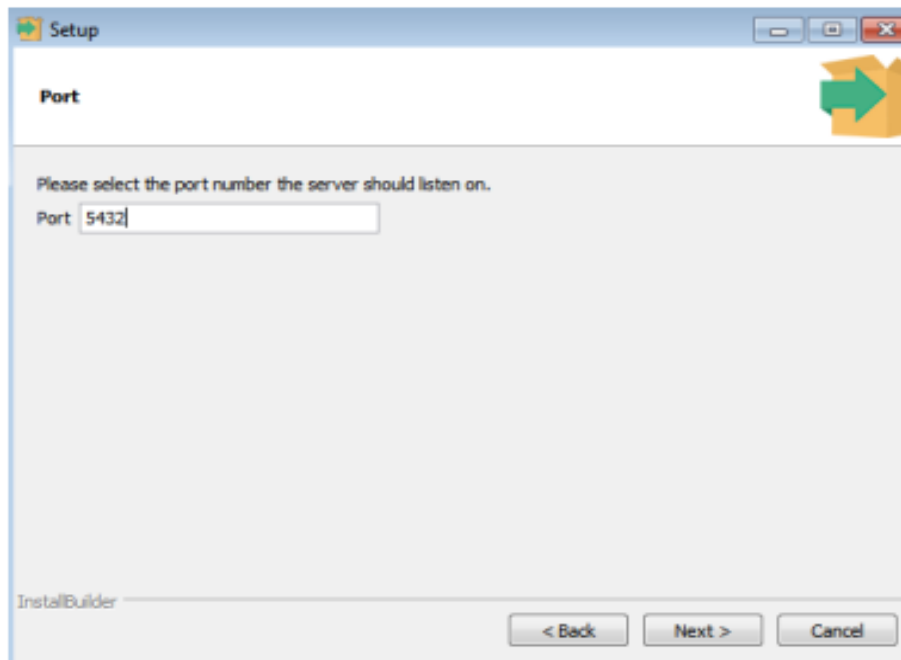
5. Definir donde se instalará la base de datos y dar clic en **NEXT**.



6. Definir contraseña para el super usuario de la base de datos, y dar clic en **NEXT**.  
*Recordarla para usarla luego, ya que será utilizada para acceder al sistema.*

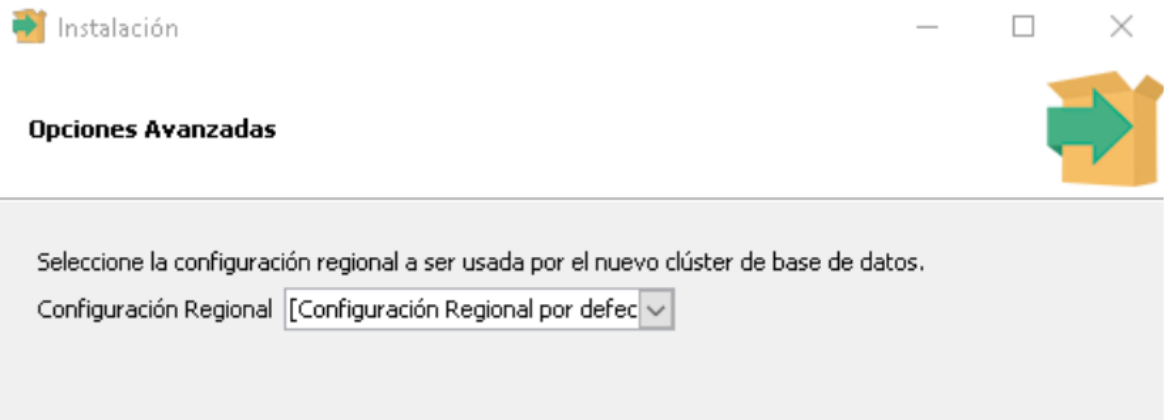


7. Definir puerto por defecto y dar clic en **NEXT**.



***Se recomienda no modificar el puerto por defecto "5432" para no tener inconvenientes posteriores.***

8. Seleccionar la configuración regional que viene por defecto y dar clic en **NEXT**.



9. **OPCIONAL:** Seleccionar la casilla “Stack Builder”, el cual es utilizado para descargar e instalar herramientas adicionales, controladores y aplicaciones para complementar la instalación de Postgres.

