18 de diciembre de 2024

**Tabla de contenido**

[Requerimientos mínimos 4](#_Toc185410332)

[Hardware 4](#_Toc185410333)

[Software 4](#_Toc185410334)

[Preparativos 4](#_Toc185410335)

[Instalación de software 4](#_Toc185410336)

[Actualización de SO 4](#_Toc185410337)

[PostgreSQL 17 5](#_Toc185410338)

[MongoDB 7 9](#_Toc185410339)

[NodeJS 21 11](#_Toc185410340)

[PM2 13](#_Toc185410341)

[Git 13](#_Toc185410342)

[Configuración e implementación de la herramienta 14](#_Toc185410343)

[Generación de la base de datos 14](#_Toc185410344)

[Código de la herramienta 15](#_Toc185410345)

[Configuración de las variables de conexión a la base de datos. 17](#_Toc185410346)

[Creación de la estructura de la base de datos y del esquema public. 21](#_Toc185410347)

[Configuración del usuario propietario para el esquema public. 22](#_Toc185410348)

[Creación del esquema dashboard 23](#_Toc185410349)

[Configuración del usuario propietario para el esquema dashboard 24](#_Toc185410350)

[Validación de la estructura de la base de datos. 25](#_Toc185410351)

[Validación de la conexión a la base de datos 26](#_Toc185410352)

[Asignación de variables para el token de mapa (MAPTOKEN) 27](#_Toc185410353)

[Configuración de IP 31](#_Toc185410354)

[Instalación de Wkhtmltopdf 32](#_Toc185410355)

[Módulo de Infraestructura 32](#_Toc185410356)

[Creación de vistas para el módulo de Infraestructura Abierta 34](#_Toc185410357)

[Creación del usuario del aplicativo 35](#_Toc185410358)

[Iniciar los servicios de los módulos 36](#_Toc185410359)

[Configuración de Google Analytics 39](#_Toc185410360)

[Personalización de la herramienta 40](#_Toc185410361)

[Cambiar logo de la institución 40](#_Toc185410362)

[Actualizar las siglas de la institución 42](#_Toc185410363)

[Actualizar el pie de página del módulo de dashboard 43](#_Toc185410364)

[Cambiar la ruta de navegación 45](#_Toc185410365)

[Anexo A: Solución a errores comunes 47](#_Toc185410366)

[Error en la creación de una nueva contratación 47](#_Toc185410367)

[Anexo B: Definiciones 47](#_Toc185410368)

[Ubuntu 47](#_Toc185410369)

[PostgreSQL 47](#_Toc185410370)

[MongoDB 47](#_Toc185410371)

[NodeJS 48](#_Toc185410372)

[NGINX 48](#_Toc185410373)

[PM2 48](#_Toc185410374)

[Git 48](#_Toc185410375)

[VIM 48](#_Toc185410376)

[Mapbox 48](#_Toc185410377)

[Sequelize-CLI 48](#_Toc185410378)

[Google Analytics 49](#_Toc185410379)

[Wkhtmltopdf 49](#_Toc185410380)

# **Requerimientos mínimos**

## Hardware

* **Memoria RAM:** 4 GB.
* **Velocidad de procesador:** 2 GHz.
* **Almacenamiento:** 250 GB.

## Software

* **Sistema operativo:** Ubuntu 22.04.3 LTS.
* **Base de datos relacional:** PostgreSQL 17.
* **Base de datos de documentos:** MongoDB 7.
* **Lenguaje de programación:** NodeJS 21.

# **Preparativos**

Se sugiere que, previo a la ejecución de cualquier comando descrito en este manual, se realice una lectura completa del mismo para obtener un conocimiento integral de su estructura, así como de las descripciones y comentarios que contiene.

Siguiendo con los preparativos, para avanzar con la instalación del software base y las configuraciones necesarias, es necesario acceder al servidor como usuario root. Para lograrlo, ejecutamos el siguiente comando. Al hacerlo, se nos solicitará la contraseña de usuario root, la proporcionamos y así estaremos listos para dar inicio al proceso.

 En caso de desconocer la contraseña del usuario root, consulte la sección inicial del Anexo.

su

# **Instalación de software**

## Actualización de SO

Realizamos la actualización de los paquetes del sistema operativo.

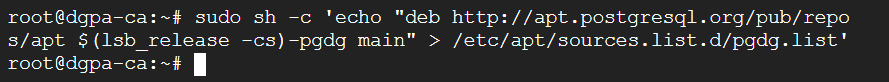
apt update

apt upgrade -y

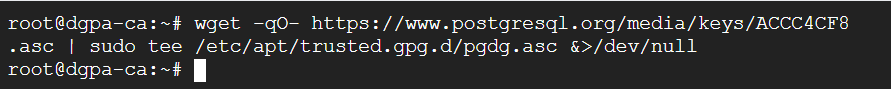
## PostgreSQL 17

Debido a la ausencia del repositorio de paquetes de PostgreSQL 17 en el repositorio predeterminado, habilitaremos dicho repositorio desde su fuente oficial ejecutando el siguiente comando.

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'



wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee /etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null



Procedemos a actualizar los paquetes con el objetivo de obtener las versiones más recientes.

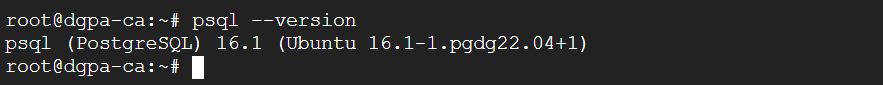
apt update

Realizamos la instalación del paquete principal de PostgreSQL, junto con un paquete que incorpora contribuciones adicionales.

apt install postgresql postgresql-contrib -y

Verificamos que la versión instalada sea la adecuada.

psql --version



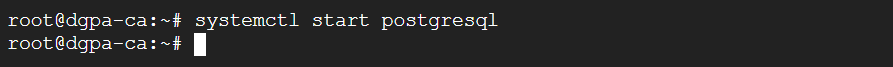
Una vez completada la instalación, activamos el inicio automático del servicio de PostgreSQL mediante el siguiente comando.

systemctl enable postgresql



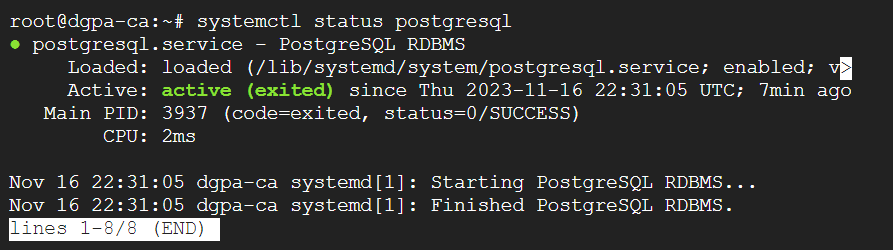
Procedemos a iniciar el servicio de PostgreSQL.

systemctl start postgresql



Comprobamos que el servicio esté activo.

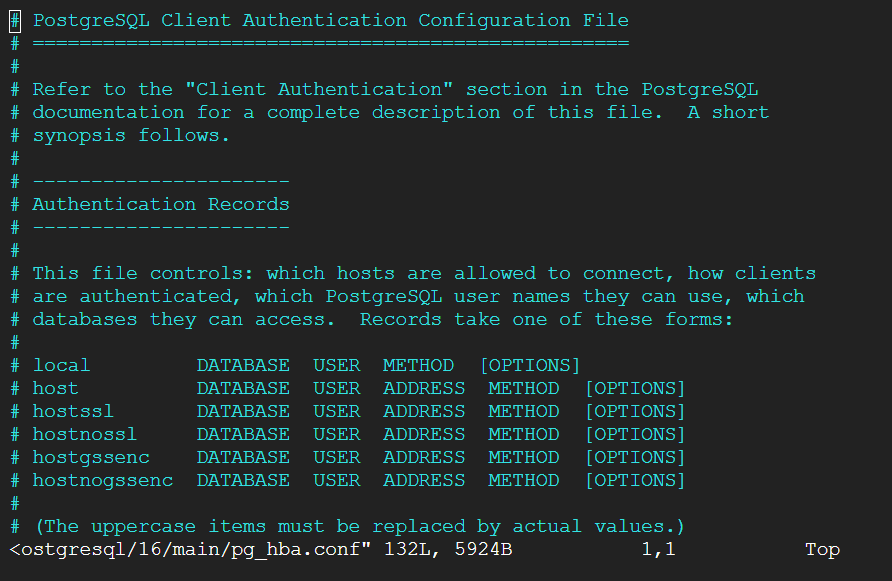
systemctl status postgresql



Para salir, presionamos **Ctrl+C**.

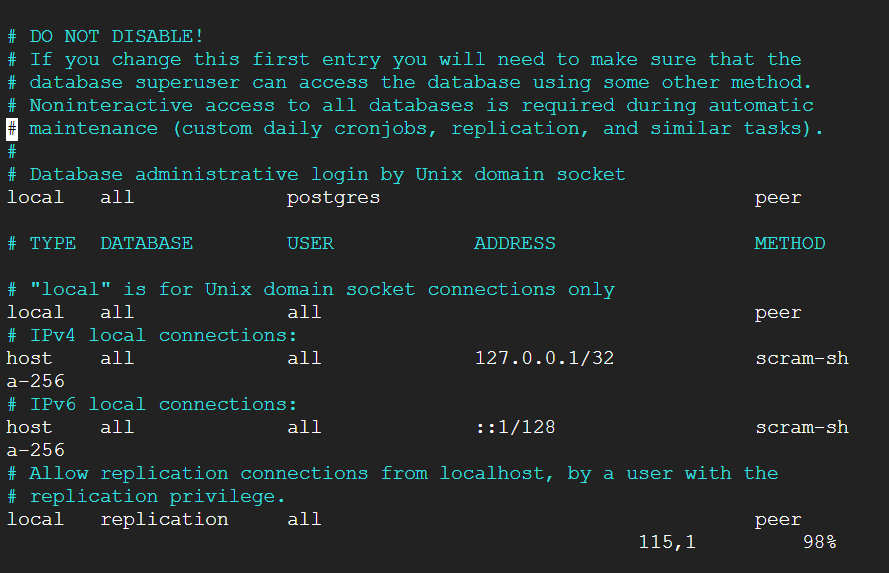
Ahora procederemos a editar el archivo **pg\_hba.conf**, ubicado en la siguiente ruta: **/etc/postgresql/17/main/**. Para llevar a cabo esta tarea, ejecutaremos el siguiente comando.

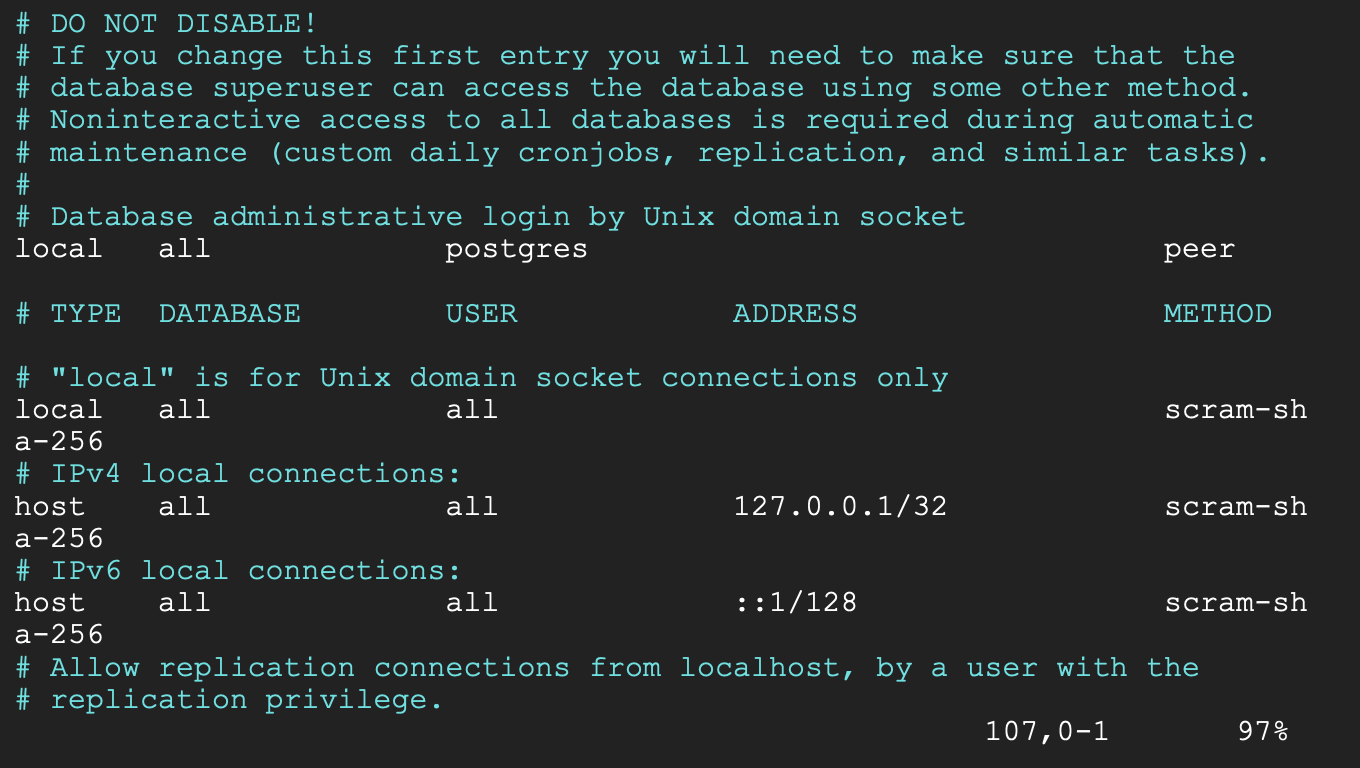
 La ruta dependerá de la versión de PostgreSQL instalada. Esta se verifica al momento de ejecutar el comando ***psql --version***, ajustando el número de la ruta según tu versión instalada.



vi /etc/postgresql/17/main/pg\_hba.conf

Descenderemos dentro del archivo hasta encontrar la configuración siguiente. La modificaremos a "scram-sha-256" para los tipos de conexión **local** e **IPv4 local**. Para esto, presionamos la tecla **i**, lo que nos permitirá editar el archivo.





Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

Reiniciamos el servicio de PostgreSQL y posteriormente, verificamos el estado del servicio.

systemctl restart postgresql

systemctl status postgresql



Para salir, presionamos **Ctrl+C**.

## MongoDB 7

Procedemos a la instalación de **GnuPG** y **Curl**. Para ello, ejecutaremos el siguiente comando.

