Contenido

[1. Arquitectura](#_gjdgxs) **3**

[2. Creación de la base de datos](#_p108n7n4imb9) **4**

[3. Descargar el archivo del código](#_1fob9te) **6**

[4. Configuración de variables de conexión a la base de datos](#_vpk8h4uwil72) **7**

[5. Creación de la base de datos edca y del esquema public](#_2et92p0) **11**

[6. Configuración del usuario propietario para el esquema public](#_oktvabjhfpa0) **15**

[7. Creación del esquema dashboard](#_simy1dsmay53) **17**

[8. Configuración del usuario propietario para el esquema dashboard](#_si7uotw8zk5l) **20**

[9. Validación de conexión a base de datos](#_38gp1qermhba) **22**

[10. Asignación de variables para conexión a la API de la Plataforma Nacional de Transparencia](#_1t3h5sf) **22**

[11. Asignación de variables MAPTOKEN](#_4d34og8) **24**

[12. Instalación de Object Relational Mapping del módulo de Infraestructura](#_17dp8vu) **28**

[13. Crear usuario del aplicativo](#_86zl5romdevr) **30**

[14. Iniciar los servicios de los módulos](#_3rdcrjn) **31**

[15. Acceder a los módulos](#_26in1rg) **32**

[A. Módulo sistema de captura](#_a1i7c4kumya) 32

[B. Módulo de visualización (dashboard)](#_35nkun2) 33

[16. Configuración de Google Analytics para ambiente productivo](#_v0eihh7i4evu) **35**

[Anexo 1. Errores comunes.](#_2xg3k7ippo39) **36**

[Anexo 2. Configuraciones Generales para más de una instancia en el mismo servidor.](#_4r0lqgwfpo83) **38**

*Fecha de actualización: 16-Febrero-2021*

# Control de cambios

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Resumen del cambio** |
| 23/12/2020 | 2.1.0 | Integración del módulo de Infraestructura Abierta y la funcionalidad de interconexión con la PNT. |
| 16/02/2021 | 2.1.2 | Integración de la sección “Datos Abiertos”. |

# 

# 

# 1. Arquitectura

Requerimientos mínimos de hardware:

|  |  |
| --- | --- |
| Memoria RAM. | 4 GB o superior. |
| Velocidad del procesador. | 2.0 GHz o superior. |
| Almacenamiento. | 250 GB o superior. |

Requerimientos específicos de software:

|  |  |
| --- | --- |
| Sistema Operativo | CentOS 7 |
| Base de datos relacional | PostgreSQL 10 |
| Base de datos de documentos | MongoDB v3 |
| Lenguaje de programación | Nodejs v10 |
| Servidor web | NGINX |

**NOTA:**

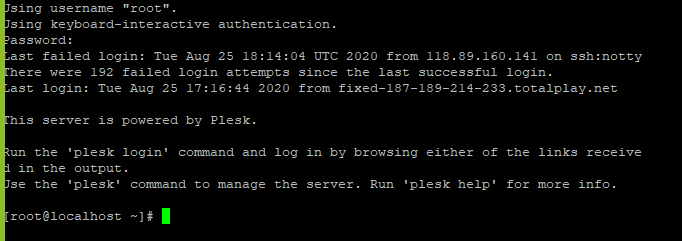
1. Se requiere que el servidor tenga salida a internet, esto permitirá la descarga de dependencias que existen para el funcionamiento de las gráficas del visualizador.
2. El servidor requiere de la configuración de dos puertos para permitir el acceso a los módulos de captura y dashboard. Actualmente, usa el puerto 3000 para el sistema de captura y el 4000 para el dashboard, sin embargo, estos puertos son parametrizables con la finalidad de poder tener más de una instancia en el mismo servidor.

Para configurar más de una instancia en el servidor, revisar el **Anexo 2**, al final de este documento.

1. Se necesita tener instalado el software: **git** y **pm2**. Para mayor referencia se pueden consultar las siguientes ligas: <https://desktop.github.com/>, <https://git-lfs.github.com/> y https://www.npmjs.com/package/pm2.
2. PostgreSQL debe de configurar los métodos de conexión **local** e **IPv4** como **trust.**

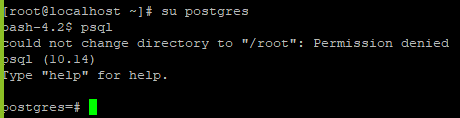
# 2. Creación de la base de datos

**Paso 1.** Conectarse al servidor con el usuario root y abrir una terminal.



**Paso 2.** ingresar a postgres para crear la base de datos, ejecutando los siguientes comandos:

|  |
| --- |
| > su postgres  > psql |



**Paso 3.** Una vez dentro de postgres, crear la base de datos llamándola “*edca”* utilizando el siguiente comando:

|  |
| --- |
| create database edca; |



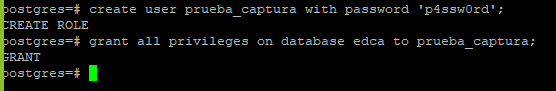
**NOTA:** Los nombres de “usuario” y la “contraseña” para esta configuración son de ejemplo.

**Paso 4.** Crear el usuario que se asignará como propietario del esquema public de la base y otorgarle todos los permisos sobre la base.

Para asignar el password, considerar usar solamente caracteres alfanuméricos y en minúsculas.

Se deben ejecutar los siguientes comandos:

|  |
| --- |
| create user *prueba\_captura* with password 'p4ssw0rd';  grant all privileges on database edca to *prueba\_captura*; |

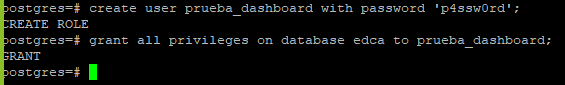


**Paso 5.** Crear el usuario que se asignará como propietario del esquema dashboard de la base y otorgarle todos los permisos sobre la base.

Para asignar el password, considerar usar solamente caracteres alfanuméricos y en minúsculas.

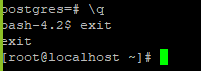
Se deben ejecutar los siguientes comandos:

|  |
| --- |
| create user *prueba\_dashboard* with password 'p4ssw0rd';  grant all privileges on database edca to *prueba\_dashboard*; |



**Paso 6.** Salir de postgres ejecutando los comandos:

|  |
| --- |
| \q  exit |

****

# 3. Descargar el archivo del código

**Paso 1.** Crear la carpeta en donde se realizará la instalación del código ejecutando el siguiente comando:

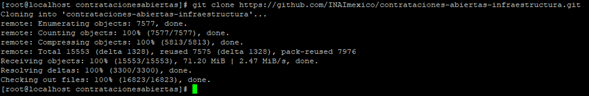
|  |
| --- |
| mkdir /var/www/html/contratacionesabiertas |



**Paso 2.** Ingresar a la carpeta recién creada y ejecutar el siguiente comando para clonar el repositorio con los archivos de instalación:

|  |
| --- |
| cd /var/www/html/contratacionesabiertas  git clone https://github.com/INAImexico/contrataciones-abiertas-infraestructura.git |





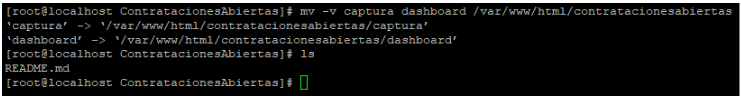
**Paso 3.** Acceder a la ruta que contiene los directorios *captura* y *dashboard,* ejecutando el siguiente comando:

|  |
| --- |
| cd contrataciones-abiertas-infraestructura/contratacionesabiertas/ |



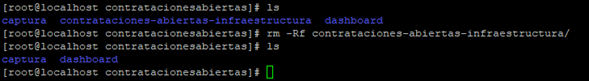
**Paso 4.** Mover los directorios *captura* y *dashboard* a la carpeta raíz, ejecutando el siguiente comando:

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



**Paso 5.** Posicionarse en la carpeta donde se clonó el repositorio para eliminarlo, ejecutando los siguientes comandos.