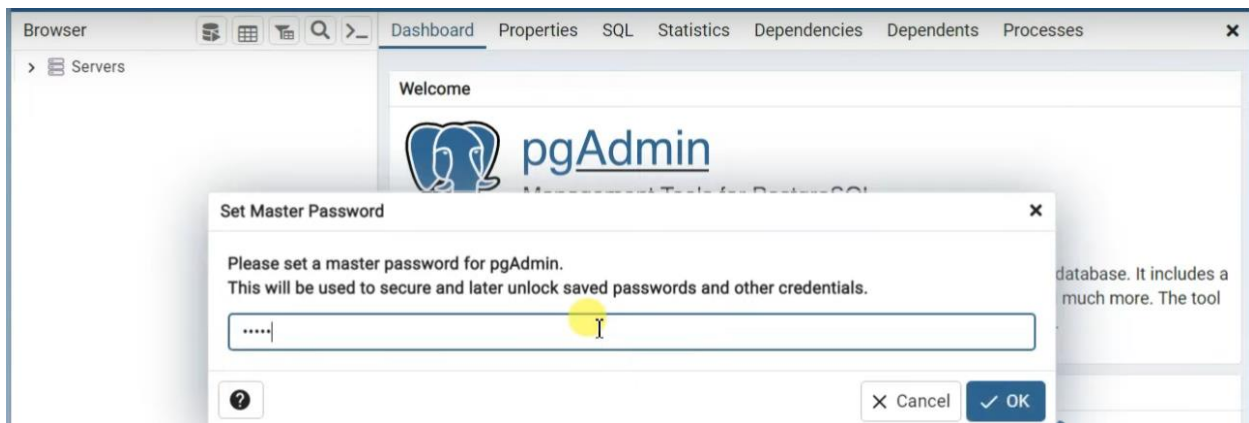




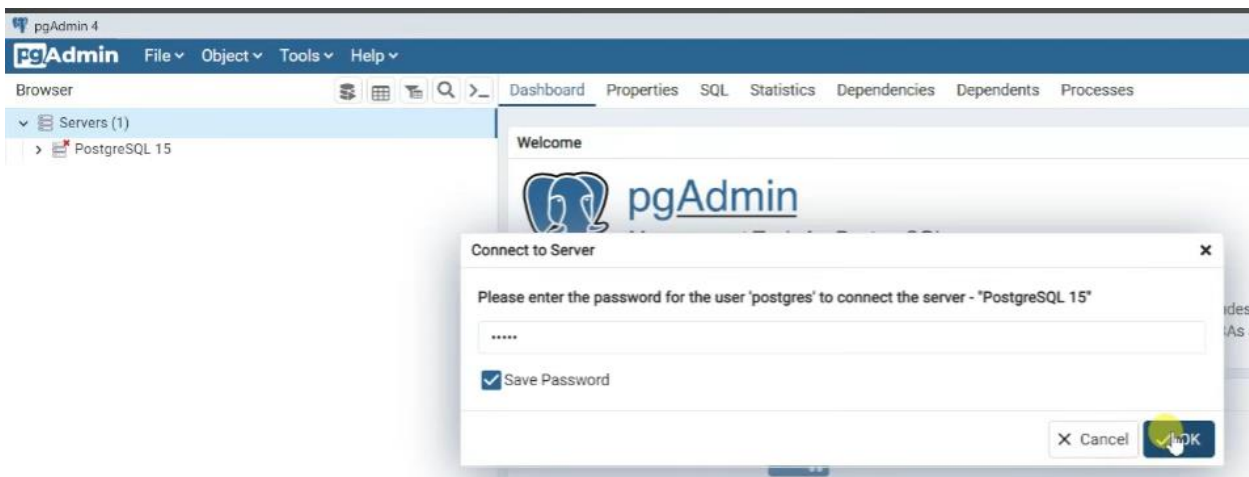
## 2.2 Configuración inicial del software.

\* Saltar este paso si es que ya se encuentra instalado en el sistema.

1. Al iniciar por primera vez mediante el software pgAdmin4 este le solicitara definir una contraseña maestra para ocupar el programa. **Recordarla para usarla luego, ya que será utilizada para usar esta interfaz y dar clic en OK.**

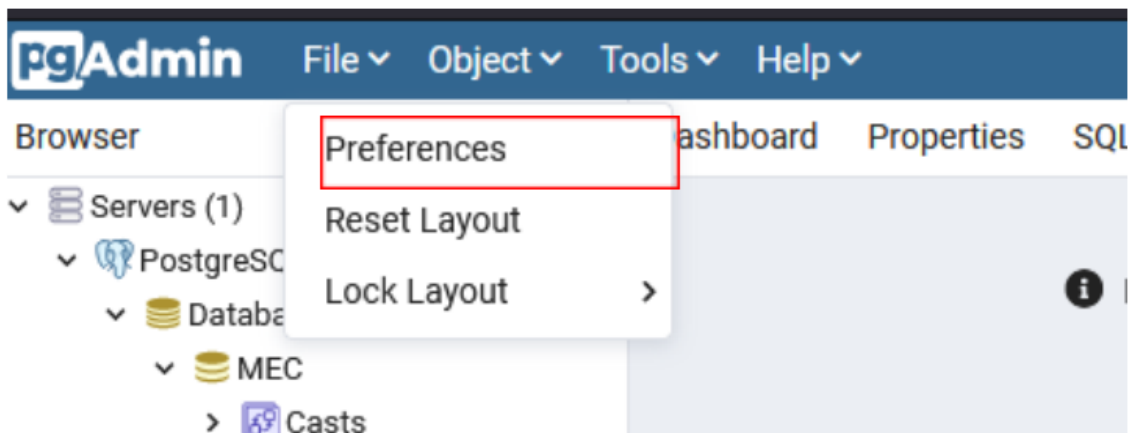


2. Al intentar acceder a los servidores creados y desglosar las bases de datos pgAdmin4 solicita la clave del super usuario creado en el paso 6 del punto [2.1](#)