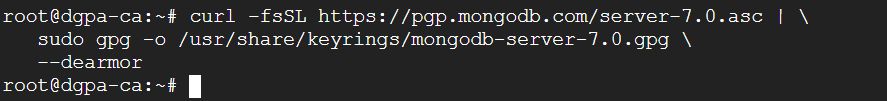
apt-get install gnupg curl

Ahora procedemos a descargar una clave pública de MongoDB, la importamos en el almacén de claves del sistema y la guardamos para su posterior uso.

curl -fsSL https://pgp.mongodb.com/server-7.0.asc | \

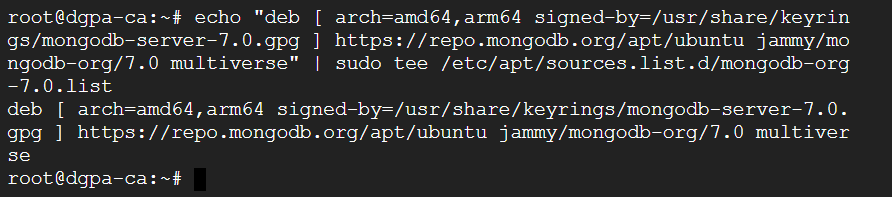
sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg \

--dearmor



Creamos una lista de fuentes denominada **mongodb-org-7.0.list** en el directorio **/etc/apt/sources.list.d/**, con el fin de permitir al sistema obtener los paquetes de software de MongoDB desde el repositorio especificado.

echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list



Recargamos las bases de datos locales de los paquetes.

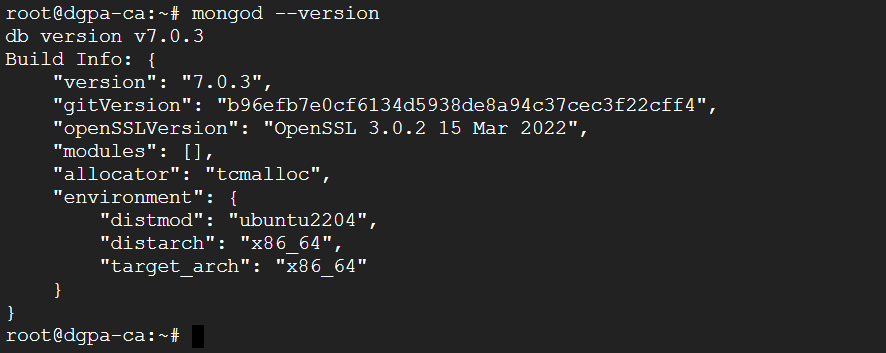
apt-get update

Procedemos a instalar MongoDB en su versión 7.

apt-get install -y mongodb-org

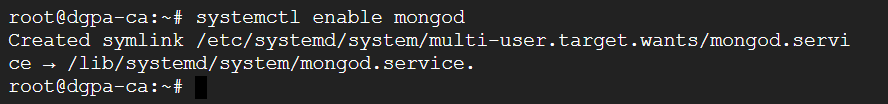
Verificamos que la versión instalada sea la adecuada.

mongod --version



Una vez completada la instalación, activamos el inicio automático del servicio de MongoDB mediante el siguiente comando.

systemctl enable mongod



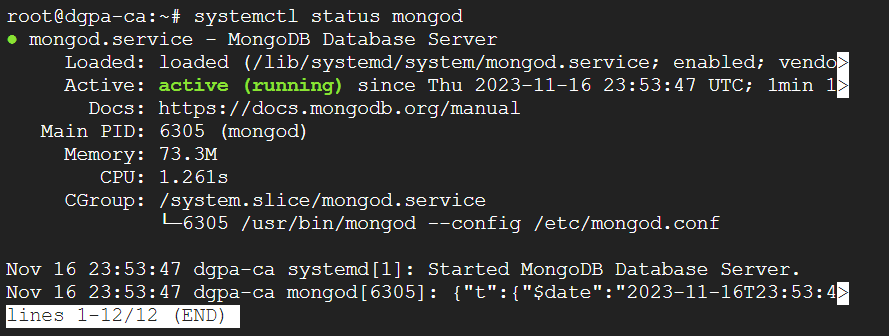
Procedemos a iniciar el servicio de MongoDB.

systemctl start mongod



Comprobamos que el servicio esté activo.

systemctl status mongod



Para salir, presionamos **Ctrl+C**.

## NodeJS 21

Llevamos a cabo la instalación de ca-certificates.

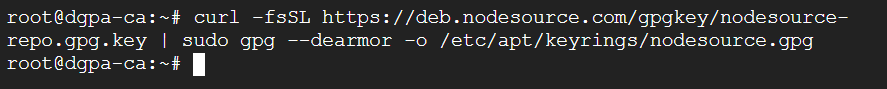
apt-get install -y ca-certificates

Creamos un directorio denominado **keyrings** en la siguiente ruta: **/etc/apt/**.

mkdir -p /etc/apt/keyrings

Por último, descargamos la clave GPG del repositorio de **Nodesource**, la convertimos en un formato adecuado y la guardamos en el sistema en el archivo llamado **nodesource.gpg** que se encuentra en la siguiente ruta: **/etc/apt/keyrings/**.

curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg



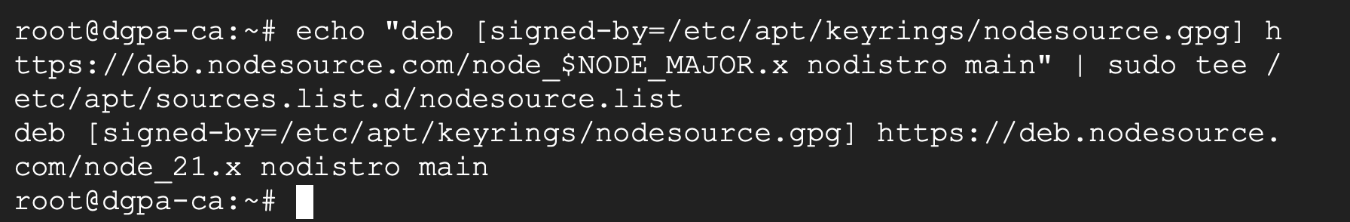
Ahora, procedemos a definir la variable de entorno **NODE\_MAJOR** y le asignamos el valor de 21, que corresponde a la versión de Node que instalaremos.

NODE\_MAJOR=21



Procedemos a crear una lista de fuentes llamada **nodesource.list** en el directorio **/etc/apt/sources.list.d/**, donde almacenaremos la configuración del repositorio de NodeSource.

echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node\_$NODE\_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list



Llevamos a cabo la actualización de las bases de datos locales de los paquetes.

apt-get update

Realizamos la instalación de NodeJS.

apt-get install nodejs -y

Verificamos que la versión instalada sea la adecuada.

node -v



Tras la instalación de **NodeJS**, verificamos la versión de **npm** instalada.

npm -v



## PM2

Realizamos la instalación de PM2.

npm install pm2@latest -g

## Git

Realizamos la instalación de Git.

apt install git

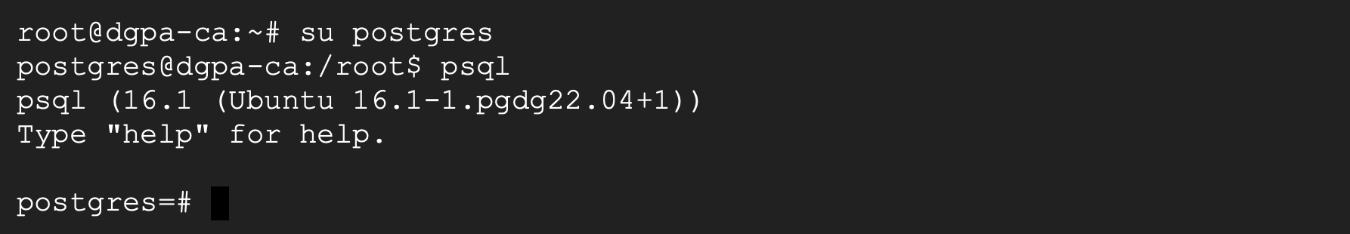
# **Configuración e implementación de la herramienta**

## Generación de la base de datos

Recordando la importancia de estar en el usuario root, ingresamos a PostgreSQL.

su postgres

psql



Dentro de PostgreSQL, procedemos a crear la base de datos denominada **edca**.

create database edca;

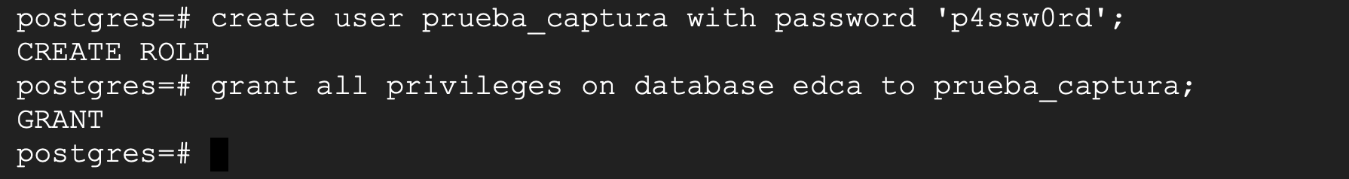


Ahora, generamos el usuario que será designado como propietario del **esquema public** del **módulo de captura** y le otorgamos los correspondientes permisos sobre la base de datos.

 Los nombres de usuario y contraseñas utilizados en este manual son ejemplos, se desaconseja su utilización en una instalación real.

create user prueba\_captura with password 'p4ssw0rd';

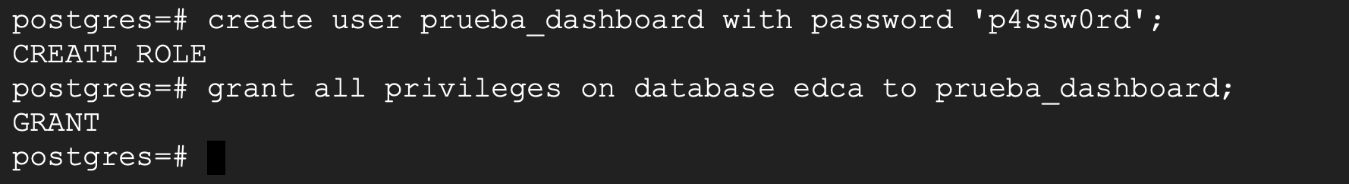
grant all privileges on database edca to prueba\_captura;



Procedemos a crear el usuario que actuará como propietario del **esquema dashboard** en la base de datos. De manera similar, le otorgamos todos los permisos sobre la base de datos.

create user prueba\_dashboard with password 'p4ssw0rd';

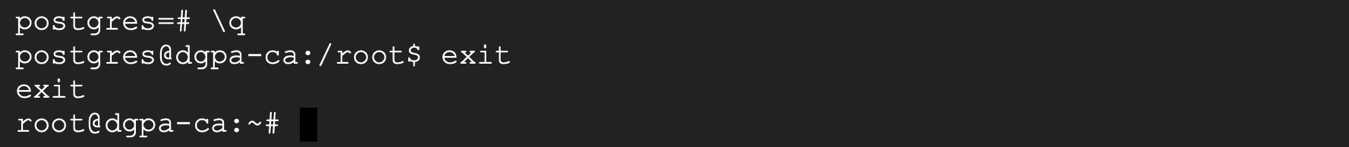
grant all privileges on database edca to prueba\_dashboard;



Finalmente, cerramos la sesión de PostgreSQL.

\q

exit



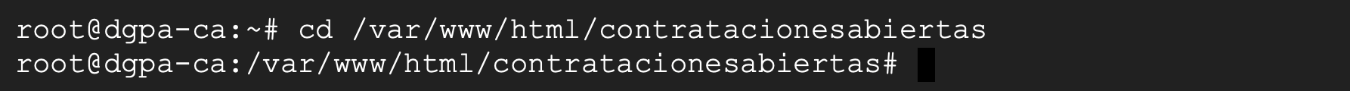
Código de la herramienta

Antes de descargar el código de la herramienta, procedemos a crear los directorios en los que estará alojado. Para ello, estableceremos la siguiente ruta: **/var/www/html/contratacionesabiertas**.

mkdir -p /var/www/html/contratacionesabiertas

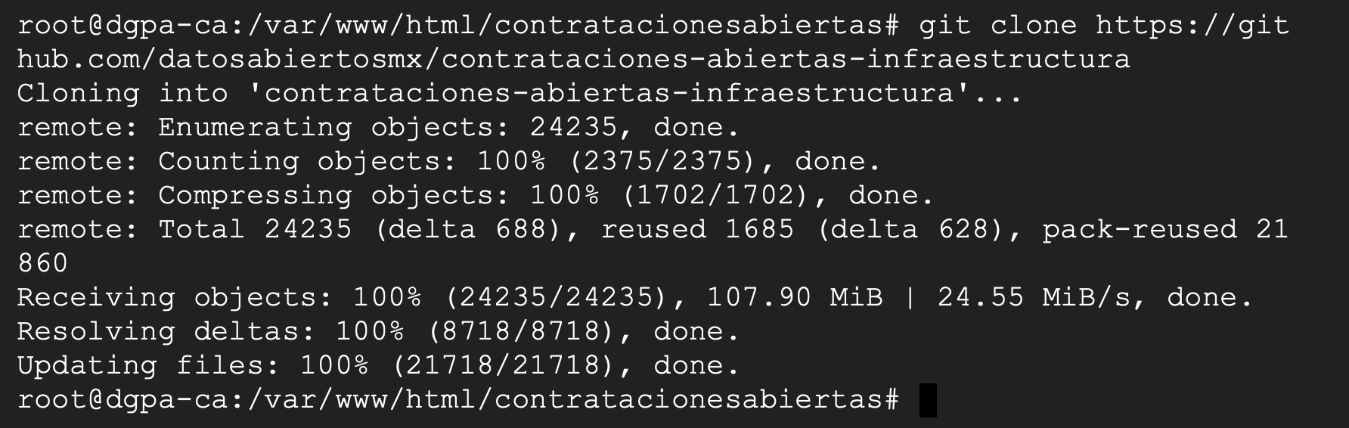
Accedemos a la ruta.

cd /var/www/html/contratacionesabiertas



Descargamos el código desde el repositorio oficial.

git clone https://github.com/datosabiertosmx/contratacionesabiertas-3.0.git



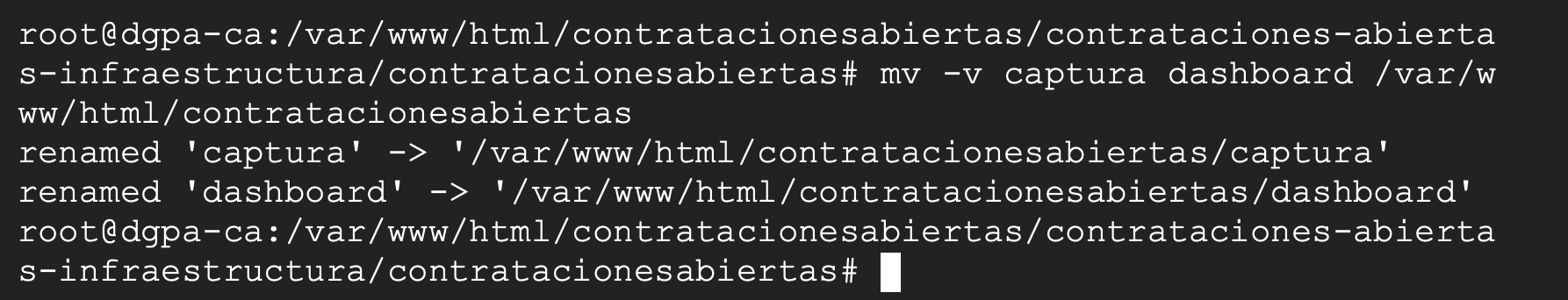
Ingresamos a la ruta que contiene los directorios de **captura** y **dashboard**.

cd contratacionesabiertas-3.0/contratacionesabiertas



Trasladamos estos directorios a la carpeta principal de **contratacionesabiertas**.