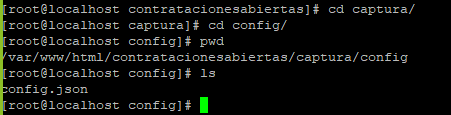
|  |
| --- |
| cd ../..  rm -rf contrataciones-abiertas-infraestructura/ |

**

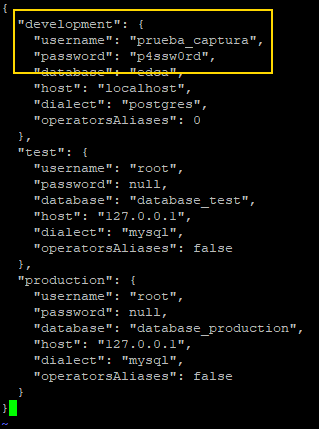
# 

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

**Paso 1.** Entrar a la ruta /contratacionesabiertas/captura/config y editar el archivo config.json, para colocar la cadena de conexión pertinente al ambiente.



Modificar las variables “username”, “password” y “database” correspondientes al usuario y password creados en el paso 4 del apartado 2, para el esquema public, módulo Sistema de captura.

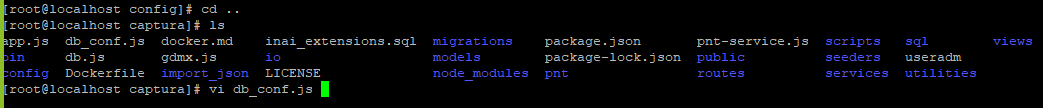


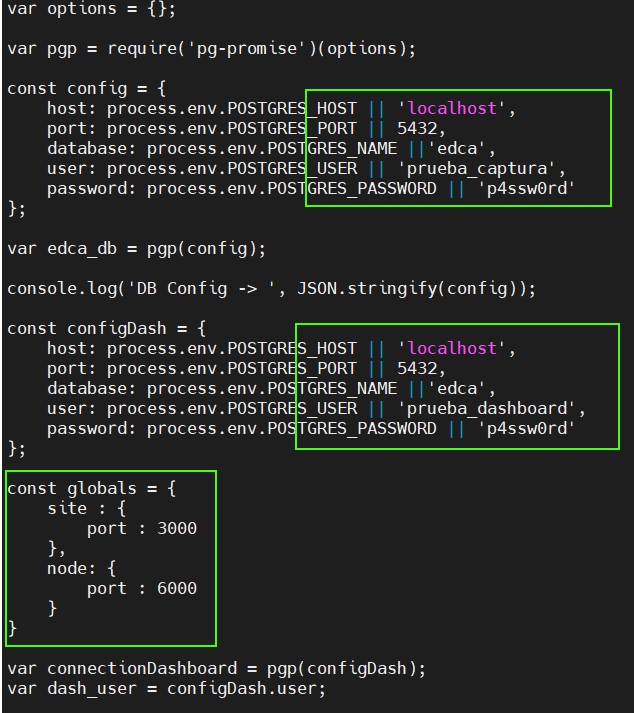
Salimos de la edición del archivo con el siguiente comando

*> :wq*

**Paso 2.** Ingresar a la carpeta /captura/ y editar en el archivo llamado “db\_conf.js” los datos de acuerdo a su configuración propia, HOST, PORT, DATABASE, USER y PASSWORD, para la conexión a la base de datos, los usuarios y contraseñas corresponden a los creados para los esquemas public y dashboard respectivamente; y configurar las “**const globals**” con los puertos designados para la instalación, ejecutando los comandos siguientes:

|  |
| --- |
| cd captura  vi db\_conf.js |



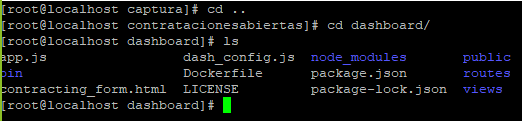


Actualizar, y guardar los cambios con el siguiente comando:

> *:wq*

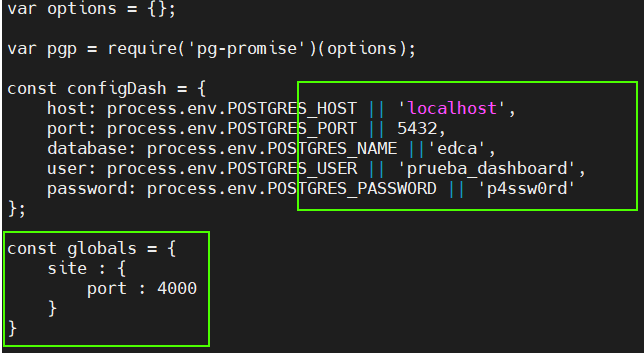
**Paso 3.** Salir de la carpeta /carpeta/sql/ e ingresar a la carpeta /dashboard/, ejecutar los siguientes comandos:

|  |
| --- |
| cd ..  cd dashboard |



**Paso 4.** Editar en el archivo llamado “dash\_conf.js” los datos de acuerdo a su configuración propia, HOST, PORT, USER y PASSWORD, para la conexión a la base de datos, el usuario y contraseña corresponde al creado para el schema dashboard; y configurar las “**const globals**” con los puertos designados para la instalación, ejecutar el comando siguiente:

|  |
| --- |
| vi dash\_conf.js |



Actualizar, y guardar los cambios con el comando:

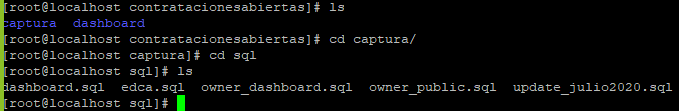
> *:wq*

# 5. Creación de la base de datos edca y del esquema public

Editar y ejecutar el script “edca.sql” para crear la estructura en la base edca

**Paso 1.** Ingresar a la carpeta /captura/sql/ y editar el archivo edca.sql que contiene la estructura de la base de datos:

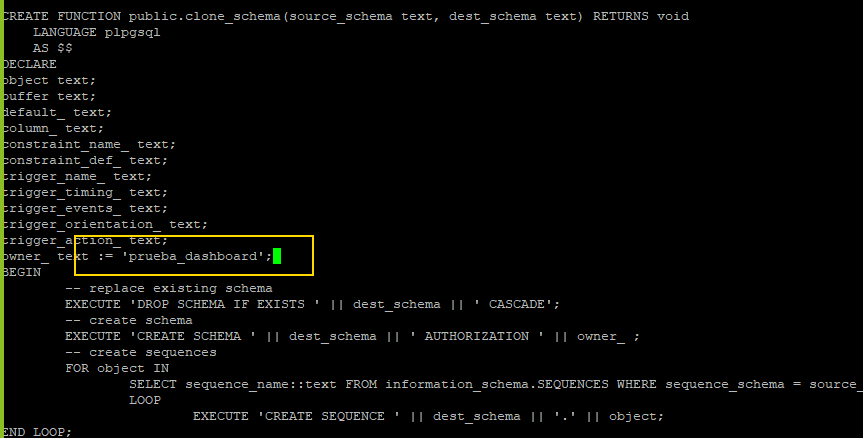
|  |
| --- |
| cd captura  cd sql |



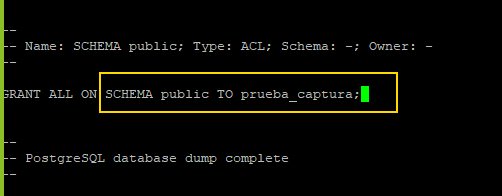
**Paso 2.** Editar el archivo para actualizar el nombre del usuario del esquema public y del esquema dashboard respectivamente, con los usuarios creados en el paso: **“2. Creación y configuración de la base de datos.”**

> vi edca.sql

Actualizar el nombre del usuario del esquema dashboard

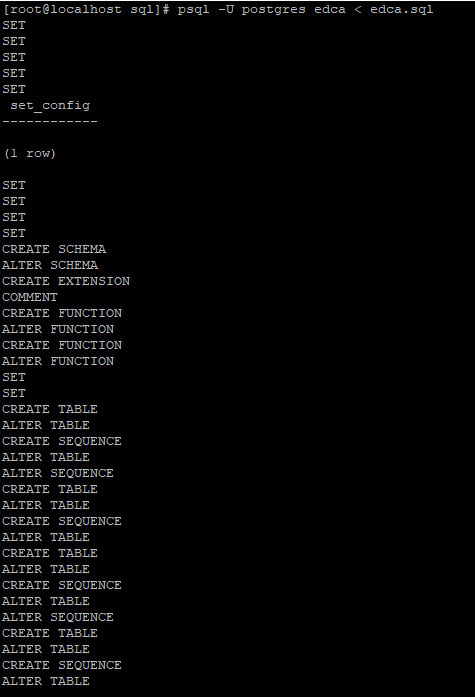


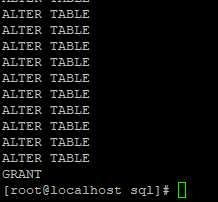
Actualizar el nombre del usuario del esquema public, que se encuentra en la parte final del documento.