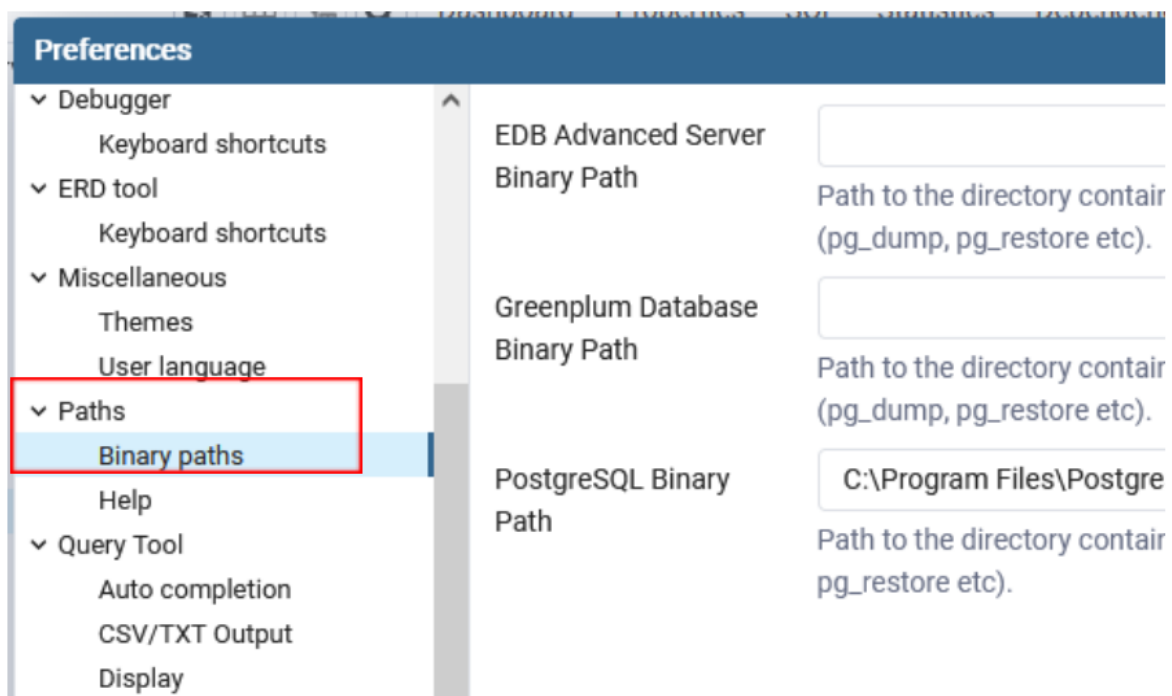


## 2.2.1 AGREGAR PATH

1. Agregar el binario de postgres al PATH de pgAdmin, para eso daremos click en FILE -> preferences

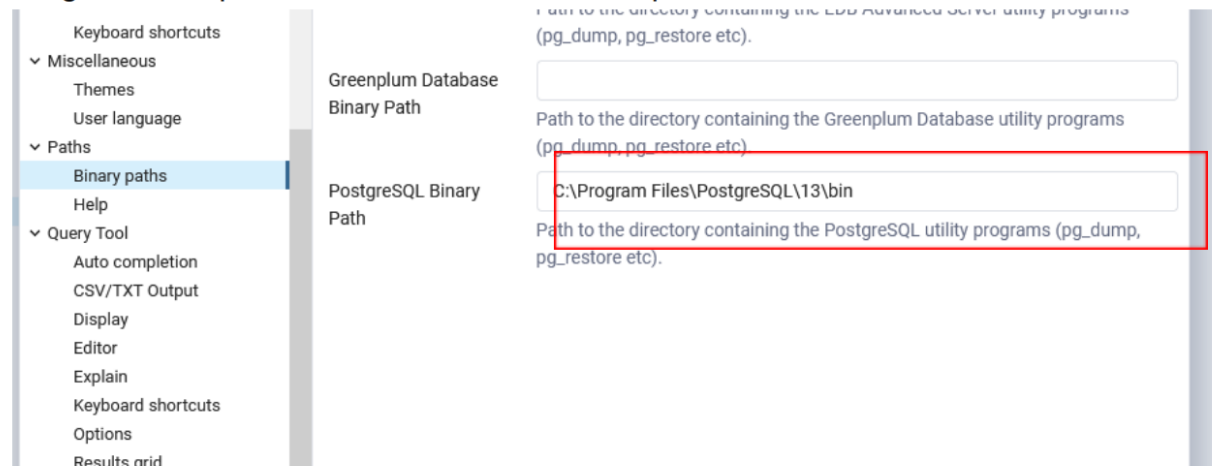


2. A la izquierda vamos a buscar la opción **Paths** -> **Binary paths**



3. En la sección de PostgreSQL **Binary Path** vamos a agregar la dirección de la carpeta “bin” de postgres (revisar directorio donde se instaló PostgreSQL). Luego de esto podemos cerrar la ventana de preferencias.