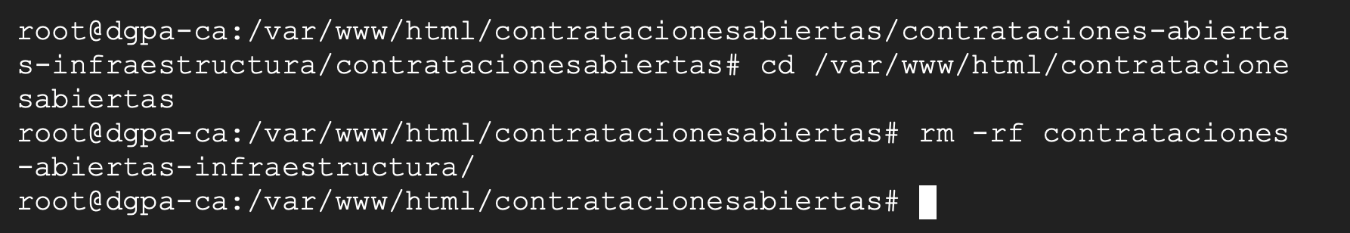
mv -v captura dashboard /var/www/html/contratacionesabiertas



Accedemos al directorio donde clonamos el repositorio y eliminamos los archivos que ya no necesitamos.

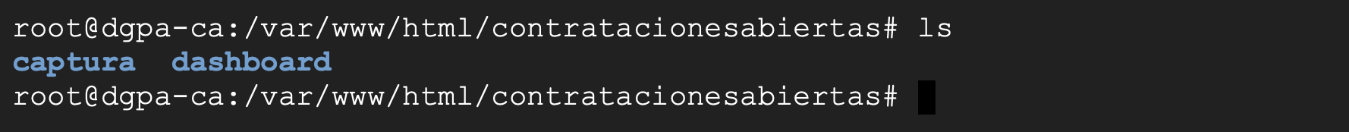
cd /var/www/html/contratacionesabiertas

rm -rf contratacionesabiertas-3.0



Verificamos que ahora solo se encuentren presentes los directorios de **captura** y **dashboard**.

ls

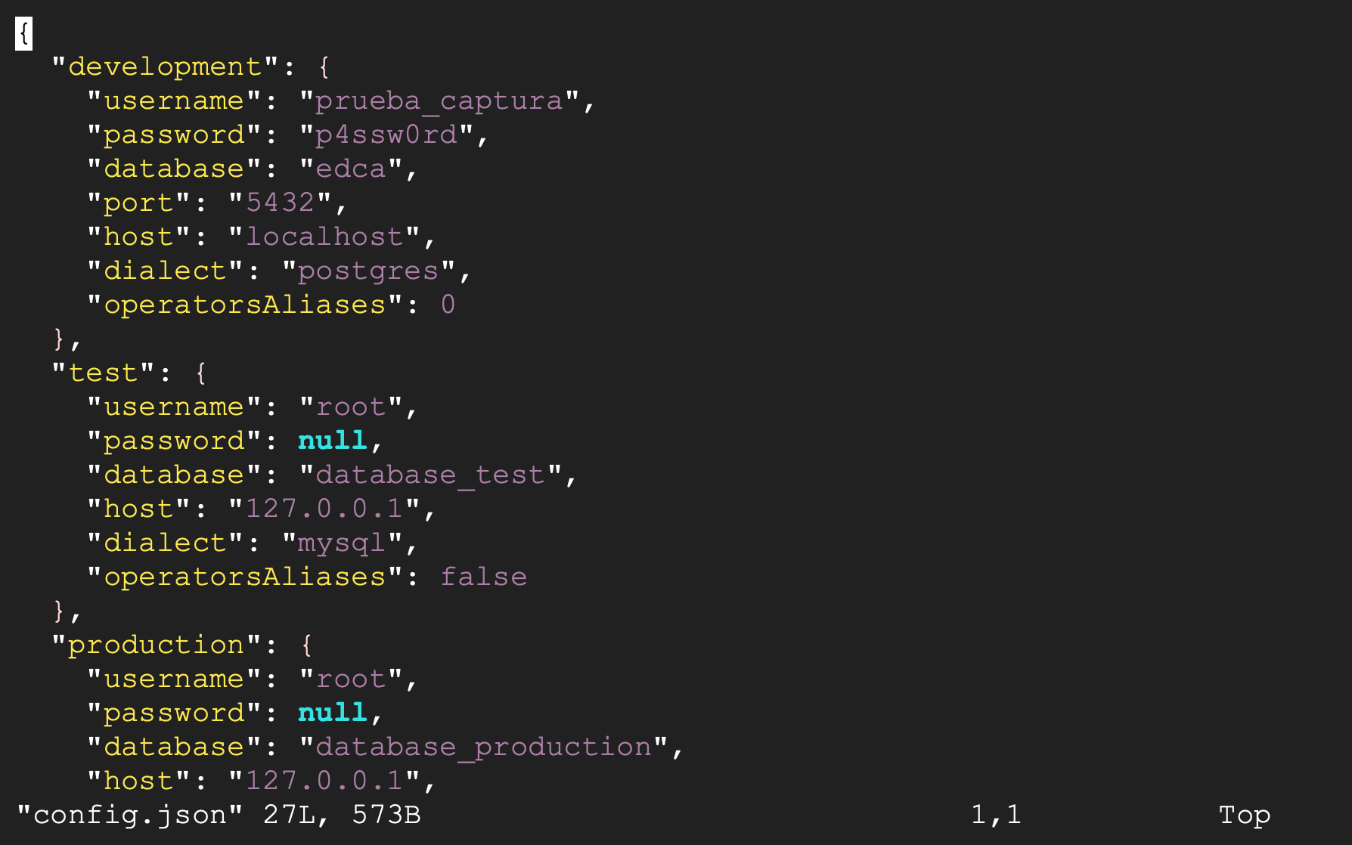


## Configuración de las variables de conexión a la base de datos.

Nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/captura/config** y editamos el archivo **config.json**. Modificamos las variables de **username**, **password** y **database** correspondientes a las creadas en el **esquema public** del módulo de captura.

cd /var/www/html/contratacionesabiertas/captura/config

vi config.json

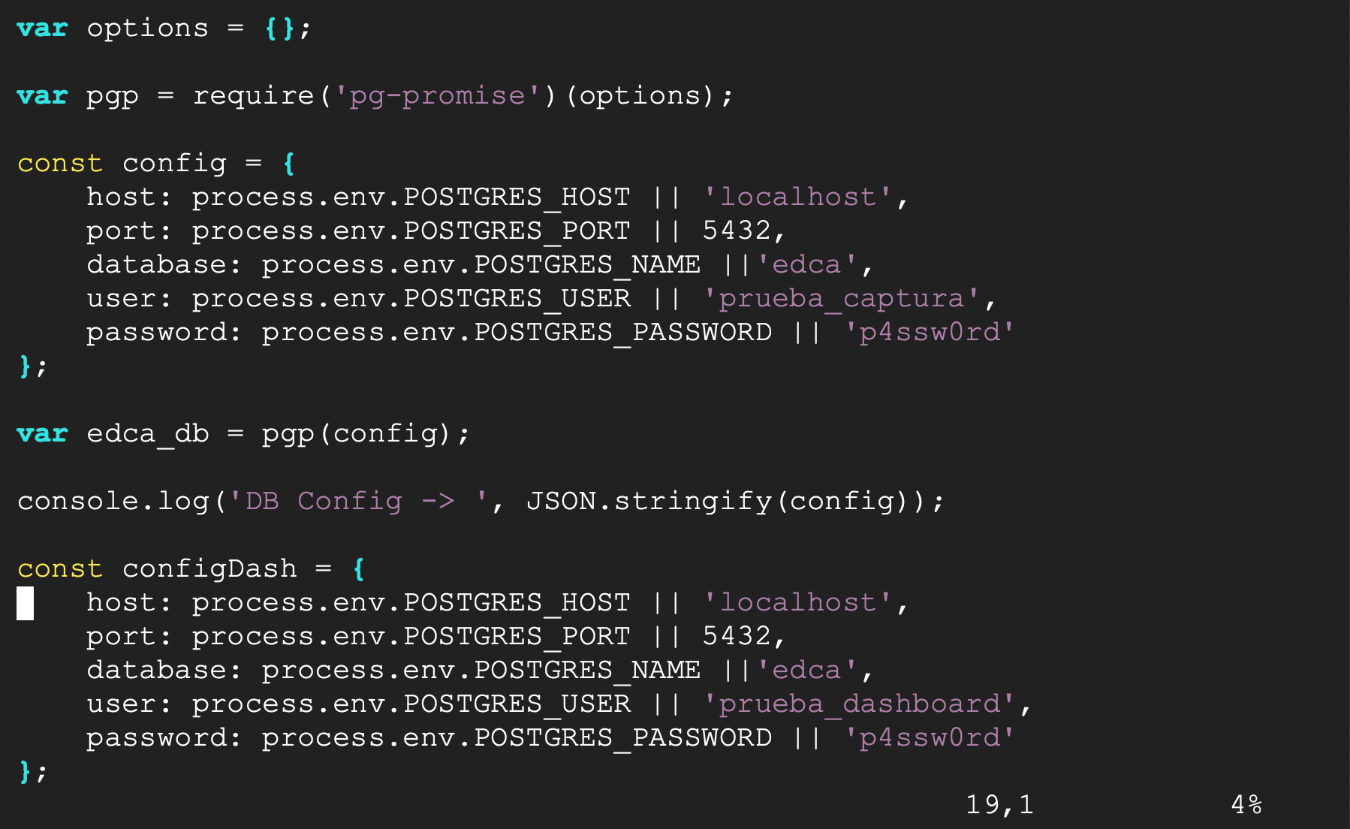


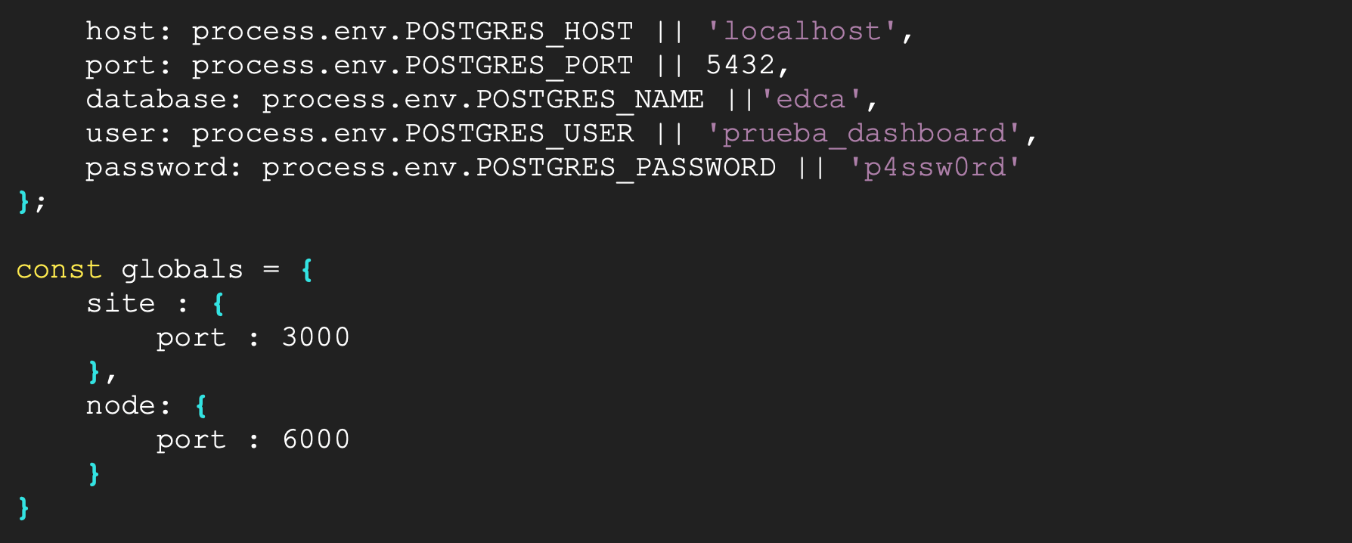
Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

Nos dirigimos al directorio de **captura** y editamos el archivo llamado **db\_conf.js**. Modificamos los valores de **HOST, PORT, DATABASE, USER** y **PASSWORD** de acuerdo a la configuración propia. Utilizamos los datos de **usuario** y **contraseña** establecidos para el **módulo de captura** y de **dashboard**. Asimismo, configuramos las **constantes globales** **(const globals)** con los puertos asignados para la instalación.