**Paso 3.** Ejecutar el script ya actualizado “edca.sql”que contiene la estructura de la base de datos, ejecutar el siguiente comando:

|  |
| --- |
| *psql -U postgres edca < edca.sql* |



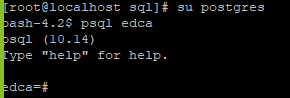


La ejecución de este script no debe de mostrar error.

**Paso 4.** Para verificar la creación correcta de la base de datos, ingresamos a postgres y a la base de datos, ejecutando los comandos:

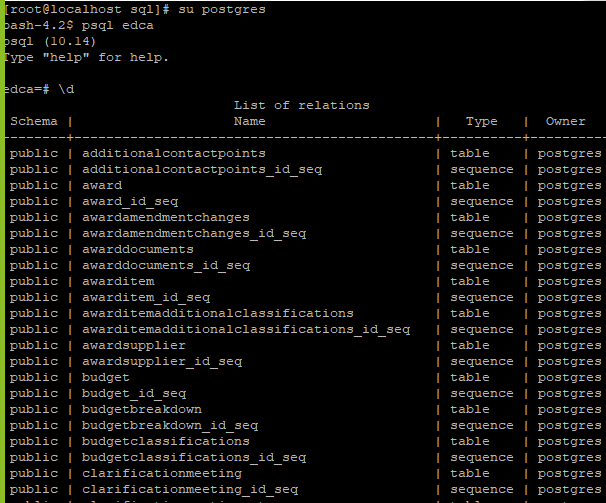
> su postgres

> psql edca



**Paso 5.** Consultar el usuario propietario del esquema public, con el comando:

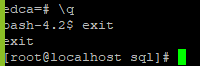
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

> exit



# 

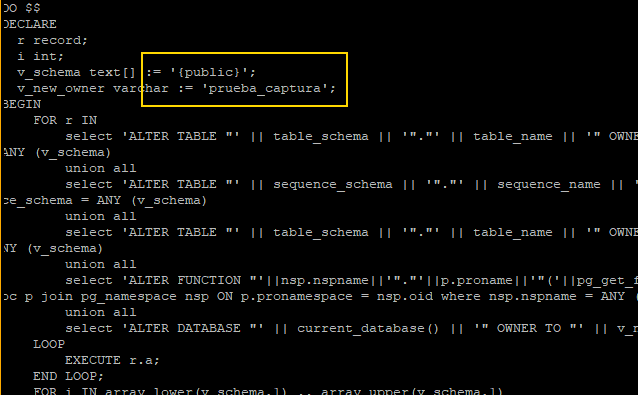
# 6. Configuración del usuario propietario para el esquema public

# 

Editar y ejecutar el script “owner\_public.sql” para actualizar el propietario del esquema public, con el usuario creado previamente

**Paso 1.** Editar el archivo “owner\_public.sql” con el comando:

> vi owner\_public.sql

****

Guardar y salir con el comando:

> :wq

**Paso 2.** Ejecutar el script actualizado con el comando:

> psql -U postgres edca < owner\_public.sql

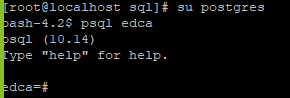


La ejecución de este script no debe de mostrar error.

**Paso 3.** Para verificar la creación correcta de la base de datos, ingresamos a postgres y a la base de datos, ejecutando los comandos:

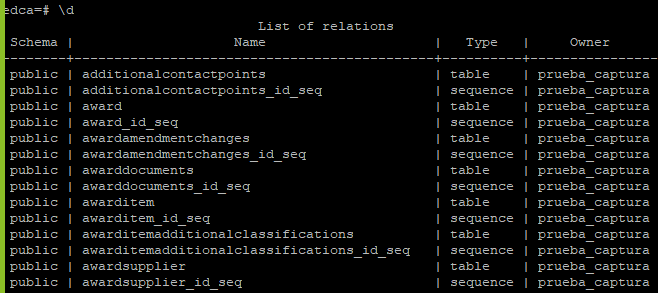
> su postgres

> psql edca



**Paso 4.** Consultar el usuario propietario del esquema public que haya sido actualizado, con el comando:

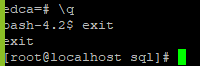
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

> exit

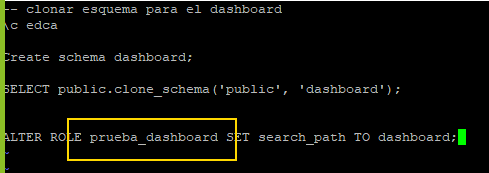


# 7. Creación del esquema dashboard

Editar y ejecutar el script “dashboard.sql” para la creación del esquema dashboard

**Paso 1.** Editar el archivo “dashboard.sql” con el comando:

> vi dashboard.sql

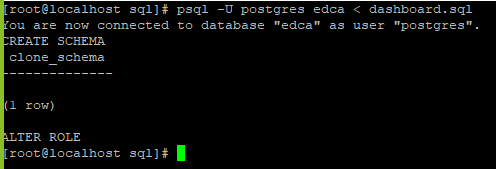


Guardar y salir con el comando:

> :wq

**Paso 2.** Ejecutar el script actualizado con el comando:

> psql -U postgres edca < dashboard.sql

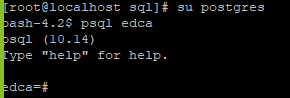


La ejecución de este script no debe de mostrar error.

**Paso 3.** Para verificar la creación correcta de la base de datos, ingresamos a postgres y a la base de datos, ejecutando los comandos:

> su postgres

> psql edca



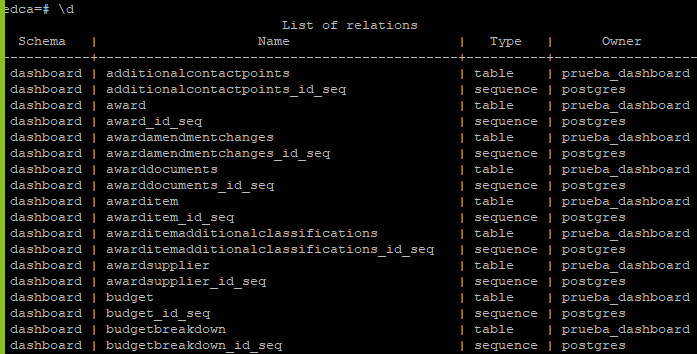
**Paso 4.** Cambiar el esquema de la base, ejecutando el comando:

> set search\_path to dashboard;



**Paso 5.** Consultar que se hayan clonado las tablas y secuencias correspondientes, ejecutando el comando:

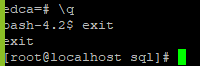
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