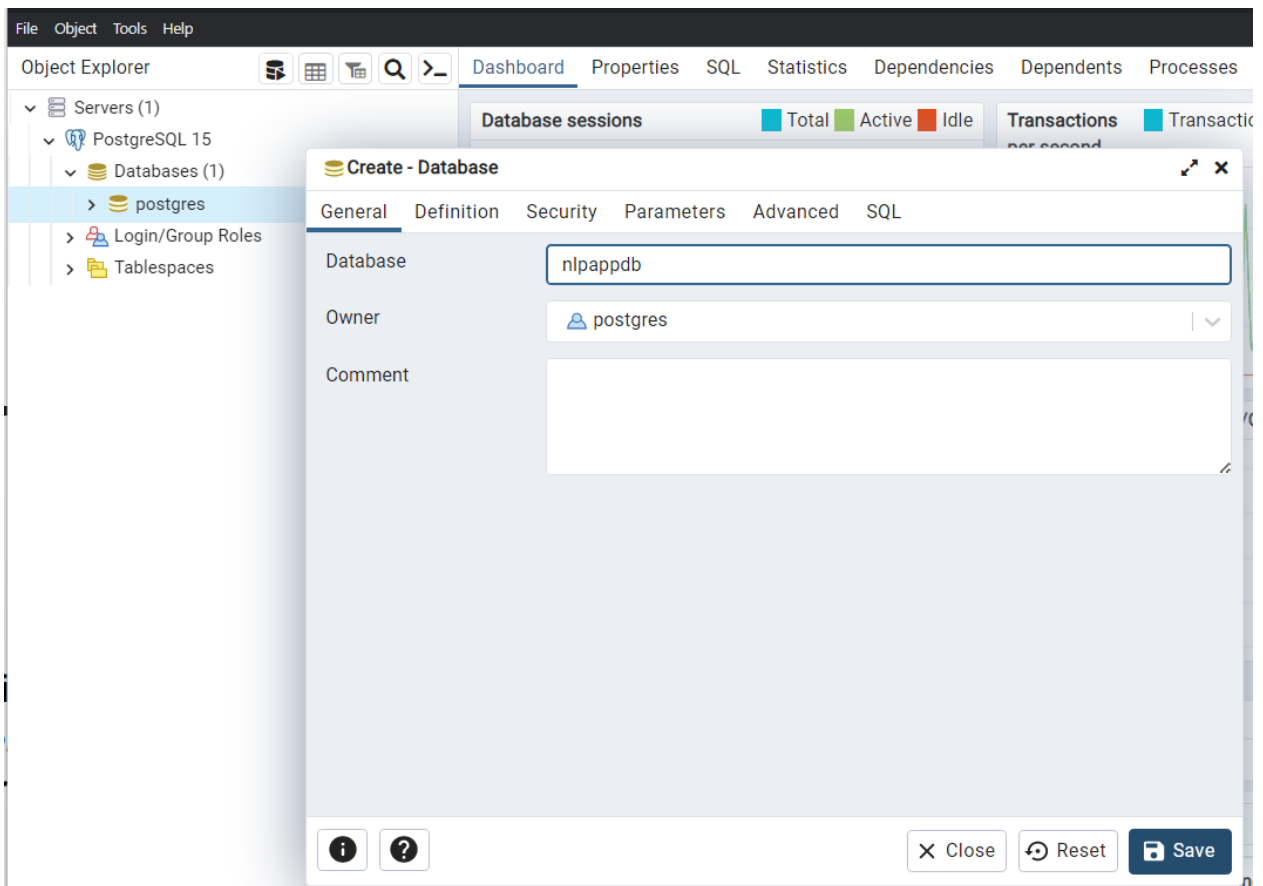
Luego de esto podemos cerrar la ventana de preferencias.



## 2.3 Carga de la base de datos

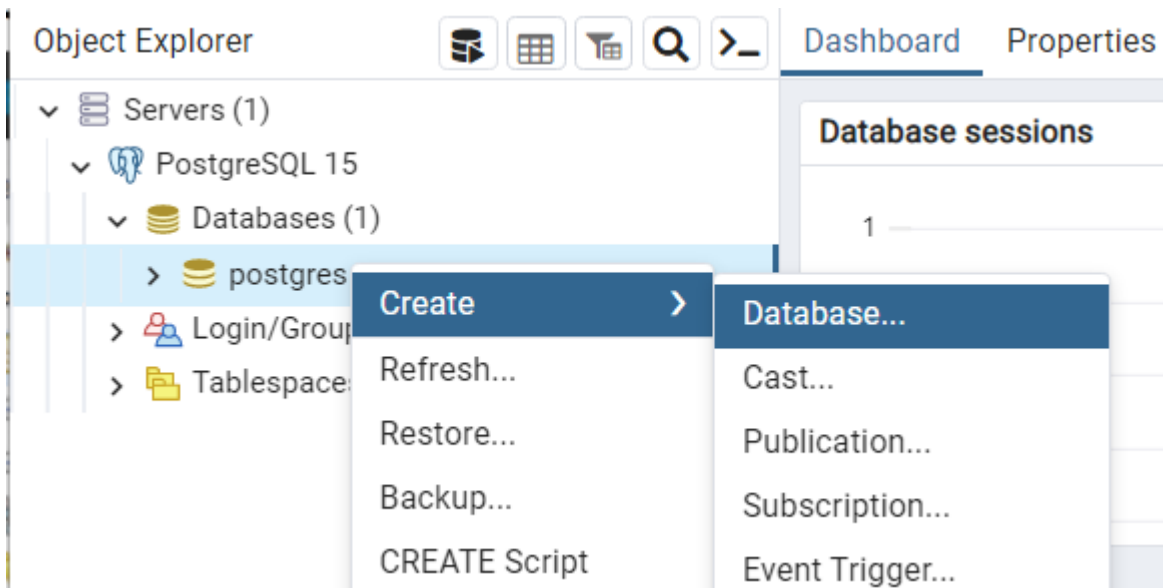
Luego de completar la instalación se ocupará un archivo de respaldo de la base de datos para su carga, denominada **backup.sql** (se encuentra en la carpeta comprimida en el directorio del proyecto \BD). Para ello se debe seguir los siguientes pasos.

1. Dar clic derecho en **Databases**, escoger la opción **Create** y a continuación **Database...**

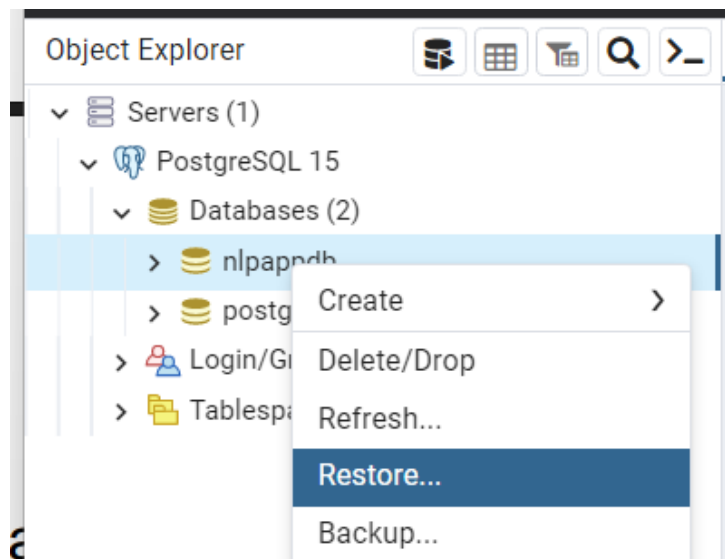


2. A continuación, se desplegará una ventana que permitirá definir le nombre de la base de datos, luego de esto dar clic en **SAVE**.