cd /var/www/html/contratacionesabiertas/captura

vi db\_conf.js



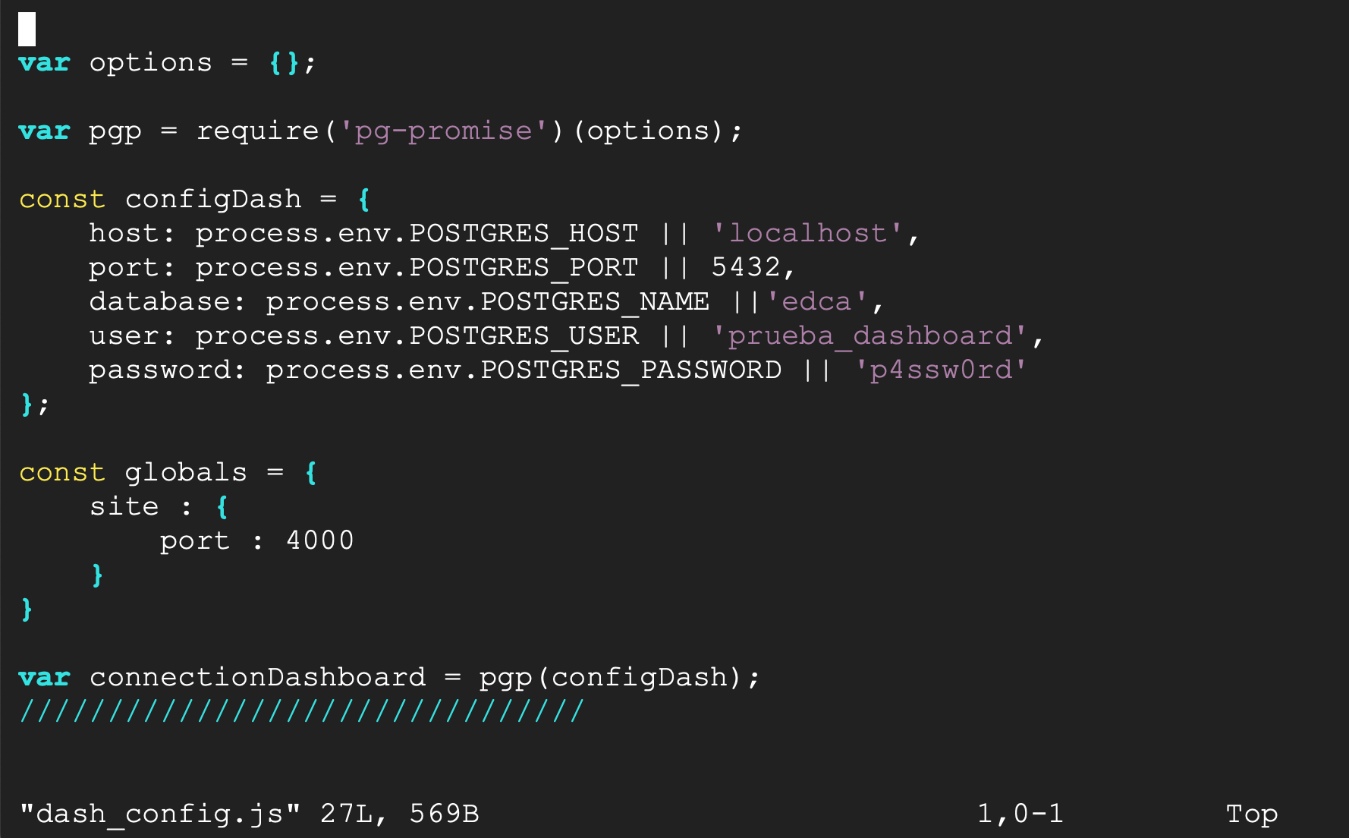


Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

Accedemos al directorio de **dashboard** y editamos el archivo llamado **dash\_config.js.** Modificamos los valores de **HOST, PORT, DATABASE, USER** y **PASSWORD** de acuerdo a la configuración propia. Utilizamos los datos de **usuario** y **contraseña** establecidos para el **módulo de dashboard**. Del mismo modo, configuramos las **constantes globales (const globals)** con los puertos asignados para la instalación.

cd /var/www/html/contratacionesabiertas/dashboard

vi dash\_config.js



Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

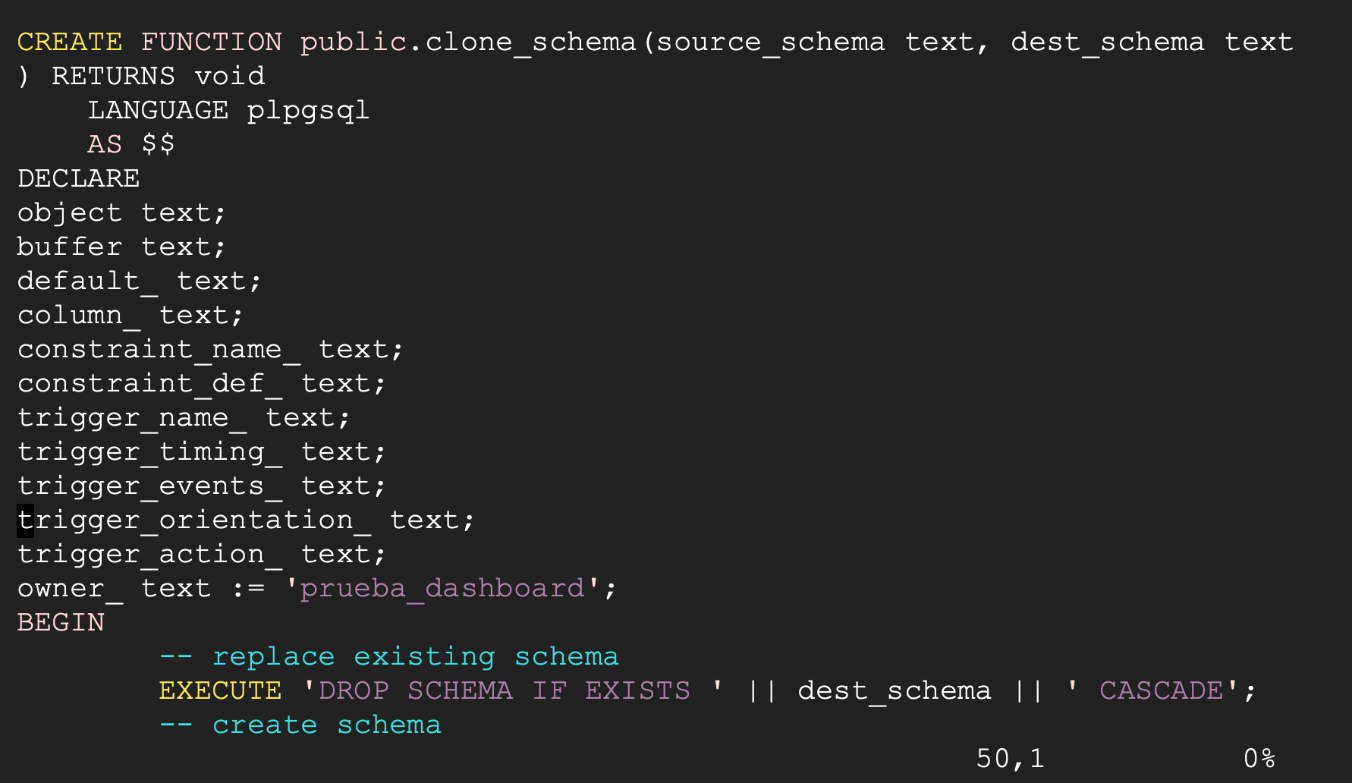
## Creación de la estructura de la base de datos y del esquema public.

Ingresamos al directorio de **sql** y editamos el archivo denominado **edca.sql**, el cual contiene la estructura de la base de datos.

cd /var/www/html/contratacionesabiertas/captura/sql

vi edca.sql

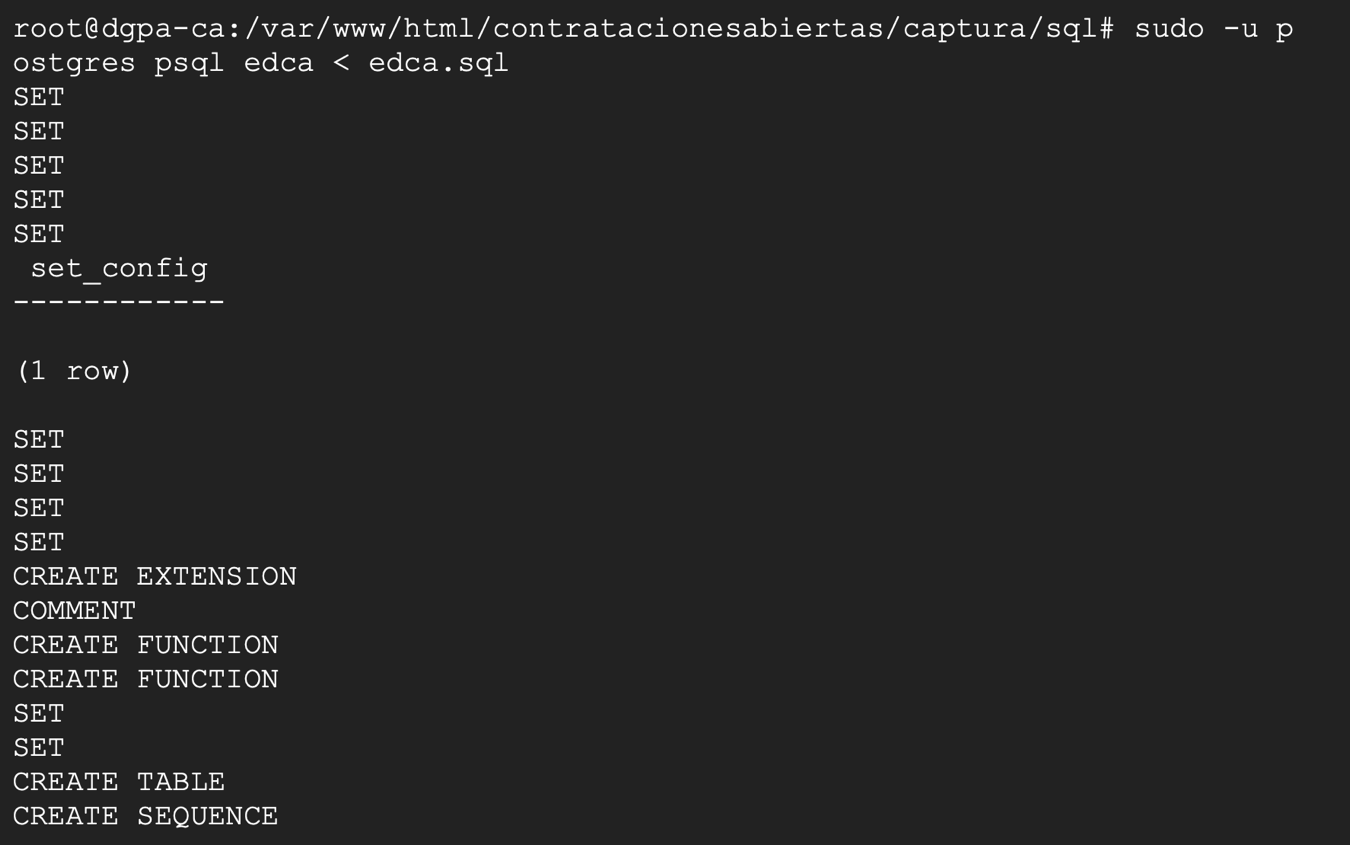
Realizamos la modificación correspondiente al **nombre de usuario** del **módulo de dashboard** al **inicio** del archivo y, al **final**, actualizamos el **nombre de usuario** del **módulo de captura**.





Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

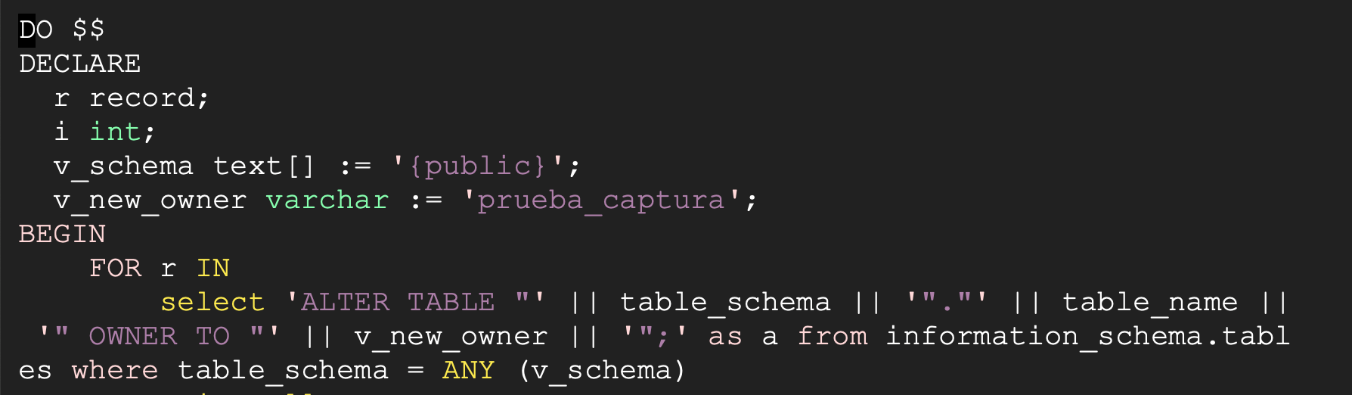
Procedemos a ejecutar el archivo modificado.



sudo -u postgres psql edca < edca.sql

## Configuración del usuario propietario para el esquema public.

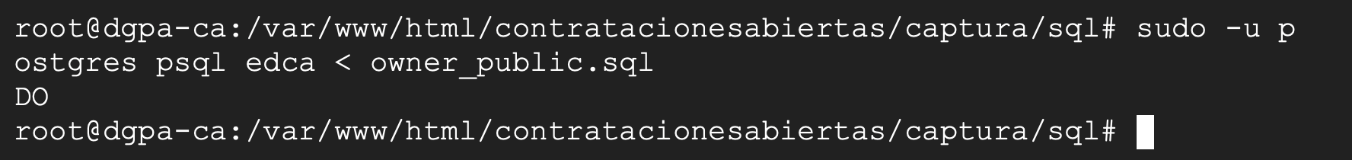
Dentro de la ruta **/var/www/html/contratacionesabiertas/captura/sql**, editaremos el archivo llamado **owner\_public.sql** y en él especificaremos el usuario que creamos para el **módulo de captura.**



vi owner\_public.sql

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

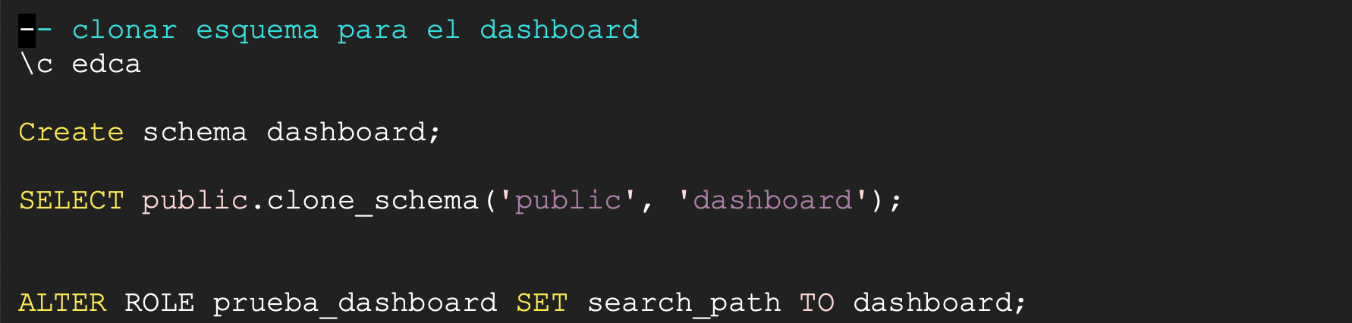
Ahora procedemos a ejecutar el script que hemos modificado.