> exit



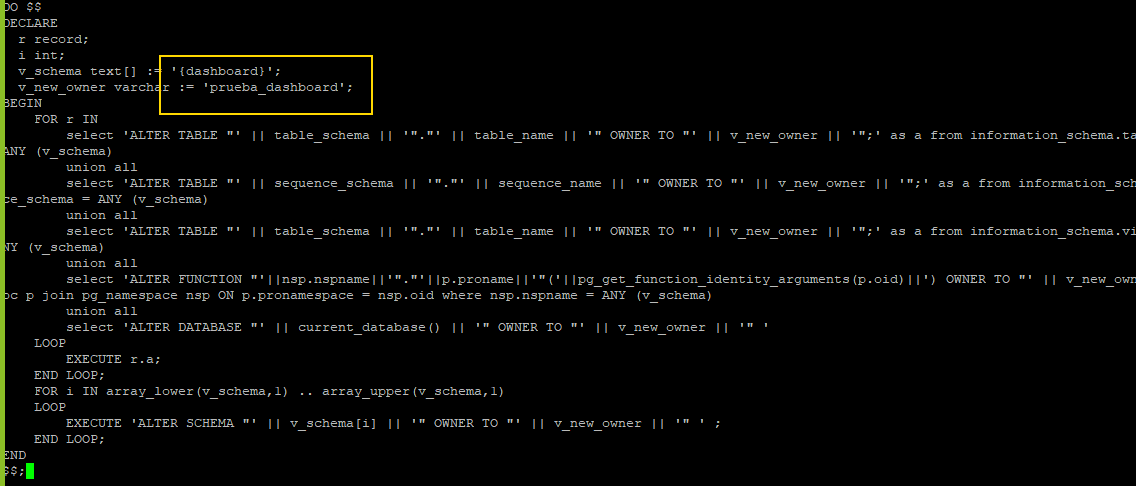
# 8. Configuración del usuario propietario para el esquema dashboard

# 

Editar y ejecutar el script “owner\_dashboard.sql” para actualizar el propietario del esquema dashboard

**Paso 1.** Editar el archivo “owner\_dashboard.sql” con el comando:

> vi owner\_dashboard.sql



Guardar y salir con el comando:

> :wq

**Paso 2**. Ejecutar el script actualizado con el comando:

> psql -U postgres edca < owner\_dashboard.sql

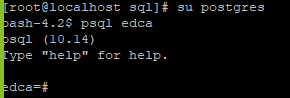


La ejecución de este script no debe de mostrar error.

**Paso 3.** Para verificar la creación correcta de la base de datos, ingresamos a postgres y a la base de datos, ejecutando los comandos:

> su postgres

> psql edca



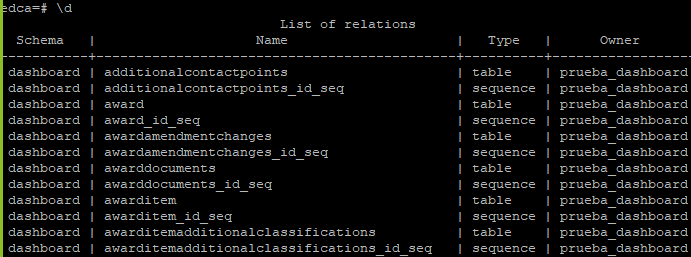
**Paso 4.** Cambiar el esquema de la base, ejecutando el comando:

> set search\_path to dashboard;



**Paso 5.** Consultar que se haya actualizado el propietario correspondiente, ejecutando el comando:

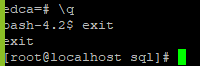
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

> exit



**Nota:** En caso de que el resultado de la ejecución de algún script haya fallado o no se entregará el resultado esperado, ponerse en contacto con el inai con el contacto contratacionesabiertas.inai.org.mx

# 9. Validación de conexión a base de datos

Paso 1. Validar que la conexión a base de datos sea correcta, ingresar a la ruta:

|  |
| --- |
| /var/www/html/contratacionesabiertas/captura/ |

Paso 2. Abrir el archivo llamado:

|  |
| --- |
| db.js |

Paso 3. La siguiente variable, debe estar configurada como se ve en la siguiente línea:

**var linkedStorage = (process.env.MONGODB\_PORT\_27017\_TCP\_ADDR || 'localhost' )**

Otra opción es setear la variable de ambiente "POSTGRES\_HOST", utilizando el

comando siguiente:

|  |
| --- |
| export POSTGRES\_HOST=localhost |

# 10. Asignación de variables para conexión a la API de la Plataforma Nacional de Transparencia

Es necesario validar que la conexión a la API de la Plataforma Nacional de Transparencia (PNT) sea correcta.

**Paso 1.** Ingresar a la carpeta “pnt”, ubicada en la ruta siguiente:

|  |
| --- |
| /var/www/html/contratacionesabiertas/captura/pnt/ |

**Paso 2.** Abrir el archivo llamado:

|  |
| --- |
| process-pnt.js |

**Paso 3.** En la variable EMAIL se configura el email de la unidad correspondiente y debe de configurarse como se ve en la siguiente línea:

**const EMAIL = process.env.PNT\_EMAIL || 'correo@institucional';**

**Paso 4.** Abrir el archivo llamado:

|  |
| --- |
| api-pnt.js |

**Paso 5.** En las siguientes variables se configuran las credenciales y debe de configurarse como se ve en las siguientes líneas:

**const** **URL = process.env.PNT\_URL || 'sitio\_webservices\_pnt:puerto';**

**const** **USER = process.env.PNT\_USER || 'correo@institucional';**

**const PASS = process.env.PNT\_PASS || 'contraseña';**

**Nota:** La funcionalidad de interconexión de la herramienta Contrataciones Abiertas contempla el envío de información de los formatos de la fracción XXVIII, del artículo 70, de la Ley General de Transparencia y Acceso a la Información Pública.

El correo electrónico, URL de PNT y contraseña son accesos que deberán gestionarse con el órgano garante local.

Es importante que, previo al envío de información a la PNT productiva, se considere un tiempo, como parte del proceso de implementación, para llevar a cabo pruebas de envío de información de Contrataciones Abiertas al sistema pre-productivo de la PNT, que existe para dicho propósito. De esta manera, se podrá validar la configuración de los formatos, el correcto envío de los datos, y en su caso, podrán realizarse ajustes a la configuración de los formatos para cumplir con las particularidades que existan como parte de la normativa local en la materia.

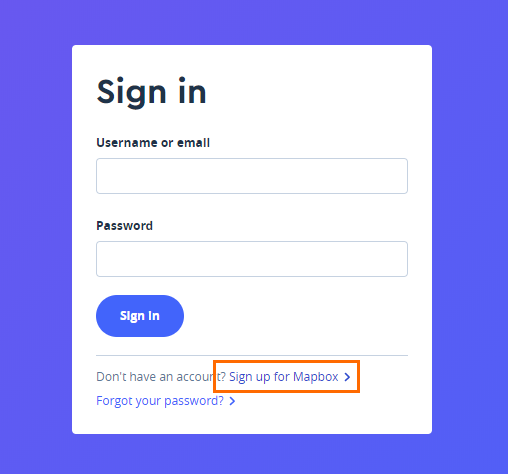
# 11. Asignación de variables MAPTOKEN

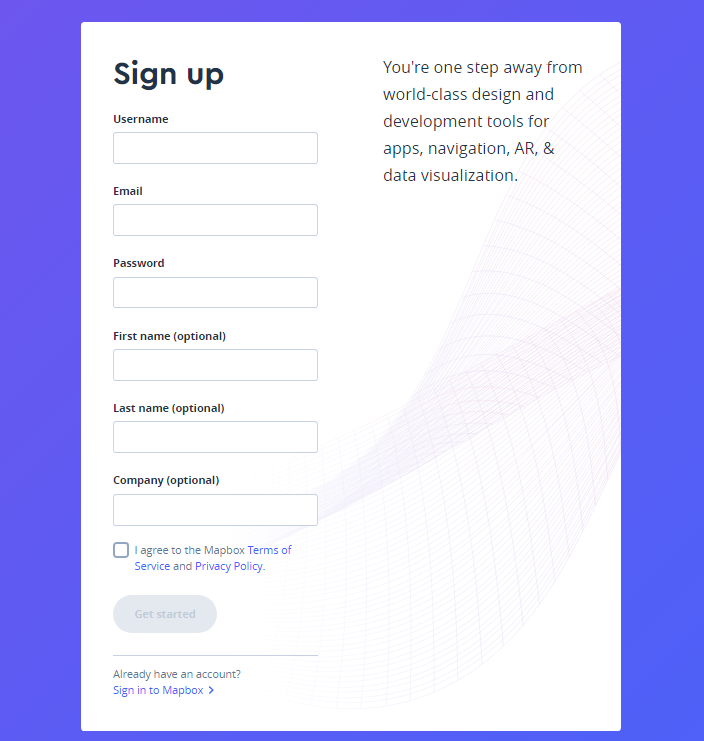
Para el servicio de Mapas es necesario configurar lo siguiente:

**Paso 1.** Para obtener el token es necesario ingresar a la siguiente URL:

<https://www.mapbox.com/signup/>

**Es necesario registrarse.**

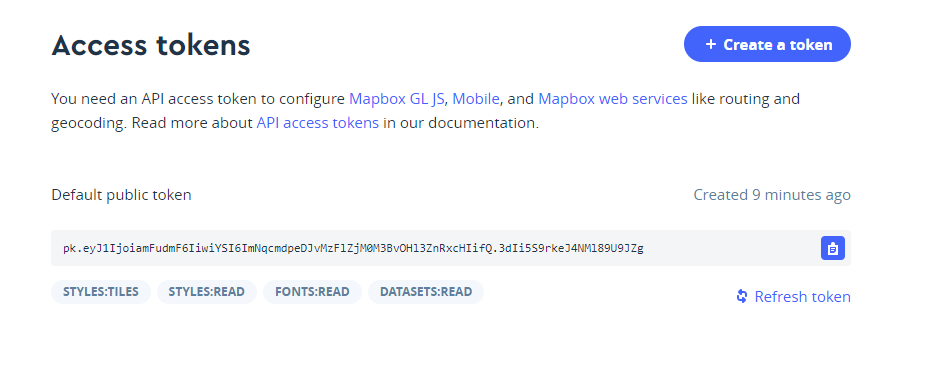




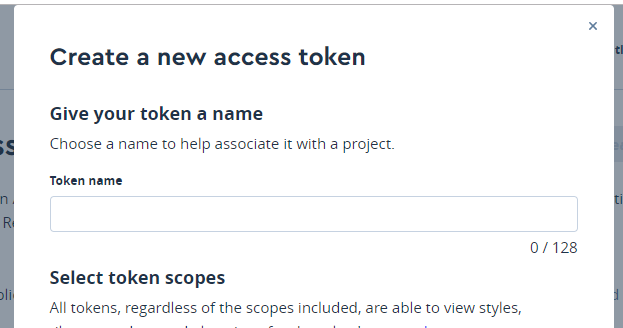
**Paso 2.** Una vez realizado el registro, ingresar a la página y le damos clic en la opción:



Debemos ver la siguiente pantalla:

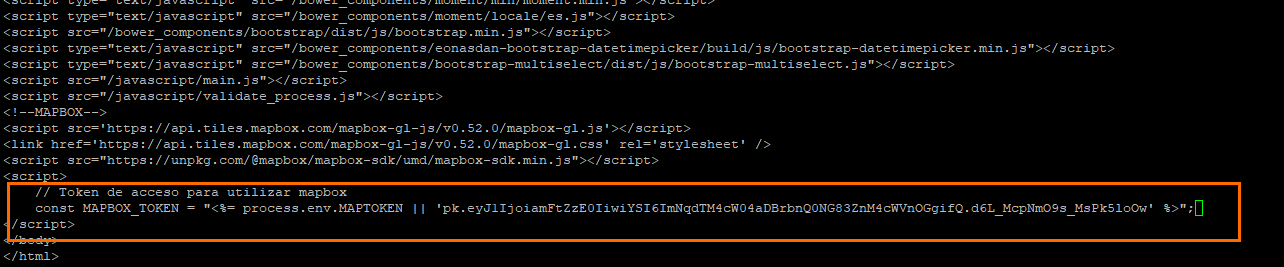


**Paso 3.** Copiar el valor del token que se encuentra ahí o crear uno asociándolo a un nombre



**Paso 4.** Ingresar a la carpeta /captura/views/ y abrir el archivo “main.ejs” para editar el valor MAPTOKEN, utilizando los siguientes comandos:

|  |
| --- |
| cd captura  cd views  vi main.ejs |



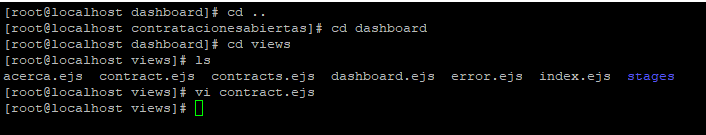
Sustituir con el valor del token

|  |  |
| --- | --- |
| **Variable** | **Valor** |
| MAPTOKEN | Token de Mapbox |

Guardar el cambio y salir.

**Paso 5.** Ingresar a la carpeta /dashboard/views/ y abrir el archivo “contract.ejs” para editar el valor MAPTOKEN, utilizando los siguientes comandos:

|  |
| --- |
| cd dashboard  cd views  vi contract.ejs |

****

Sustituir con el valor del token

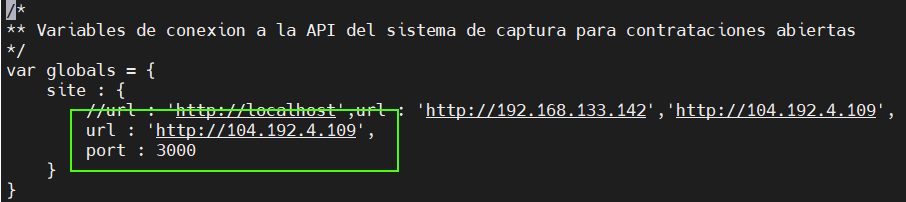
|  |  |
| --- | --- |
| **Variable** | **Valor** |
| MAPTOKEN | Token de Mapbox |



Guardar el cambio y salir.

**Paso 6.** Ingresar a la carpeta /contratacionesabiertas/dashboard/public/javascript/, editar el archivo “**common.js**” para asignar la IP correspondiente del servidor y el puerto asignado para el módulo Sistema de captura.

|  |
| --- |
| *vi common.js* |



Guardar el cambio y salir.

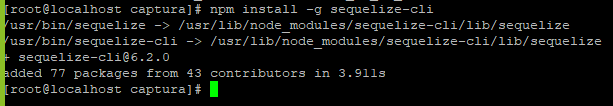
# 12. Instalación de Object Relational Mapping del módulo de Infraestructura

# 

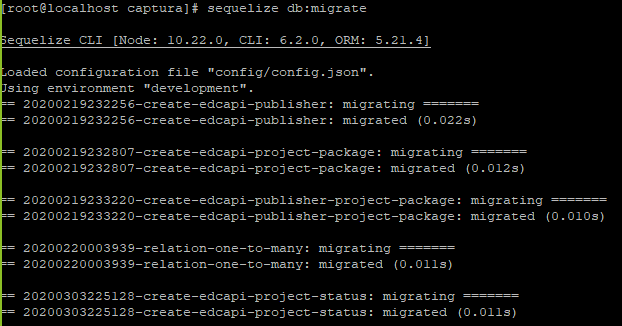
Instalación del Object Relational Mapping (ORM) para migración de tablas del módulo de Infraestructura

**Paso 1.** Ingresar a la carpeta captura/ y ejecutar los comandos:

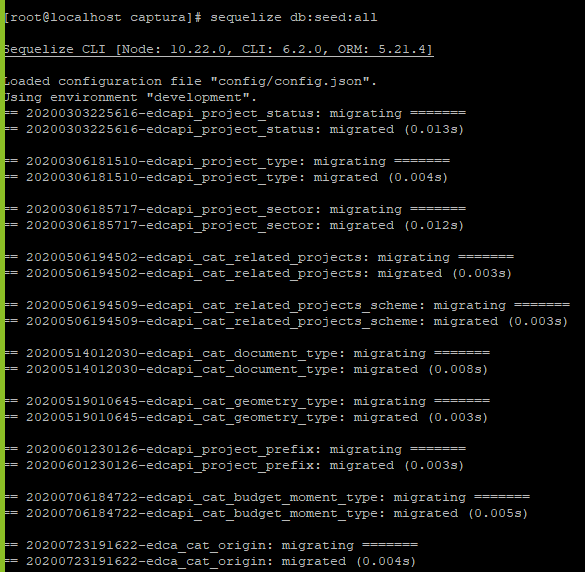
> npm install -g sequelize-cli



> sequelize db:migrate



> sequelize db:seed:all



La ejecución de estos comandos no debe de mostrar ningún error.