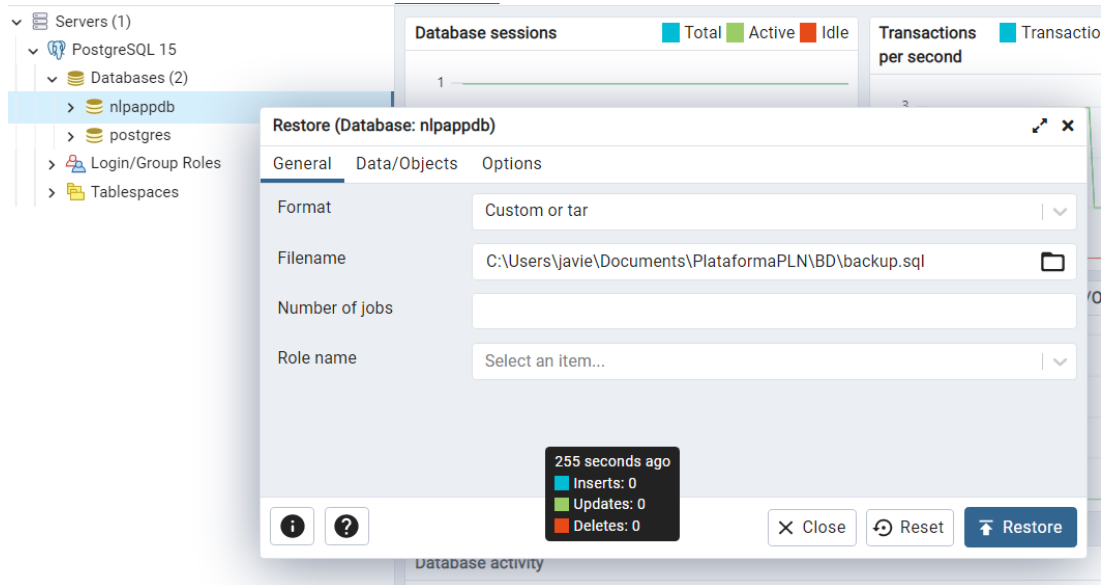
**Recomendación:** Definir nombre de la base de datos “*nlppapdb*” para no tener inconvenientes con las futuras configuraciones.



3. Hacer clic derecho sobre la base de datos creada y escoger la opción **RESTORE**



4. Buscar la ubicación del archivo backup en formato .sql en el directorio \BD. Hacer clic sobre el archivo y dar la opción **RESTORE**.



### 3 BACKEND

Todas las instrucciones en esta sección deben ser ejecutadas como prioridad en la instalación. En esta sección se detalla los requerimientos previos, como levantar el servicio que controla el manejo de contenido de la aplicación referente a la información de los usuarios y el servicio que controla el flujo de peticiones y datos.

#### 3.1 Requerimientos de software

Para lo antes mencionado se necesita tener instalado el entorno de ejecución Node.js y el manejador de paquetes

##### 1. Instalar Node.js y NPM

- 1.1. Desde el navegador, ir hacia la página <https://nodejs.org/en/download/>, luego hacer clic en **Windows Installer**. El propio instalador incluye el manejador de paquetes NPM



- 1.2. Cuando el instalador termine de descargarse deberá de iniciarlo y ejecutar el software de instalación, este le preguntara la ubicación y deberá hacer clic en **NEXT**, hasta finalizar el proceso.

1.3. Para verificar la instalación podrá ejecutar a través de la consola \$node -v y \$npm -v

```
C:\Users\javie>npm -v
9.5.1

C:\Users\javie>node - v
Welcome to Node.js v18.16.0.
Type ".help" for more information.
> |
```

## 3.2 Microservicio de gestión de contenido

Dentro del directorio del proyecto se encuentra la carpeta llamada **CMS**, abrir una consola dentro de este y ejecutar el comando \$npm run dev. Con este podrá iniciar el servicio accediendo a las funcionalidades, para verificar el correcto inicio se deberá ver lo siguiente, de caso contrario deberá hacer el paso [2](#) de la base de datos.