sudo -u postgres psql edca < owner\_public.sql

## Creación del esquema dashboard

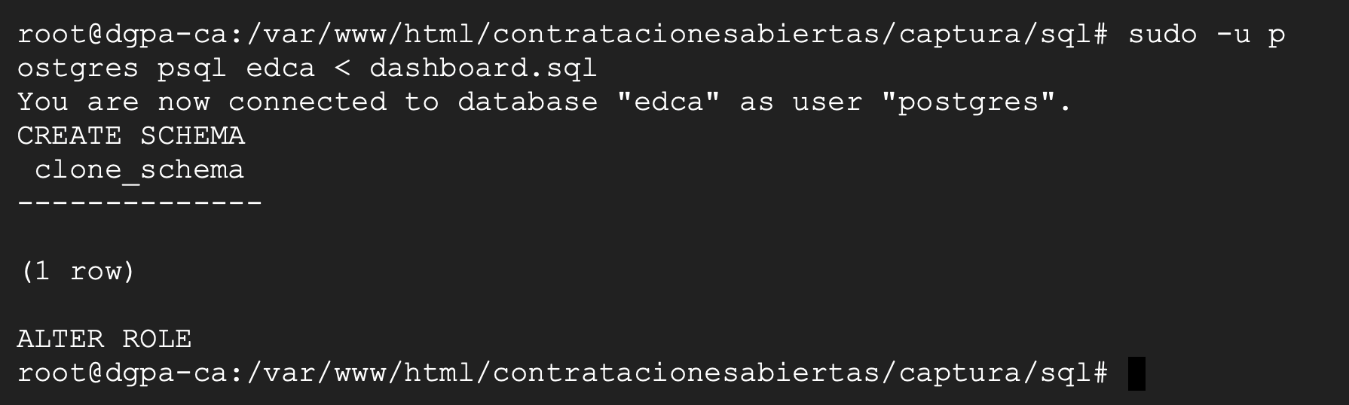
De manera similar, desde **/var/www/html/contratacionesabiertas/captura/sql**, editaremos el archivo denominado **dashboard.sql**, en el cual especificaremos el usuario que creamos para el **módulo de dashboard**.



vi dashboard.sql

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

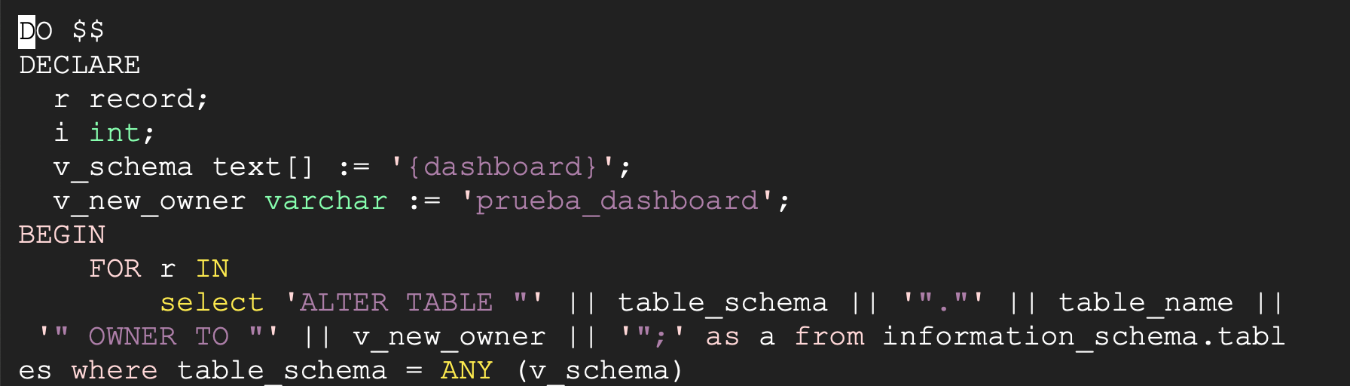
Ahora procedemos a ejecutar el script que hemos modificado.



sudo -u postgres psql edca < dashboard.sql

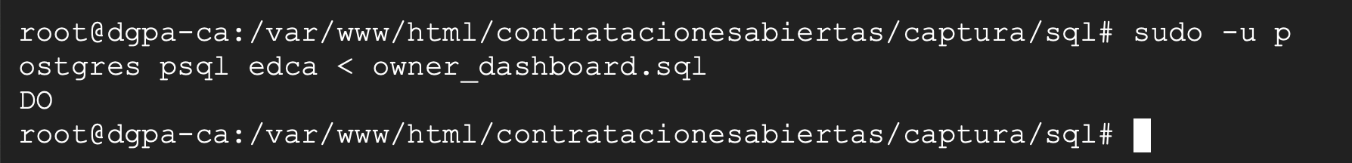
## Configuración del usuario propietario para el esquema dashboard

Continuando en **/var/www/html/contratacionesabiertas/captura/sql**, editaremos el archivo denominado **owner\_dashboard.sql**, y en él especificaremos el usuario que creamos para el **módulo de dashboard**.

 Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

vi owner\_dashboard.sql

Ahora procedemos a ejecutar el script que hemos modificado.



sudo -u postgres psql edca < owner\_dashboard.sql

## Validación de la estructura de la base de datos.

Para confirmar que el **esquema public** de la base de datos se haya creado correctamente, ingresamos a PostgreSQL y verificamos que contenga los elementos que se muestran a continuación.

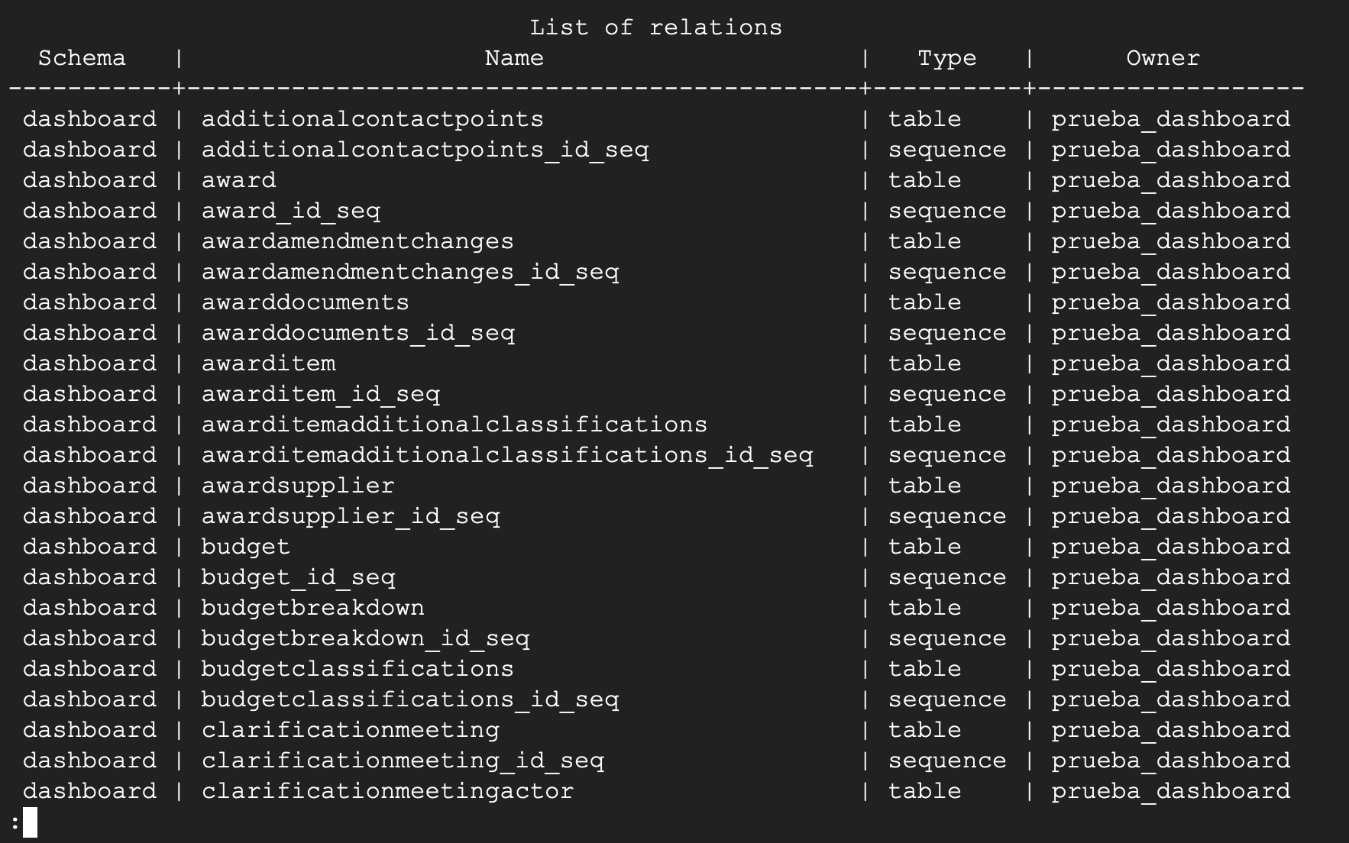
 Para salir, presionamos la tecla **q**.

su postgres

psql edca

\d

Para validar el **esquema dashboard**, ejecutamos los siguientes comandos, los cuales deben arrojar una respuesta similar a la siguiente.



set search\_path to dashboard;

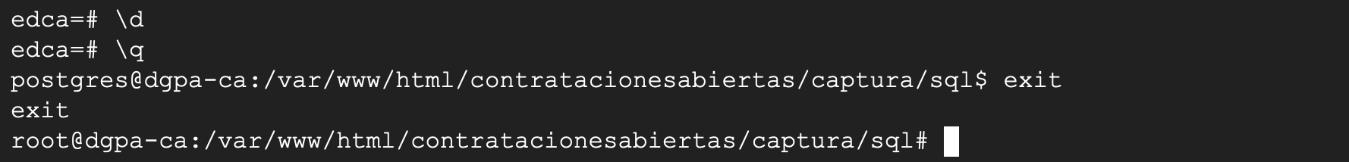
\d

Para salir, presionamos la tecla **q**.

Finalmente, cerramos la sesión de PostgreSQL.

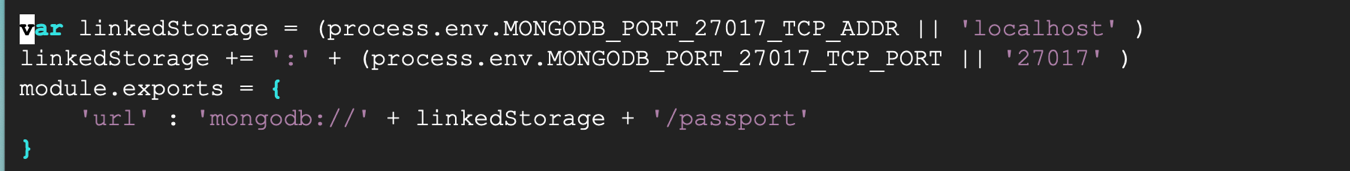
\q

exit



## Validación de la conexión a la base de datos

Ingresamos a la ruta siguiente: **/var/www/html/contratacionesabiertas/captura/**. Dentro de ella, abrimos el archivo llamado **db.js** y verificamos que la variable **linkedStorage** esté definida de la siguiente manera.



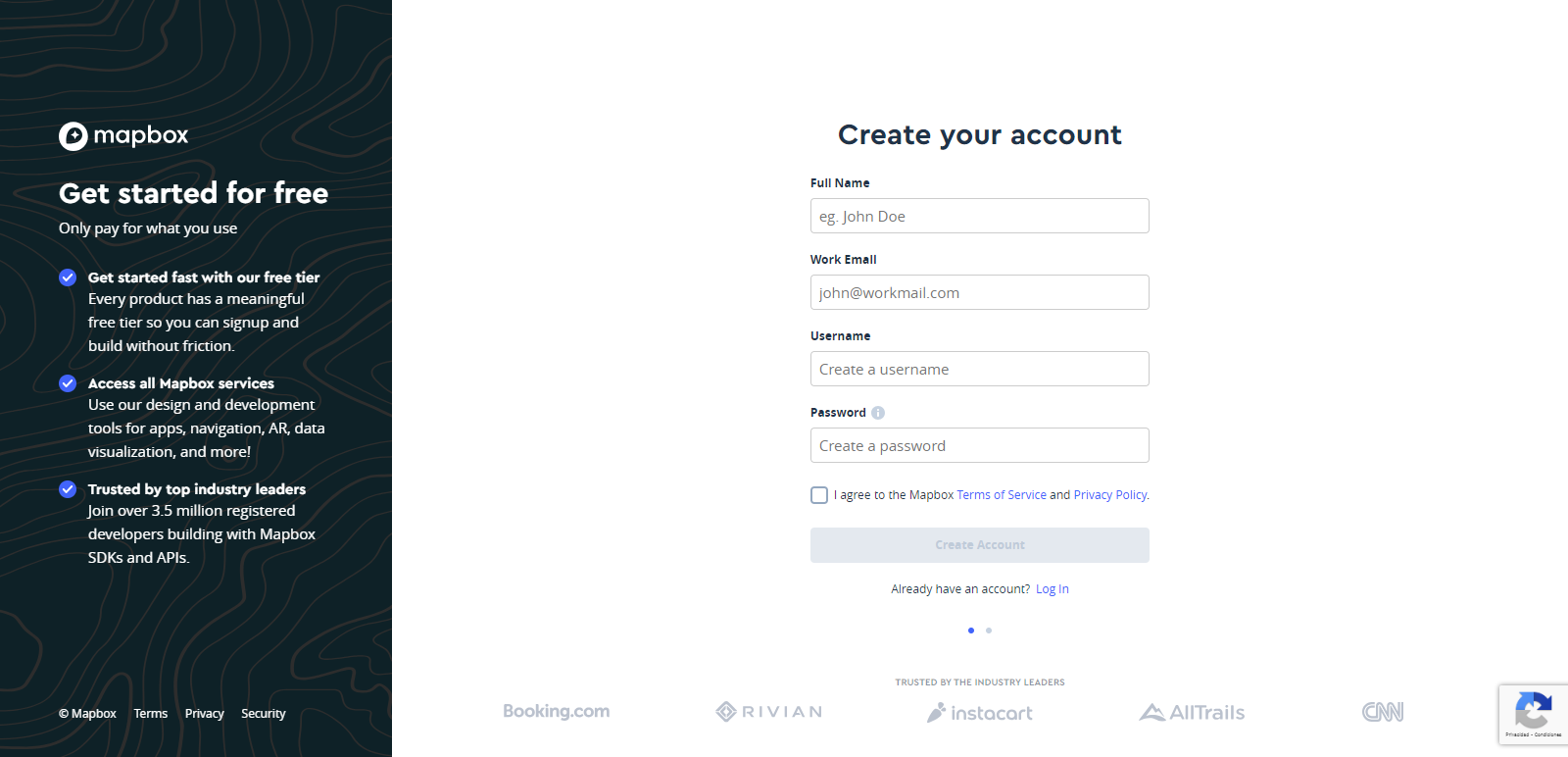
cd /var/www/html/contratacionesabiertas/captura