**Nota:** En caso de que se desee deshacer la migración de las tablas, se debe de ejecutar el siguiente comando:

> sequelize db:migrate:undo:all

# 13. Crear usuario del aplicativo

**Paso 1.** Para crear el usuario del aplicativo, ir a la carpeta “captura”, ubicada en la siguiente ruta: /var/www/html/contratacionesabiertas/ y ejecutar el script para la creación del usuario administrador del sistema de captura.

|  |
| --- |
| ./useradm add *username* |

Donde “username” se actualiza por su usuario administrador que deseen.

**Nota:** El archivo *useradm* debe tener permisos de ejecución, en caso de no tenerlos, ejecutar el siguiente comando dentro de la carpeta captura:

|  |
| --- |
| chmod u+x useradm |

**Nota:** Los campos deben cumplir las siguientes características:

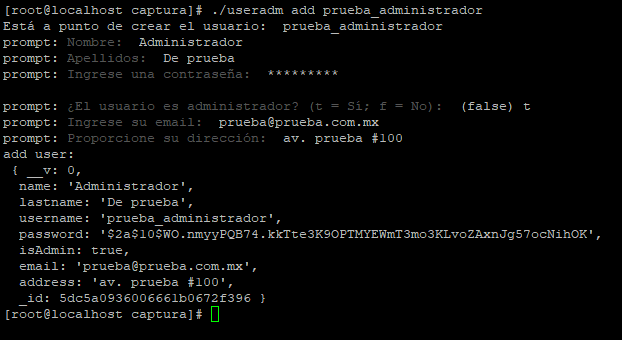
* Nombre: solo se permiten caracteres alfabéticos. Los nombres para los usuarios no se pueden repetir.
* Apellido: solo se permiten caracteres alfabéticos.
* Contraseña: se permiten caracteres alfanuméricos y espacios.
* ¿El usuario es un administrador?: este campo permite elegir el tipo de usuario que se pretende crear. Los tipos de usuario son: administrador y usuario del sistema.

1) El usuario administrador tiene acceso a funcionalidades de creación de usuarios desde la herramienta, puede capturar información, eliminar usuarios y asignar permisos sobre procesos de contratación.

2) El usuario del sistema solo tiene acceso para capturar información.

**Nota:** Para seleccionar si el usuario es administrador o no, se puede indicar “true o t” para verdadero o “false o f” para falso.

* Correo: debe presentar la estructura de un correo válido como la que se muestra a continuación; user@example.com
* Dirección: se permite ingresar la dirección completa incluyendo calle, número, municipio, etc.



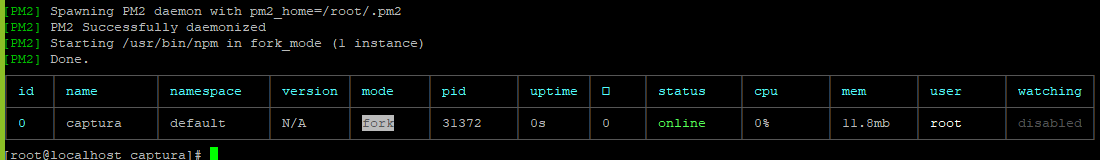
# 14. Iniciar los servicios de los módulos

**Paso 1.** Ingresar a la carpeta captura, utilizando el siguiente comando:

|  |
| --- |
| cd captura |

**Paso 2.** Iniciar el servicio del módulo “Sistema de captura”, ejecutando el siguiente comando:

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

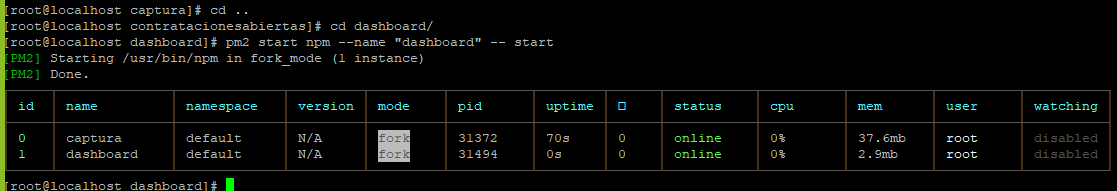


**Paso 3.** Salir de la carpeta captura e ingresar a la carpeta dashboard, ejecutando los siguientes comandos:

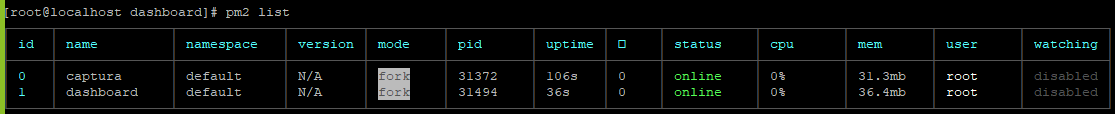
|  |
| --- |
| cd ..  cd dashboard |

**Paso 4.** Iniciar el servicio del módulo “Dashboard - visualizador”, ejecutando el siguiente comando:

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



**Paso 5.** Ejecutar el comando pm2 list para ver el estado de los servicios y que estén correctos.



# 15. Acceder a los módulos

# 

## A. Módulo sistema de captura

Ingresar a la siguiente dirección:

|  |
| --- |
| http://url\_del\_servidor:3000 |



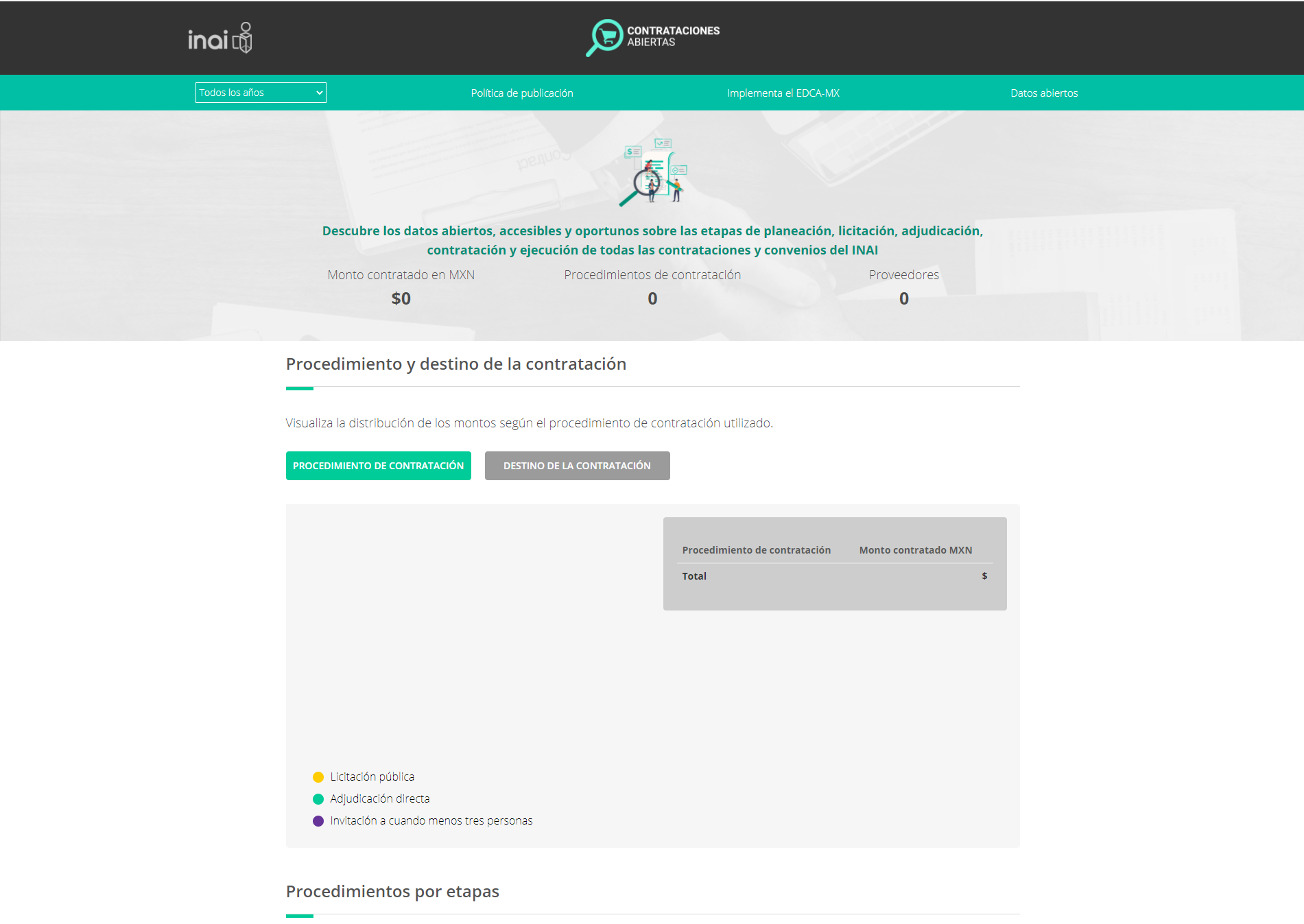
Ingresar con el usuario y contraseña creados en el paso 13