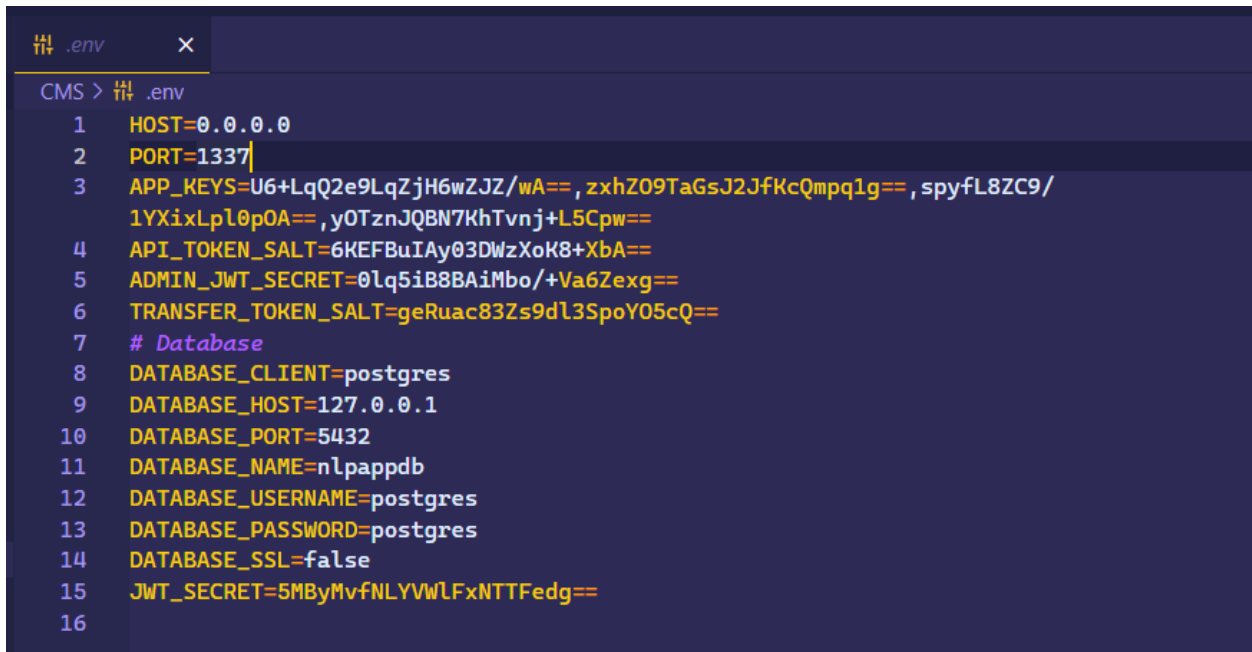
```
Actions available

Welcome back!
To manage your project 🚀, go to the administration panel at
http://localhost:1337/admin

To access the server ⚡, go to:
http://localhost:1337
```



Dentro del directorio del proyecto se encuentra un archivo llamado “.env” con las configuraciones del entorno, a lo cual la siguiente ilustración muestra la utilizada para la base de datos a modo de ejemplo.

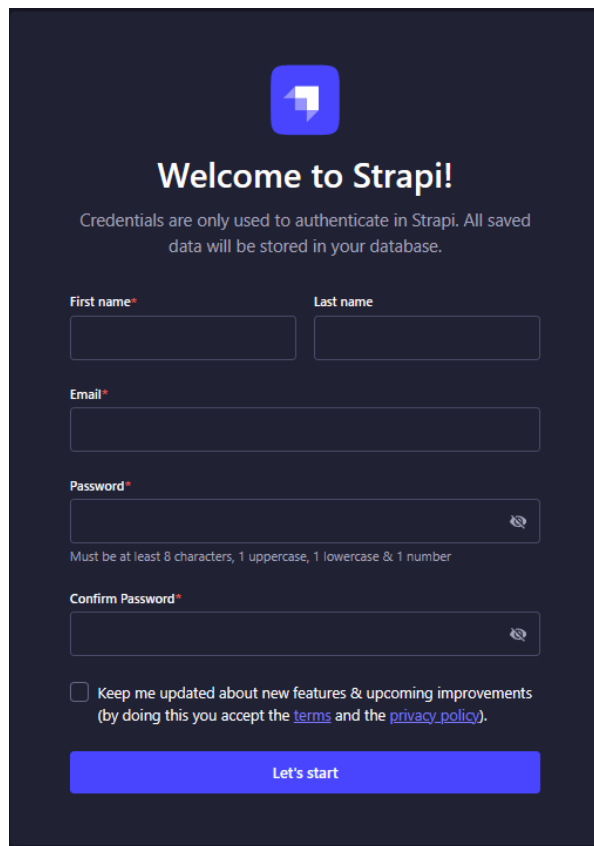


```
CMS > .env
1  HOST=0.0.0.0
2  PORT=1337
3  APP_KEYS=U6+LqQ2e9LqZjH6wZJZ/wA==,zxhZ09TaGsJ2JfKcQmpq1g==,spyfL8ZC9/
   1YXixLp10p0A==,yOTznJQBN7KhTvnj+L5Cpw==
4  API_TOKEN_SALT=6KEFBuIAy03DWzXoK8+XbA==
5  ADMIN_JWT_SECRET=0lq5iB8BAiMbo/+Va6Zexg==
6  TRANSFER_TOKEN_SALT=geRuac83Zs9dL3SpoY05cQ==
7  # Database
8  DATABASE_CLIENT=postgres
9  DATABASE_HOST=127.0.0.1
10 DATABASE_PORT=5432
11 DATABASE_NAME=nlppdb
12 DATABASE_USERNAME=postgres
13 DATABASE_PASSWORD=postgres
14 DATABASE_SSL=false
15 JWT_SECRET=5MByMvfNLYVWLFxNTTFedg==
16
```

### 3.2.1 CONFIGURACIÓN INICIAL DE SERVICIO

Dentro de estos pasos se configurará el servicio de gestión de contenido para poder ser accedido, y se creará un super usuario que será utilizado como administrador de este, **junto con otro usuario** para el manejo del flujo de información en la página web.

1. Al iniciar por primera vez y dirigirse al enlace que nos entrega la [consola](#), el programa nos pedirá registrar un **super usuario** para la plataforma de gestión de contenido.



The image shows the Strapi registration form. It has a dark blue background with a light blue Strapi logo at the top. The text 'Welcome to Strapi!' is centered, followed by a note: 'Credentials are only used to authenticate in Strapi. All saved data will be stored in your database.' Below this are input fields for 'First name\*', 'Last name', 'Email\*', 'Password\*', and 'Confirm Password\*'. The Password field has a hint: 'Must be at least 8 characters, 1 uppercase, 1 lowercase & 1 number'. There is a checkbox for 'Keep me updated about new features & upcoming improvements (by doing this you accept the [terms](#) and the [privacy policy](#)).' and a 'Let's start' button at the bottom.