vi db.js

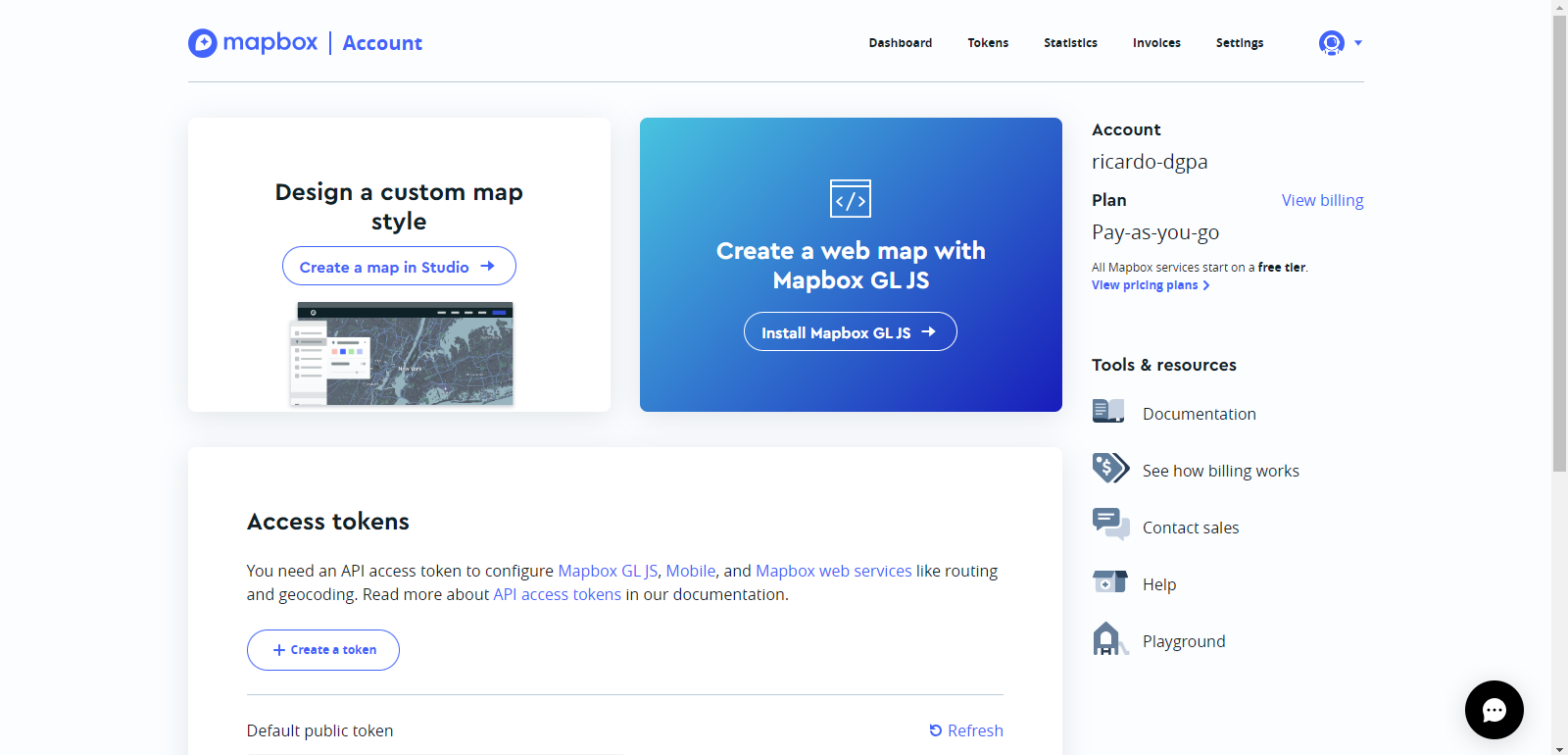
Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:q**.

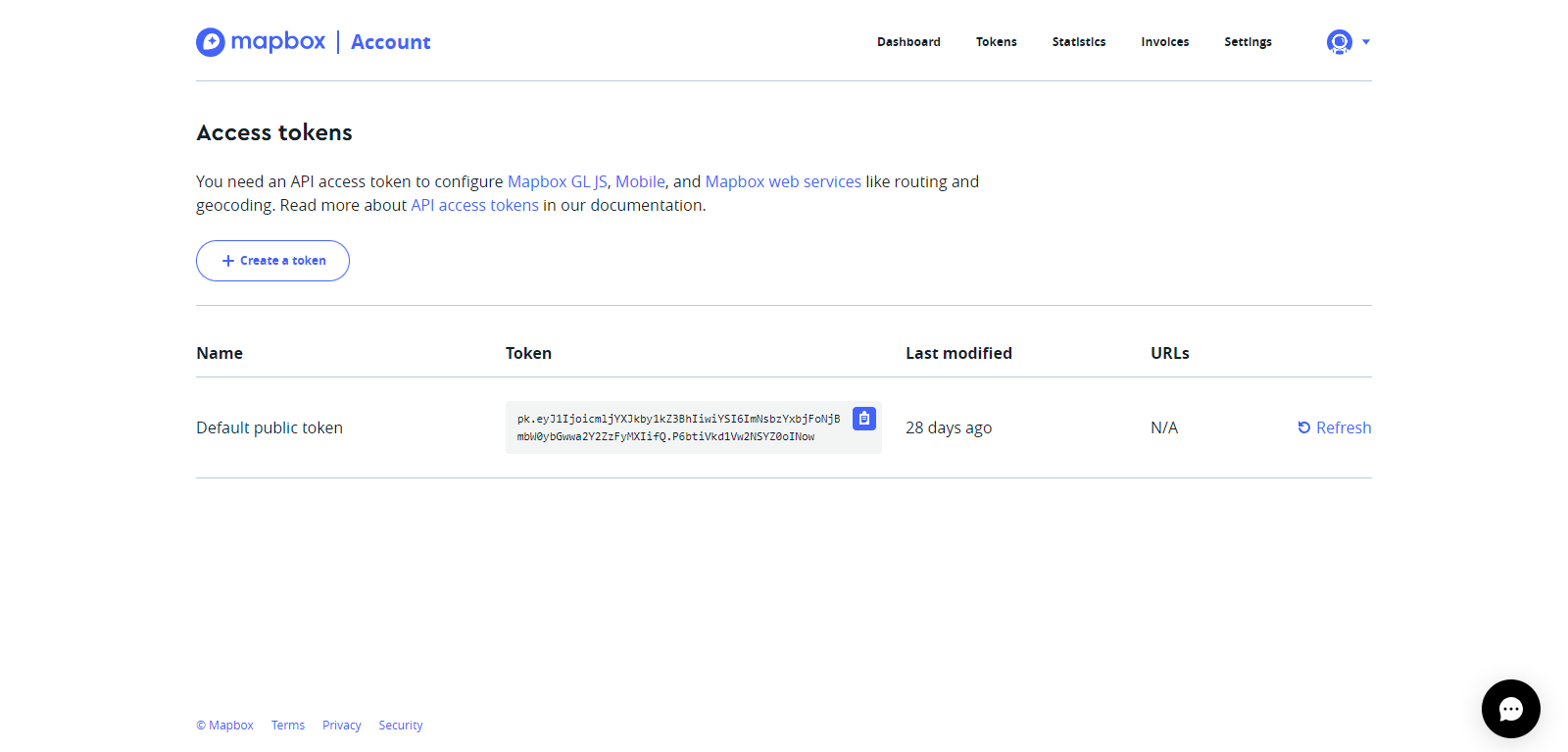
## Asignación de variables para el token de mapa (MAPTOKEN)

Ingresamos a https://www.mapbox.com/signup/, nos registramos e iniciamos sesión.

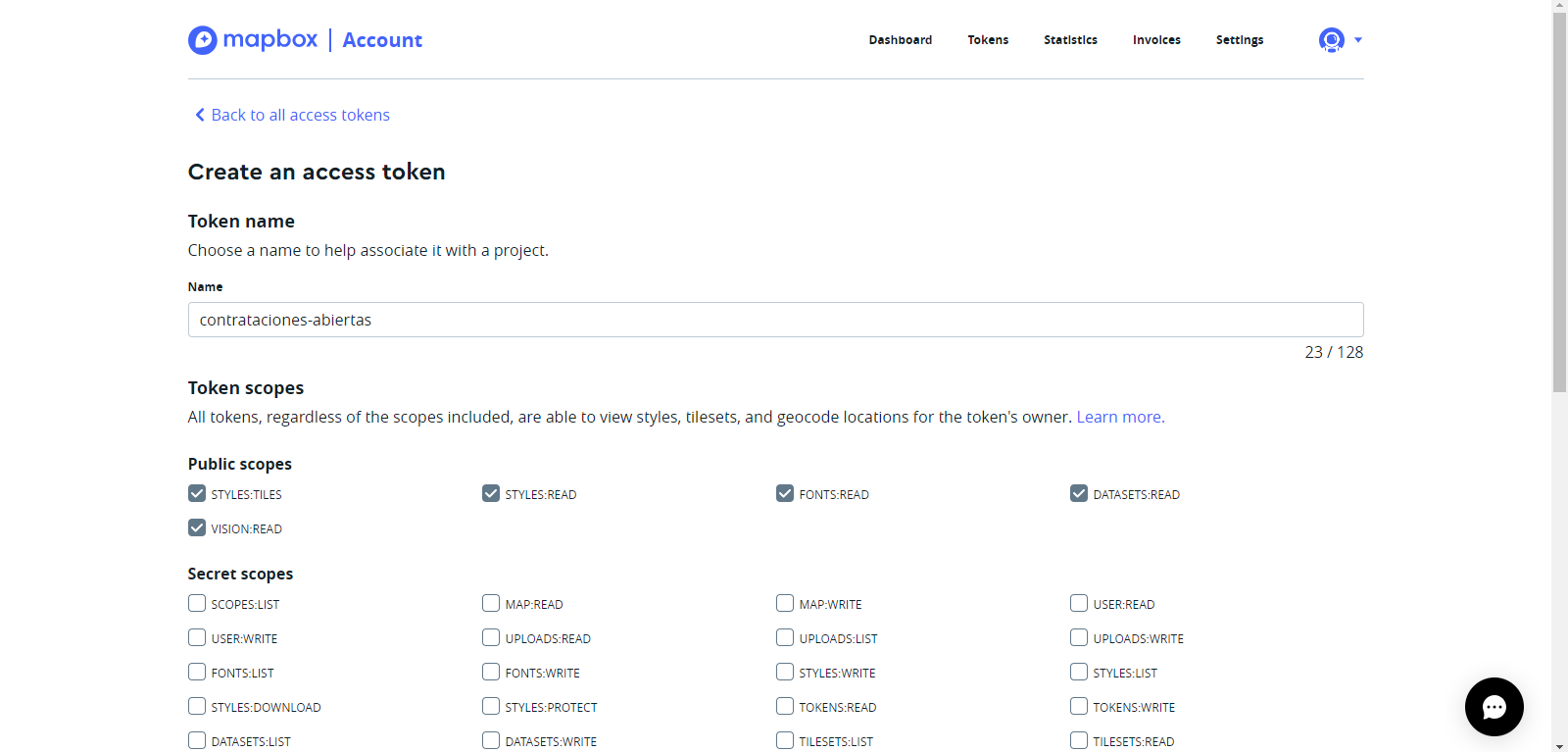


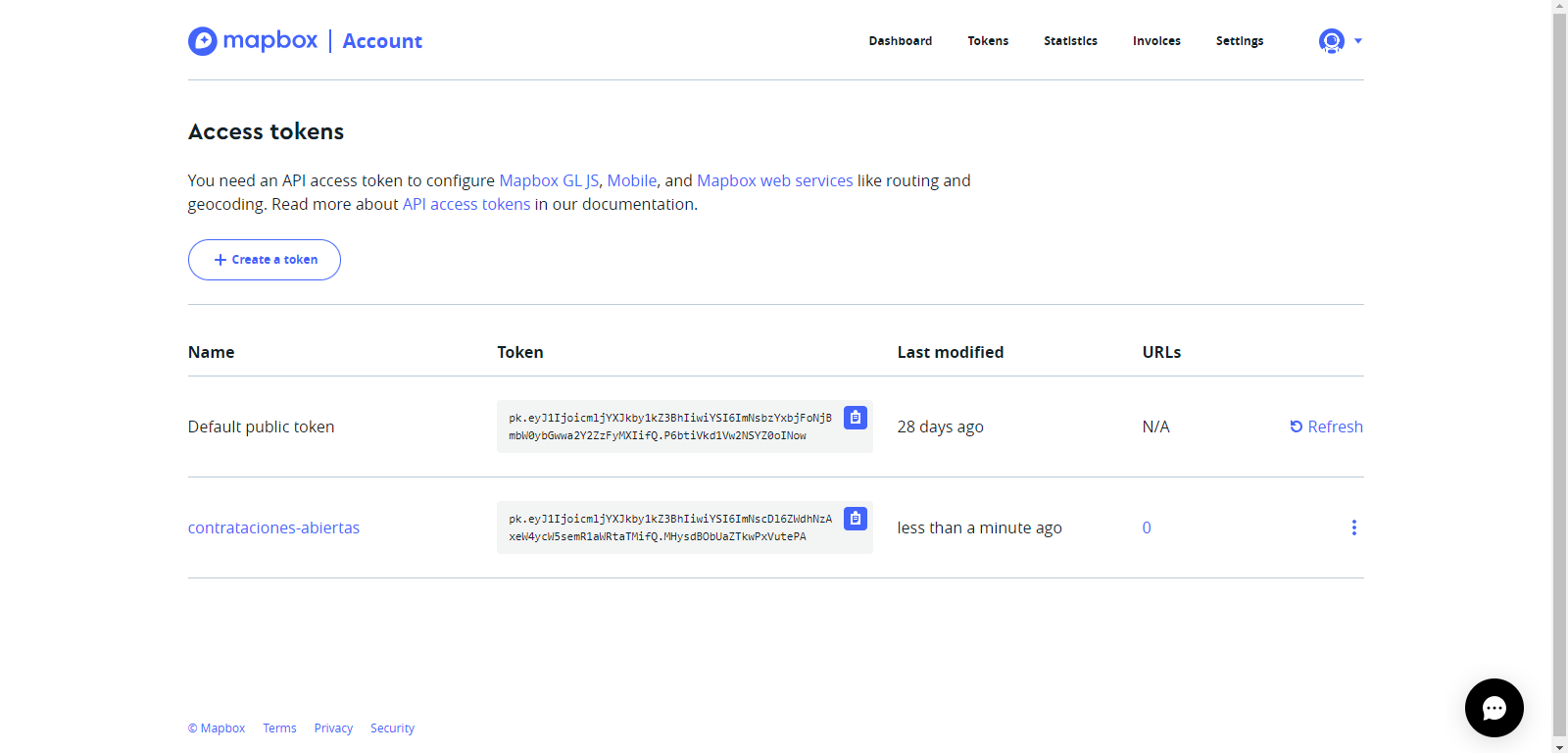
Una vez dentro, nos dirigimos a la sección de **tokens** y hacemos clic en el botón que dice **Create a token.**