## B. Módulo de visualización (dashboard)

Ingresar a la siguiente dirección:

|  |
| --- |
| http://url\_del\_servidor:4000/contratacionesabiertas/contratos |



# 16. Configuración de Google Analytics para ambiente productivo

Para comenzar a registrar el número de visitas al sitio, es necesario utilizar herramientas como Google Analytics, para configurar es necesario realizar los siguientes pasos:

1. Crear una cuenta en Google
2. Crear una propiedad de seguimiento exclusivo de Google Analytics para recoger los datos de la herramienta Contrataciones Abiertas.
3. Se generará una etiqueta global de sitio web, es necesario copiar esa etiqueta en cada página web de la herramienta justo después de la etiqueta de apertura <head>.

Para mayor información: <https://support.google.com/analytics/answer/1008080?hl=es>

# Anexo 1. Errores comunes.

1. Error en la creación de una nueva contratación.

Causa. Omisión de la ejecución de las sentencias del paso “*5.1 Migración de las tablas para el del módulo de infraestructura”*

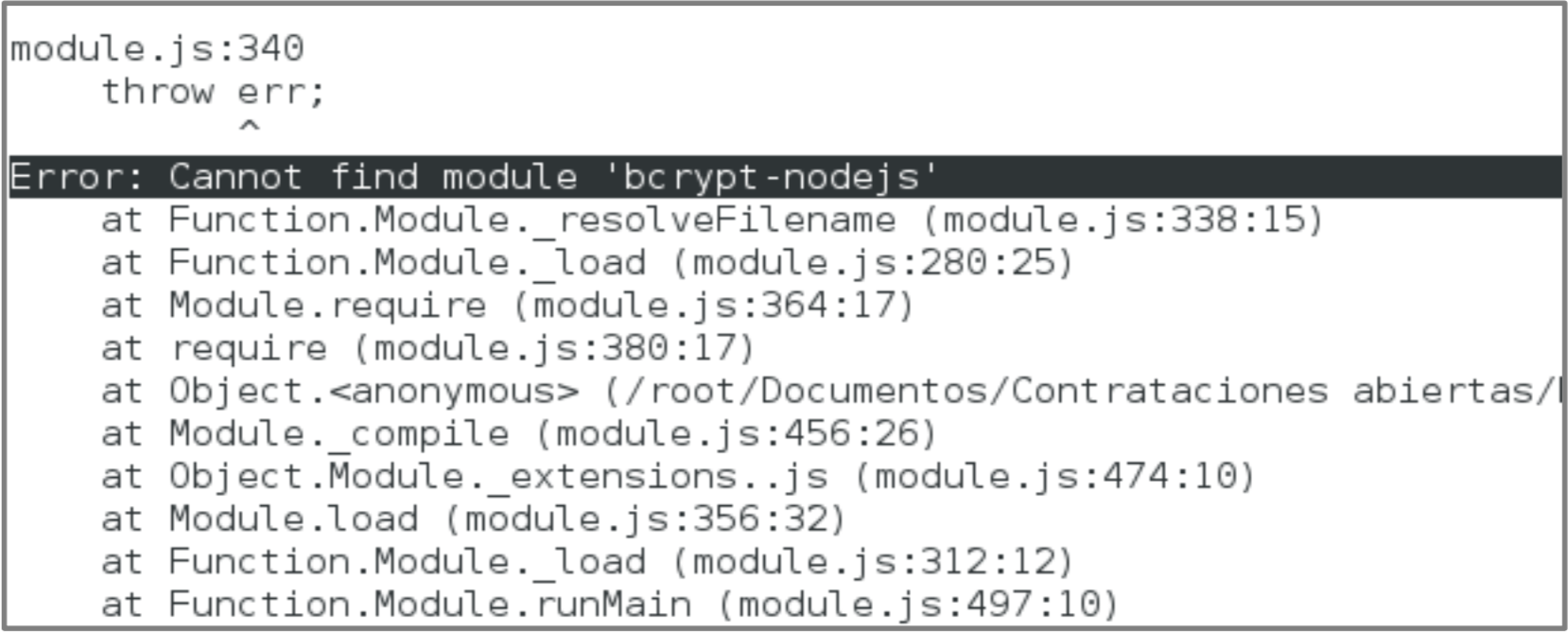


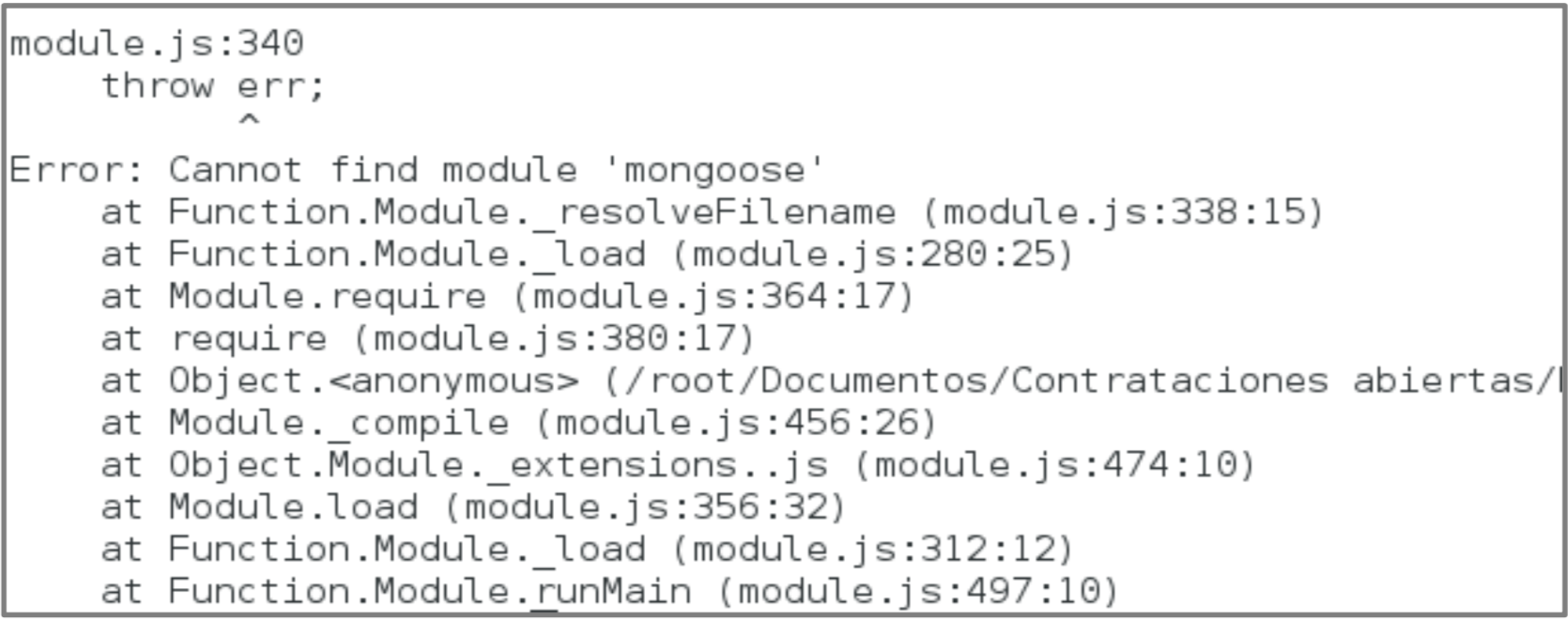
Solución. Ejecutar dentro de la carpeta captura, las sentencias:

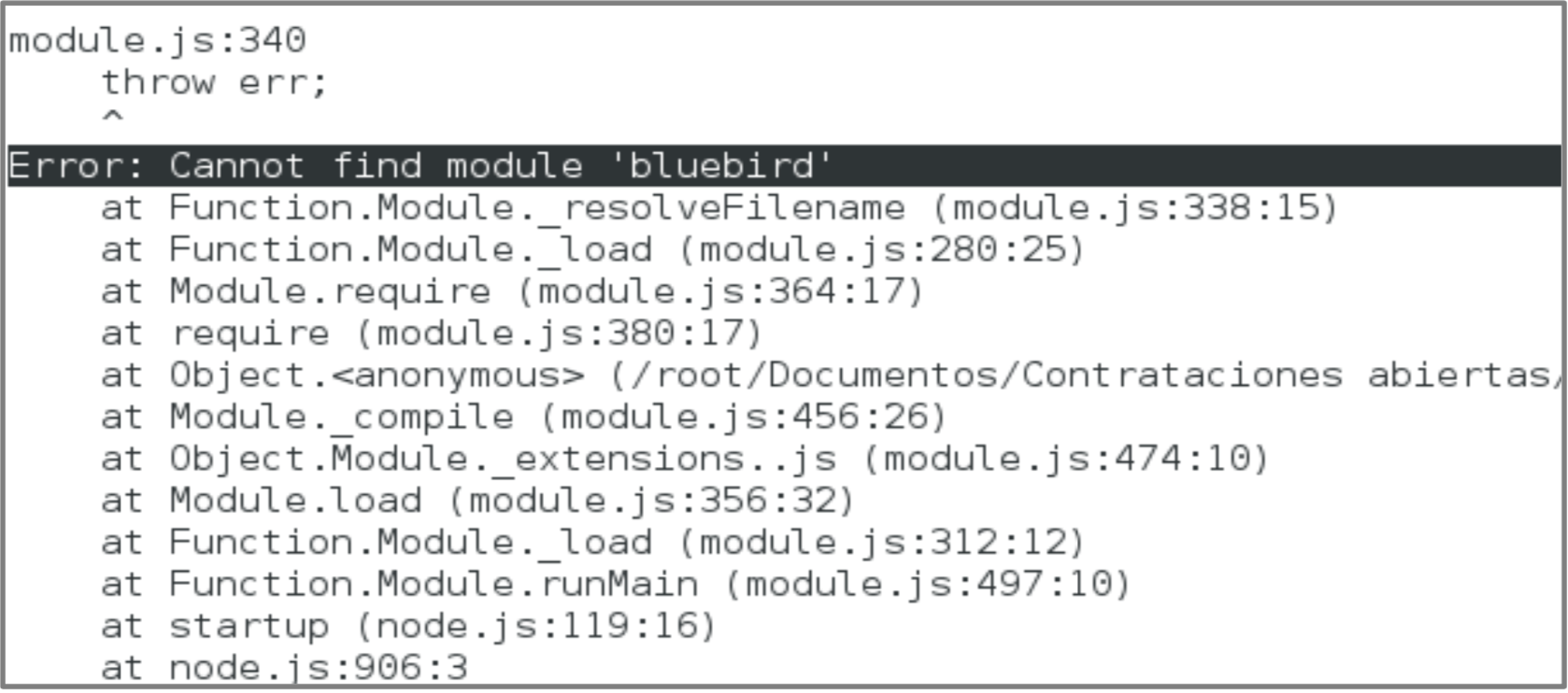
*> sequelize db:migrate*

*> sequelize db:seed:all*

1. Si al crear un usuario salen los siguientes errores:







Se deben instalar los siguientes módulos:

**Bycrypt-nodejs**

|  |
| --- |
| npm install bycrypt-nodejs |

**Bluebird**

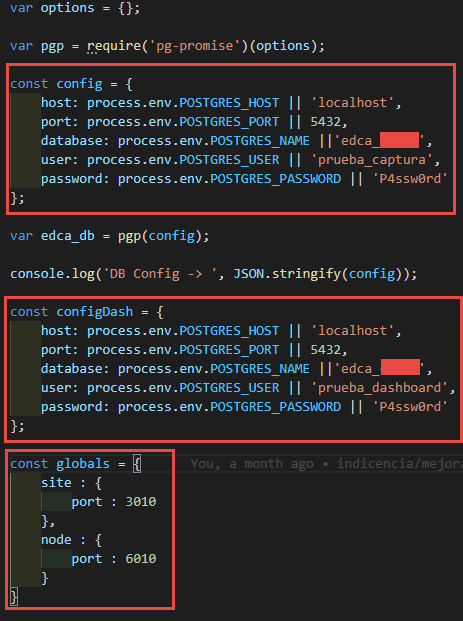
|  |
| --- |
| npm install bluebird |

**Mongoose**

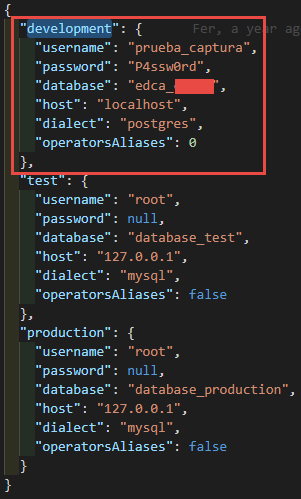
|  |
| --- |
| npm install mongoose |

# Anexo 2. Configuraciones Generales para más de una instancia en el mismo servidor.

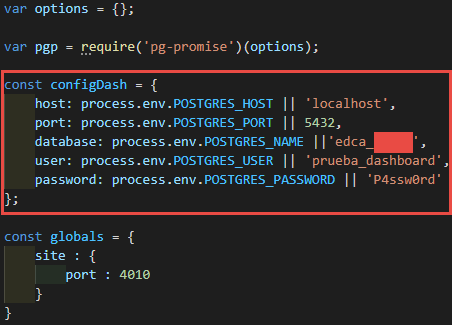
1. Configuración de cadenas de conexión para el “*sistema de captura*”:
   1. Modificar el archivo “*db\_conf.js*” de la ruta “*/captura/db\_conf.js*” en las variables de conexión “*host*”, “*database”, “user” y “password”* en “const *config”* y “const *configDash”*.  
        
      Modificar “const globals” con las variables correspondientes a los puertos.



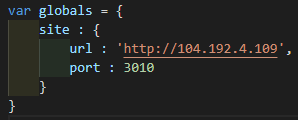
* 1. La configuración de los puertos por default es el número 3000 y 6000, estos se deberán cambiar si se desea tener otra instancia de la aplicación. Se puede ocupar el número de puerto 3001 hasta el 3999 y del 6001 hasta el 6999 respectivamente para cada instancia.
  2. Entrar al archivo “*config.json*” de la ruta “*/captura/config/config.json*” en el objeto “*development*” modificar las variables de conexión “*host*”, “*database”, “username” y “password”*.



1. Configuración de cadenas de conexión para el sistema “*dashboard*”:  
   1. Entrar en el archivo “*dash\_config.js*” de la ruta “*/dashboard/dash\_config.js*” y modificar las variables de conexión “*host*”, “*database”, “user”, “password”* en “const *configDash”*.
   2. Modificar “const globals” con las variables correspondientes al puerto.



* 1. La configuración del puerto es por default el número 4000, esto se deberán cambiar si se desea tener otra instancia de la aplicación. Se puede ocupar el número del puerto desde el 4001 hasta el 4999 uno por instancia.
  2. Entrar en el archivo “*common.js*” de la ruta “*/dashboard/javascript/common.js*” modificar la variable del puerto y de la ip correspondiente a la instancia instalada previamente en el archivo “*db\_conf.js*” de la ruta “*/captura/db\_conf.js*”.



**NOTA:** Es necesario tener las reglas de firewall pertinentes en el servidor para el uso de estos puertos.

**NOTA 2:** Es necesario para que funcionen correctamente las múltiples instancias especificar una base de datos única para cada instancia, como por ejemplo “*edca\_gen*”.