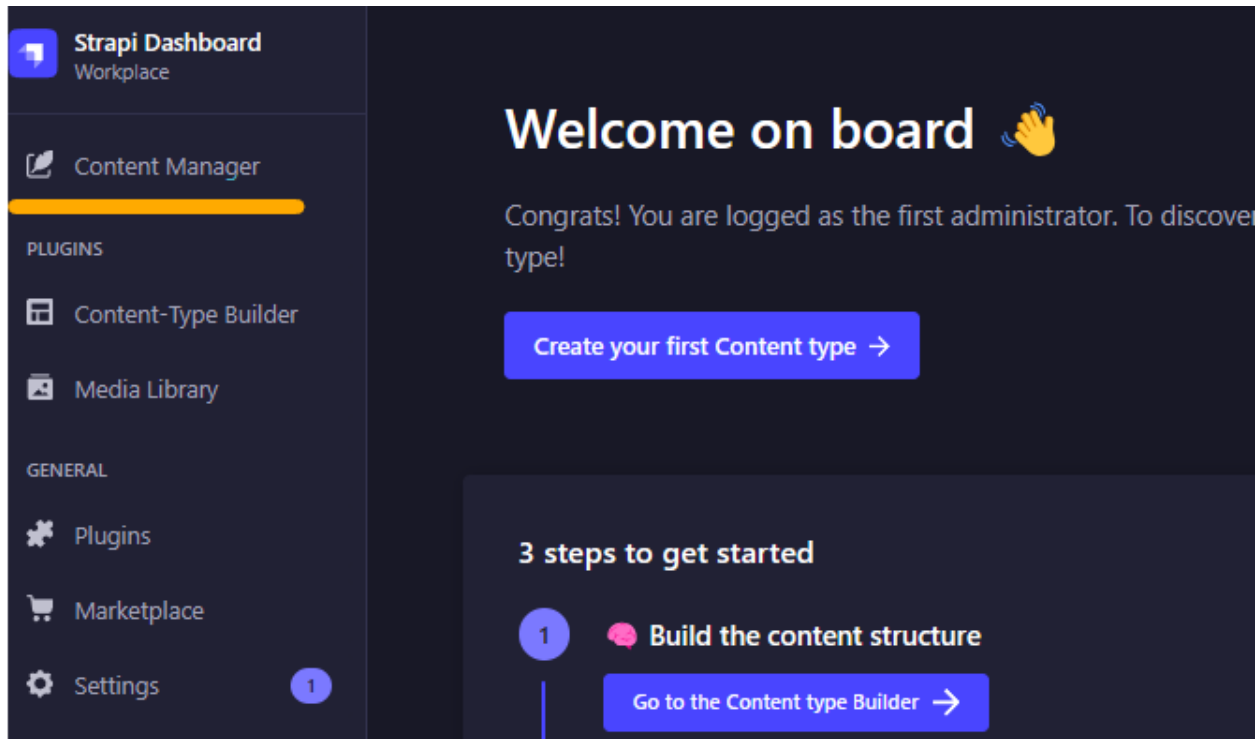
Un ejemplo de super usuario para la plataforma Strapi sería el siguiente:

First name : user , Last name : user

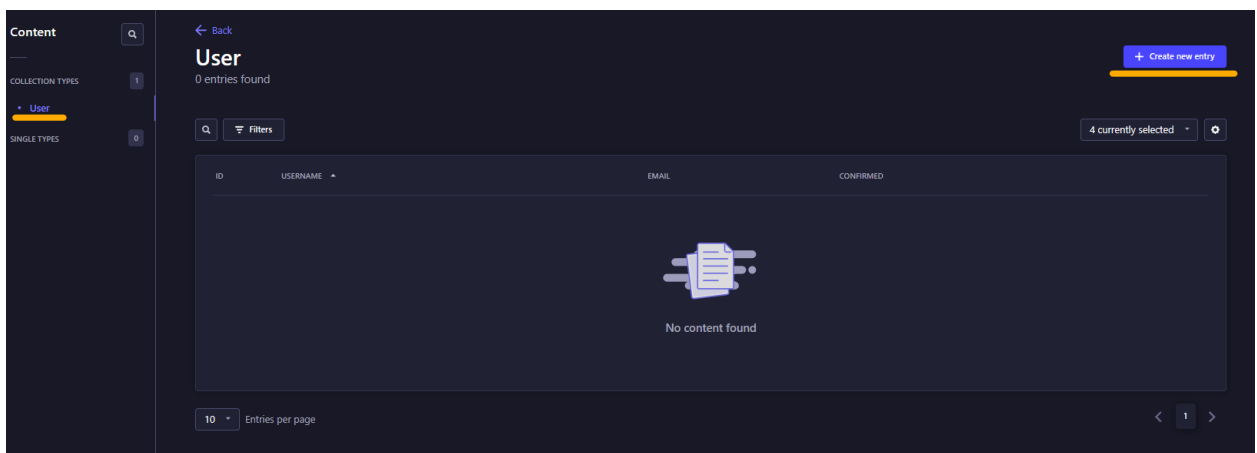
Email: [correo@prueba.com](mailto:correo@prueba.com)

Password: E8URA2Sw

2. Luego de registrarse el programa nos dejara ingresar a la plataforma en ella deberemos dirigirnos a la sección **Content Manager** para crear un usuario.