Le asignamos un nombre y dejamos la configuración existente. Finalmente, pulsamos el botón **Create token.**



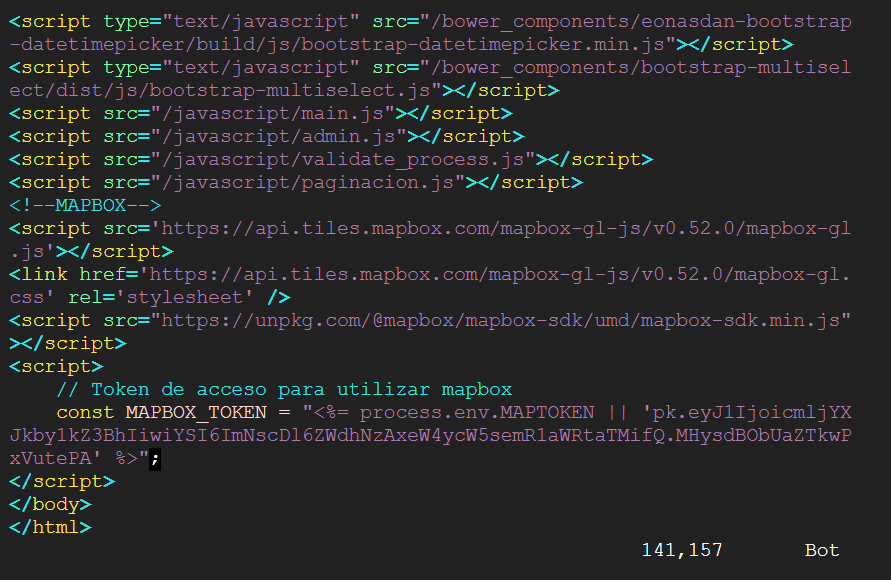


Una vez obtengamos nuestro token, lo copiaremos y lo pegaremos en las siguientes rutas. La primera ruta es **/var/www/html/contratacionesabiertas/captura/views**. Dentro de esta carpeta, editamos el archivo llamado **main.ejs** para asignar el valor de nuestro token a la variable **MAPBOX\_TOKEN**.



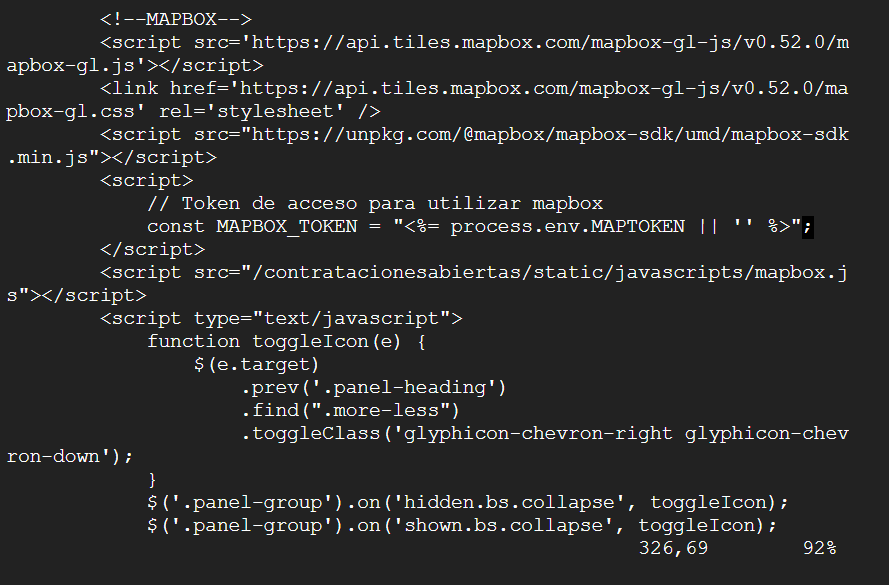
cd /var/www/html/contratacionesabiertas/captura/views

vi main.ejs



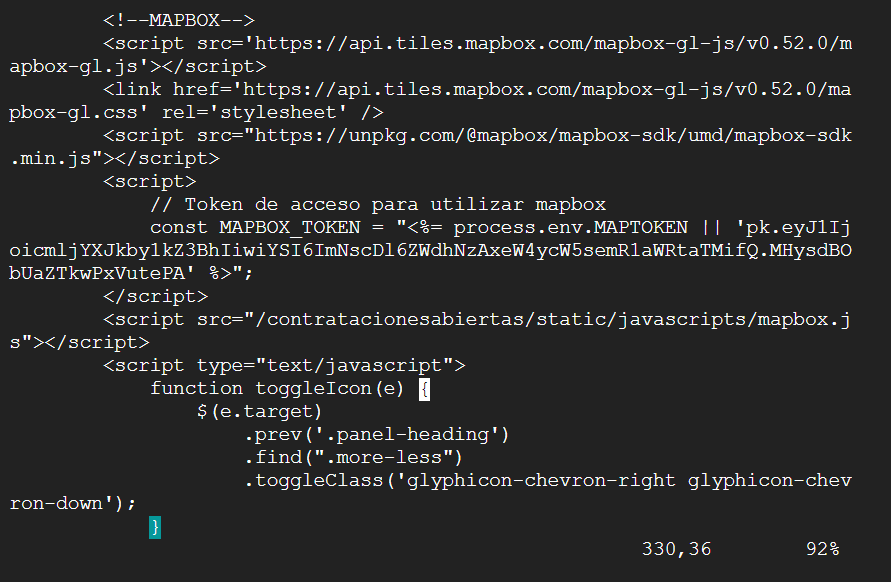
Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

Ahora ingresamos **a /var/www/html/contratacionesabiertas/dashboard/views** y editamos el archivo llamado **contract.ejs** para asignar el valor de nuestro token a la variable **MAPBOX\_TOKEN**.



cd /var/www/html/contratacionesabiertas/dashboard/views

vi contract.ejs

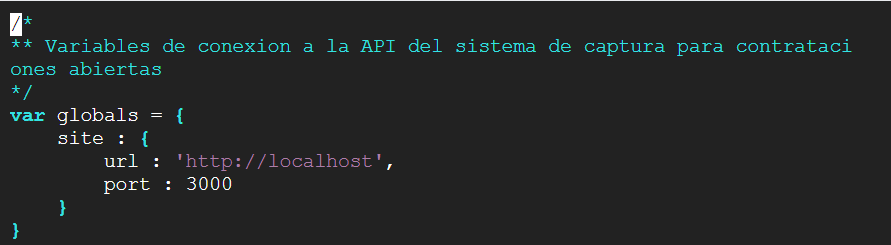


Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

## Configuración de IP

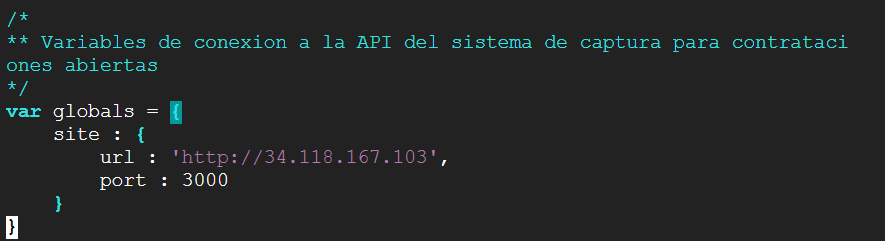
Ingresamos a la ruta **/var/www/html/contratacionesabiertas/dashboard/public/javascripts** y editamos el archivo **common.js** para asignar la IP correspondiente del servidor y el puerto asignado al módulo de captura.

 Si la herramienta se despliega sobre el protocolo **HTTP**, es necesario indicar en la variable url la IP del servidor con este protocolo; en caso de que se despliegue sobre **HTTPS**, se debe especificar en esta misma variable. **Esto es fundamental para evitar errores en las peticiones de la herramienta tecnológica con el API.**



cd /var/www/html/contratacionesabiertas/dashboard/public/javascripts

vi common.js



Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

## Instalación de Wkhtmltopdf

Realizamos la instalación de de wkhtmltopdf.

apt-get install -y wkhtmltopdf

Instalamos el paquete de wkhtmltopdf dentro de dashboard.

cd /var/www/html/contratacionesabiertas/dashboard

npm install wkhtmltopdf

## Módulo de Infraestructura

Antes de configurar el módulo de infraestructura, debemos instalar **pg** y **sequelize-cli**. Para ello, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/captura** y ejecutamos lo siguiente.

cd /var/www/html/contratacionesabiertas/captura

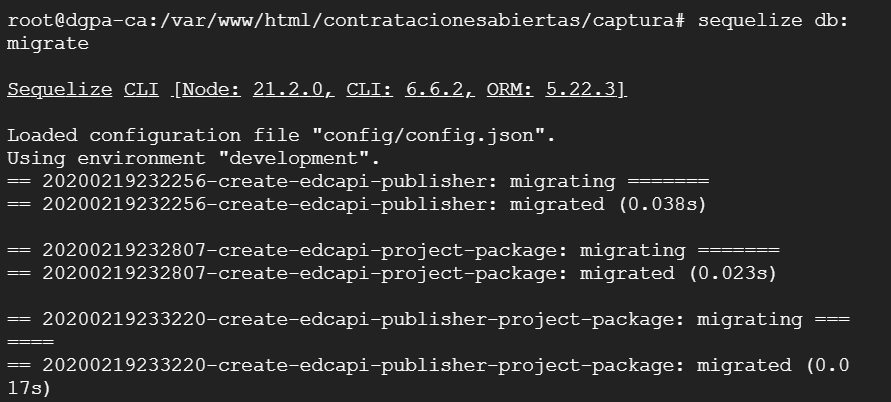
rm -Rf node\_modules

npm install

npm install --save pg@latest

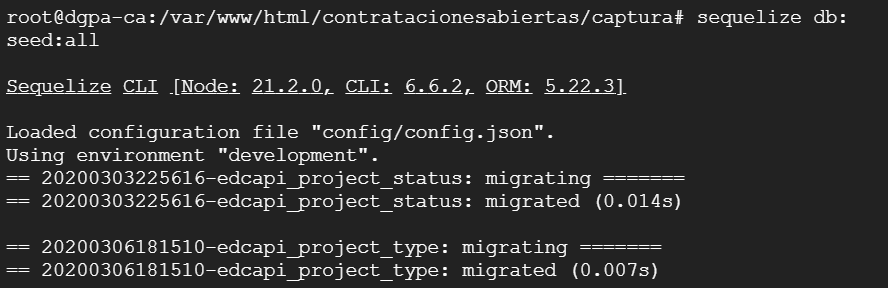
npm install -g sequelize-cli

Una vez instalado lo necesario, procedemos a ejecutar los siguientes comandos para instalar el módulo.



sequelize db:migrate

sequelize db:seed:all



## Creación de vistas para el módulo de Infraestructura Abierta

Nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/captura/sql** y ejecutamos el siguiente comando.

cd /var/www/html/contratacionesabiertas/captura/sql

sudo -u postgres psql edca < update\_InfraestructuraAbiertav1.sql



## Creación del usuario del aplicativo

Ingresamos a la ruta **/var/www/html/contratacionesabiertas/captura**, le asignamos los permisos de ejecución al script **useradm** y lo ejecutamos para crear el usuario administrador del **módulo de captura.**

 Se utiliza **administrador** a modo de ejemplo, este nombre puede ser modificado por cualquier otro que se desee.



cd /var/www/html/contratacionesabiertas/captura

chmod u+x useradm

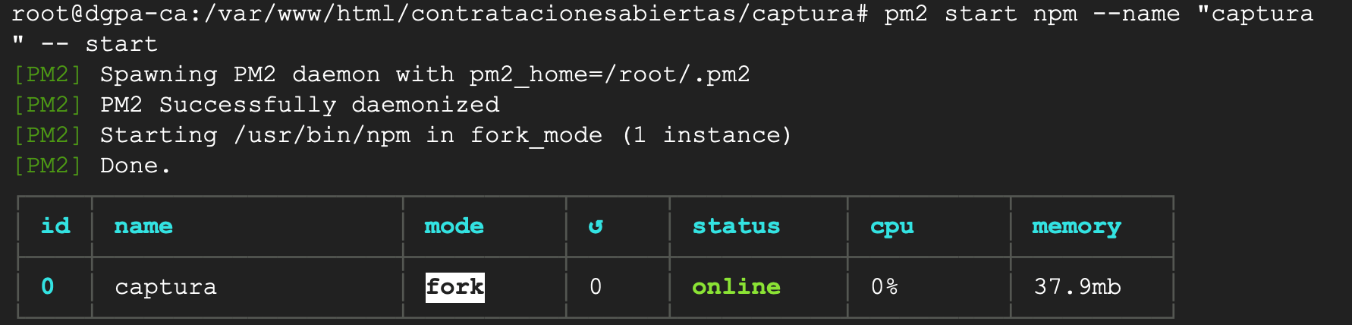
./useradm add administrador

Recordando que los datos ingresados deben cumplir con las siguientes características:

* Nombre y Apellido: Solo se permiten caracteres alfabéticos.
* Contraseña: Se permiten caracteres alfanuméricos.
* ¿El usuario es un administrador?: Admite únicamente "true" o "false".
* Correo: Permite caracteres alfanuméricos y debe de mantener la estructura de usuario@servidor.com.
* Dirección: Se permiten caracteres alfanuméricos y debe de incluir dirección completa (calle, número, municipio, entre otros).