3. Seleccionar en **COLLECTION TYPES -> User** y luego **Create new Entry**



4. Llenar las casillas de username, email y password correspondientes, un ejemplo de esto sería lo siguiente. **Recomendación: Definir un usuario genérico ya que este será el ocupado para ingresar a la aplicación.**

The screenshot shows a user registration form with the following fields and values:

- username\***: demo (min. 3 characters)
- email\***: demo@demo.cl (min. 6 characters)
- password**: ..... (min. 6 characters, with a toggle icon)
- confirmed**: TRUE (selected, highlighted with a yellow bar)
- blocked**: FALSE (selected)
- role (I)**: Add relation (dropdown menu)
- role (I)**: 1 (selected, with a close icon)

**Importante: Marcar la casilla confirmed como TRUE, blocked como FALSE y role como 1 o Authenticated**

Un ejemplo de usuario que se utilizara en la aplicación seria el siguiente:

username : demo , email : [demo@demo.cl](mailto:demo@demo.cl)

password: demodemo, confirmed: TRUE



2. Ingresar en la plataforma y cargar los datos etiquetados que serán utilizados para entrenar el modelo [Introducción a Vertex AI](#) | [Vertex AI](#) | [Google Cloud](#)
3. Luego de entrenar el modelo para poder utilizarse en el microservicio, se debe crear un extremo o **endpoint** para el modelo, la siguiente figura muestra el endpoint creado

**Predicción en línea**

**EXTREMOS** GRUPOS DE RECURSOS DE IMPLEMENTACIÓN

Los extremos son modelos de aprendizaje automático disponibles para las solicitudes de predicción en línea. Son útiles para las predicciones oportunas de muchos usuarios (por ejemplo, en respuesta a una solicitud de aplicación). También puedes solicitar predicciones por lotes si no necesitas resultados inmediatos.

Para crear un extremo, necesitas al menos un modelo de aprendizaje automático. [Más información](#)

Región: us-central1 (Iowa)

**Extremos** CREAR

Filtro: Ingresar un nombre de propiedad

Nombre	ID	Estado	Modelos	Grupo de recursos de implementación	Región	Monitoring	Alertas más recientes
<a href="#">Endpoint1</a>	3214087992268292096	✓ Listo	0	—	us-central1	Inhabilitada	—
<a href="#">Endpoint3</a>	4966340091036303360	✓ Listo	0	—	us-central1	Inhabilitada	—
<a href="#">Endpoint2</a>	8721990336542408704	✓ Listo	0	—	us-central1	Inhabilitada	—

**Sample Request**

REST PYTHON

Ahora puedes ejecutar consultas con la interfaz de línea de comandos (CLI).

1. Asegúrate de tener el [SDK de Google Cloud](#) instalado.
2. Ejecuta el siguiente comando para autenticarte con tu Cuenta de Google.
 

```
$ gcloud auth application-default login
```
3. Crea un objeto JSON para conservar los datos.
 

```
{
  "instances": [
    { "instance_key_1": "value", ... }, ...
  ],
  "parameters": { "parameter_key_1": "value", ... }, ...
}
```
4. Crea variables de entorno para conservar los ID de los extremos y los proyectos, y el objeto JSON.
 

```
$ ENDPOINT_ID="3214087992268292096"
PROJECT_ID="985882468314"
INPUT_DATA_FILE="INPUT-JSON"
```
5. Ejecuta la solicitud.
 

```
$ curl \
  -X POST \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
  -H "Content-Type: application/json" \
  https://us-central1-aiplatform.googleapis.com/v1/projects/$PROJECT_ID/locations/$LOCATION/publishers/google/models/$MODEL_ID/instances \
  -d @$INPUT_DATA_FILE
```

**DONE**

4. Luego se debe integrar a los archivos de configuración de entorno “.env” del directorio del proyecto del microservicio API-NLP quedado de la siguiente manera

The screenshot shows the Visual Studio Code editor with the following components:

- Explorer Panel:** Displays the project structure for 'PLATAFORMA NLP'. The 'NLP-API' folder is expanded, showing files like .env, .eslintrc.json, .gitignore, .prettierrc, credenciales.json, LICENSE.md, nodemon.json, package-lock.json, package.json, README.md, and LICENSE.
- Editor Panel:** Shows the content of the '.env' file for the 'NLP-API' project. The file contains the following configuration:
 

```

1 # Local development environment
2 PORT=5000
3
4 # Google Cloud Platform
5 #Vertex AI Config
6 GOOGLE_APPLICATION_CREDENTIALS = "./credenciales.json"
7 LOCATION = "us-central1"
8 PROJECT_ID = "905882468314"
9
10
11 ENDPOINT1_ID = "3214087992268292096"
12 ENDPOINT2_ID = "8721990336542408704"
            
```
- Terminal Panel:** Shows the output of the server. It indicates the server is ready at 'http://localhost:5000' and displays two GET requests:
 

```

GET / 200 9.727 ms - 26
GET /favicon.ico 404 1.580 ms - 52
            
```
- Taskbar:** Shows the system tray with various icons, including the taskbar, system clock, and network status.

### 3.5 Microservicio de procesamiento NLP

Finalmente dentro del directorio del proyecto se encuentra la carpeta llamada **NLP-API**, abrir una carpeta dentro de este y ejecutar el comando `$npm run dev`. Con este podrá iniciar el servicio accediendo a las funcionalidades, para verificar el correcto inicio se deberá ver lo siguiente

```
[nodemon] ignoring: .\.git/**/* node_modules/**/node_modules
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,json
[nodemon] starting `node --harmony ./src/index.js`
[nodemon] spawning
[nodemon] child pid: 8500
[nodemon] watching 55 files

🚀 Server ready at: http://localhost:5000
```