## Iniciar los servicios de los módulos

Ingresamos a la ruta **/var/www/html/contratacionesabiertas/captura** e iniciamos el servicio del módulo de captura.



pm2 start npm --name "captura" -- start

Para acceder al **módulo de captura**, necesitamos colocar nuestra URL en el navegador, seguida del puerto 3000 o el que hayamos configurado. Ejemplo: **http://url\_servidor:3000**.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Antes de iniciar el servicio del **módulo de dashboard**, es necesario actualizar los paquetes a la última versión. Para ello, primero ingresamos a la siguiente ruta **/var/www/html/contratacionesabiertas/dashboard** e instalamos **npm-check-updates**.

cd /var/www/html/contratacionesabiertas/dashboard

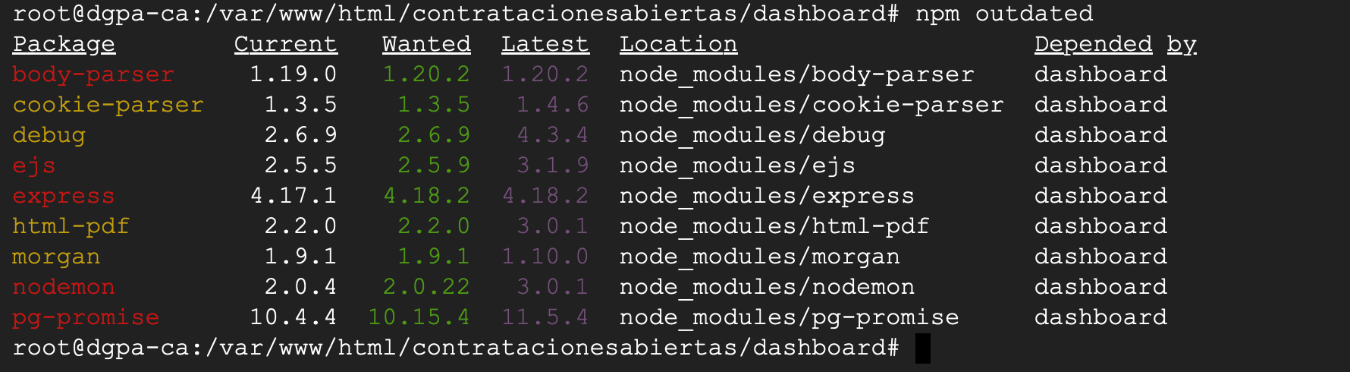
npm install -g npm-check-updates

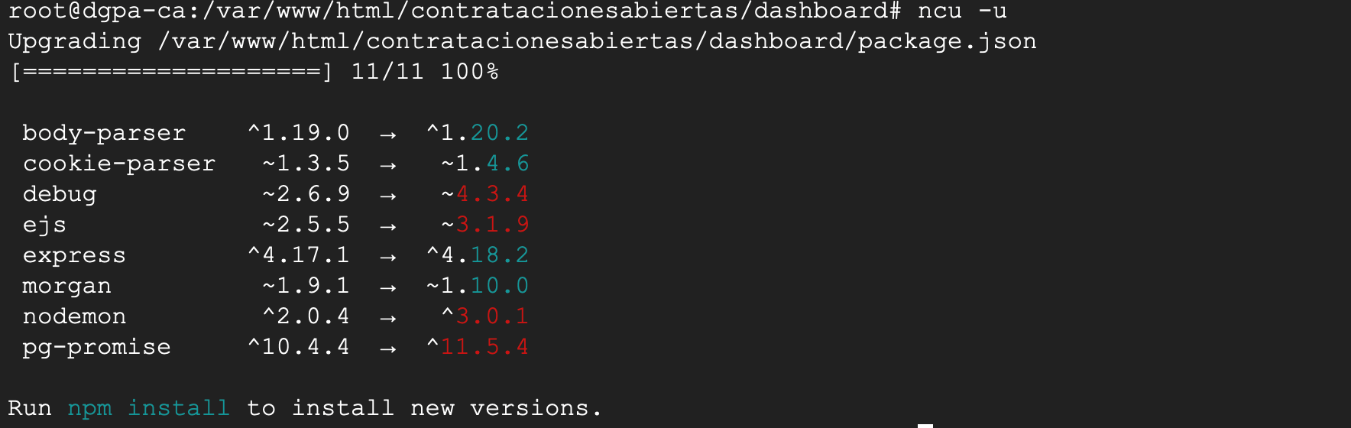
Ahora ejecutamos el siguiente comando para verificar la versión más reciente de los paquetes del **módulo de dashboard**. Posteriormente, mediante el segundo comando, modificaremos el archivo **package.json** para que al ejecutar el tercer comando se realice la actualización.

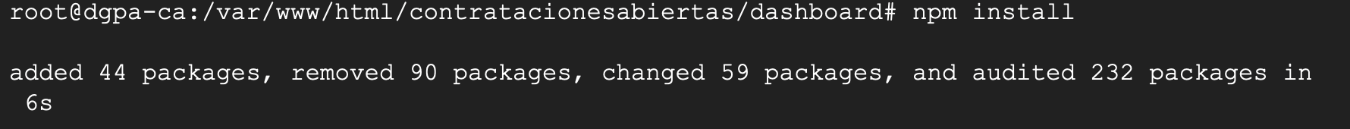
npm outdated

ncu -u

npm install

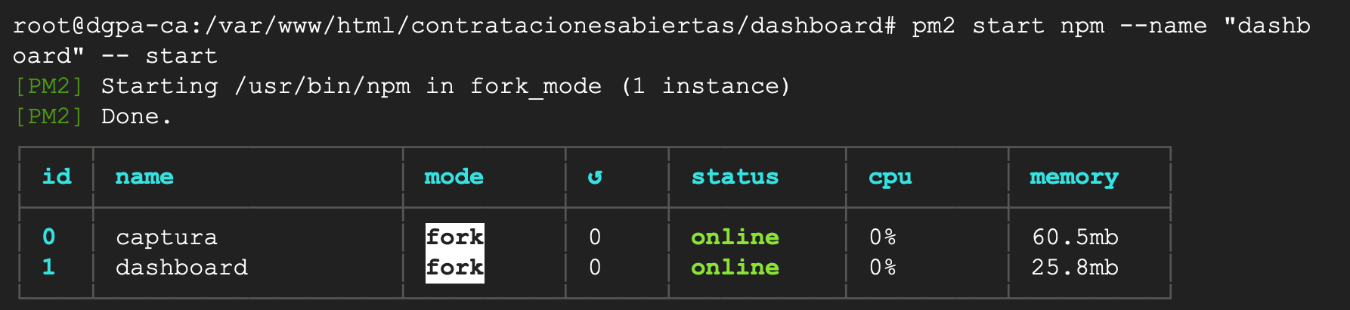






Una vez completada la actualización, procedemos a iniciar el servicio del **módulo de dashboard**.

pm2 start npm --name "dashboard" -- start



Para acceder al **módulo de dashboard**, necesitamos colocar nuestra URL en el navegador, seguida del puerto 4000 o el que hayamos configurado, además de la ruta **contratacionesabiertas** y **contratos**. Ejemplo: **http://url\_servidor:4000/contratacionesabiertas/contratos**.

Escala de tiempo

Descripción generada automáticamente

# **Configuración de Google Analytics**

Para empezar, creamos una cuenta en **Google Analytics** y creamos una propiedad exclusiva para recopilar información de la herramienta de contrataciones abiertas. Esto generará una **etiqueta global** para el sitio web.

 Para obtener más detalles sobre cómo crear la propiedad, puedes consultar el siguiente enlace: <https://support.google.com/analytics/answer/1008080?hl=es#zippy=%2Csecciones-de-este-artículo>.

Posicionaremos esta etiqueta en diferentes secciones dentro de la herramienta. Nos dirigiremos a la ruta /var/www/html/contratacionesabiertas/dashboard/views y editaremos los archivos **contracts.ejs, dashboard.ejs, datosabiertos.ejs, implementa.ejs, index.ejs y politicadepublicacion.ejs**, colocando la etiqueta generada justo debajo de la etiqueta head.

cd /var/www/html/contratacionesabiertas/dashboard/views

vi contract.ejs

vi dashboard.ejs

vi datosabiertos.ejs

vi implementa.ejs

vi index.ejs

vi politicadepublicacion.ejs



Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

# **Personalización de la herramienta**

## Cambiar logo de la institución

Este logotipo se visualizará tanto en el **módulo de captura** como en el **módulo de dashboard** en la parte superior izquierda.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Sitio web, Escala de tiempo

Descripción generada automáticamente

Para realizar el cambio primero se debe de considerar que el logotipo debe de estar en formato PNG con las dimensiones de 390x194 pixeles.

En el **módulo de captura**, se debe cargar el logotipo en la ruta **/var/www/html/contratacionesabiertas/captura/public/images**. Una vez cargado, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/captura/public/inai-navbar** y editamos el archivo **dgm-navbar.html**. En este archivo, actualizaremos el nombre del logotipo y la URL del sitio institucional.

cd /var/www/html/contratacionesabiertas/captura/public/inai-navbar

vi dgm-navbar.html

Texto

Descripción generada automáticamente

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

En el **módulo de dashboard**, se debe cargar el logotipo en la ruta **/var/www/html/contratacionesabiertas/dashboard/public/images**. Una vez cargado, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/dashboard/public/inai-navbar** y editamos el archivo **dgm-navbar.html**. En este archivo, actualizaremos el nombre del logotipo y la URL del sitio institucional.

cd /var/www/html/contratacionesabiertas/dashboard/public/inai-navbar

vi dgm-navbar.html

Captura de pantalla de computadora

Descripción generada automáticamente

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

## Actualizar las siglas de la institución

Estas siglas se encuentran en el **módulo de dashboard** y deben ser actualizadas para que la herramienta haga referencia de manera precisa a la institución implementadora.

Sitio web, Escala de tiempo

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para ello, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/dashboard/views**. En este directorio, modificamos el archivo **dashboard.ejs** y editamos las siglas de **INAI** por las correspondientes a la institución.

cd /var/www/html/contratacionesabiertas/dashboard/views

vi dashboard.ejs

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

## Actualizar el pie de página del módulo de dashboard

Actualizar el pie de página nos permitirá incluir las redes sociales, los sitios de interés, datos de contacto, así como el logo de la institución implementadora para que los usuarios puedan consultarlo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para realizar el cambio, se debe cargar el logotipo en la ruta **/var/www/html/contratacionesabiertas/dashboard/public/images**. También se puede utilizar el que previamente se empleó en el header. Una vez cargado, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/dashboard/public/inai-footer** y editamos el archivo **dgm-footer.html**. En este archivo, actualizaremos el nombre del logotipo y la URL del sitio institucional.

cd /var/www/html/contratacionesabiertas/dashboard/public/inai-footer

vi dgm-footer.html

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Sin salirnos del archivo, modificamos las URL de las redes sociales correspondientes a la institución implementadora.

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

Siguiendo en el archivo, debajo de las redes sociales, modificamos los sitios de interés por aquellos que se deseen colocar. Ya sea la misma cantidad de sitios o una cantidad diferente, esto queda a decisión de la institución implementadora.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Finalmente, antes de cerrar este archivo, modificamos la sección de contacto por la correspondiente a la institución implementadora.

Imagen que contiene Aplicación

Descripción generada automáticamente

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

## Cambiar la ruta de navegación

El cambio de ruta de navegación es importante para evitar confusiones entre las personas que consultan la información y garantizar que estén seguras de estar accediendo a la correcta.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Para esto, nos dirigimos a la ruta **/var/www/html/contratacionesabiertas/dashboard/views**. En esta ruta, editaremos varios archivos. El primero de ellos es **politicadepublicacion.ejs**. En este, cambiamos el nombre del **INAI** por el de la institución implementadora sin abreviaturas.

cd /var/www/html/contratacionesabiertas/dashboard/views

vi politicadepublicacion.ejs

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

El siguiente archivo es **implementa.ejs,** de la misma forma debemos de cambiar el nombre del INAI por el de la institución implementadora sin abreviaturas.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

vi implementa.ejs

Texto

Descripción generada automáticamente

Para salir del editor, presionamos la tecla **Esc** y posteriormente introducimos la siguiente secuencia de teclas **:wq**.

El siguiente archivo es **datosabiertos.ejs,** de la misma forma debemos de cambiar el nombre del INAI por el de la institución implementadora sin abreviaturas.

Interfaz de usuario gráfica

Descripción generada automáticamente

vi datosabiertos.ejs

Captura de pantalla de un celular

Descripción generada automáticamente

El último archivo es **contract.ejs,** de la misma forma debemos de cambiar el nombre del INAI por el de la institución implementadora sin abreviaturas.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

vi contract.ejs

Texto

Descripción generada automáticamente

# **Anexo A: Solución a errores comunes**

## Error en la creación de una nueva contratación

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web

Descripción generada automáticamente

Este problema surge debido a la omisión del apartado [**Creación de vistas para el módulo de Infraestructura Abierta**](#_Creación_de_vistas). Estas deben ser creadas para garantizar el correcto funcionamiento de la herramienta. Para solucionarlo, diríjase a esta sección y lleve a cabo los pasos que en ella se describen.

# Anexo B: Definiciones

## Ubuntu

Sistema operativo de código abierto basado en Linux. Proporciona un entorno operativo robusto y seguro para computadoras personales y servidores, es ampliamente utilizado en el desarrollo y despliegue de software.

## PostgreSQL

Sistema de gestión de bases de datos relacional de código abierto. Almacena y gestiona datos estructurados, es utilizado en aplicaciones web y empresariales.

## MongoDB

Base de datos NoSQL orientada a documentos. Almacena datos en formato BSON (JSON binario), es eficiente para datos no estructurados o semiestructurados.

## NodeJS

Entorno de ejecución de JavaScript del lado del servidor. Permite desarrollar aplicaciones escalables y de alto rendimiento, especialmente aplicaciones web.

## NGINX

Servidor web y proxy inverso. Maneja la distribución de tráfico web, mejora la velocidad y maneja solicitudes HTTP y HTTPS.

## PM2

Gestor de procesos para aplicaciones Node.js. Mantiene aplicaciones Node.js en ejecución continua, gestiona el equilibrio de carga y facilita la administración de procesos.

## Git

Sistema de control de versiones distribuido. Registra cambios en el código fuente durante el desarrollo de software, facilita la colaboración entre desarrolladores y el control de versiones.

## VIM

Editor de texto modal de código abierto. Permite la edición eficiente de archivos de texto en entornos de terminal, con una amplia variedad de comandos y modos, siendo popular entre desarrolladores y administradores de sistemas.

## Mapbox

Plataforma de mapeo en la nube y servicios de geolocalización. Proporciona herramientas y APIs para integrar mapas personalizados en aplicaciones, permitiendo la visualización de datos geoespaciales y la creación de experiencias de mapas interactivos.

## Sequelize-CLI

Interfaz de línea de comandos para Sequelize, un ORM (Object-Relational Mapping) para Node.js. Facilita la interacción con bases de datos relacionales, generando modelos, migraciones y ayudando en el desarrollo de aplicaciones Node.js que utilizan bases de datos SQL.

## Google Analytics

Servicio de análisis web ofrecido por Google. Permite a los propietarios de sitios web y aplicaciones rastrear y analizar el tráfico, el comportamiento del usuario y otros datos relevantes, proporcionando información valiosa para la toma de decisiones en marketing digital y mejora de la experiencia del usuario.

## Wkhtmltopdf

Herramienta de conversión de HTML a PDF de código abierto. Utiliza el motor de renderizado WebKit para generar archivos PDF a partir de contenido HTML, permitiendo la creación de documentos de alta calidad y bien formateados a partir de páginas web.