**Contenido**

[1. Arquitectura](#_gjdgxs) 3

[2. Creación y configuración 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. Editar y ejecutar el script “edca.sql” para crear la estructura en la base edca](#_2et92p0) 11

[6. Editar y ejecutar el script “owner\_public.sql” para actualizar el propietario del esquema public, con el usuario creado previamente](#_oktvabjhfpa0) 15

[7. Editar y ejecutar el script “dashboard.sql” para la creación del esquema dashboard](#_simy1dsmay53) 17

[8. Editar y ejecutar el script “owner\_dashboard.sql” para actualizar el propietario del esquema dashboard](#_si7uotw8zk5l) 20

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

[10. Asignación de variables para conexión a la API de PNT](#_1t3h5sf) 22

[11. Asignación de variables MAPTOKEN](#_4d34og8) 23

[12. Instalación de ORM, para migración de tablas del módulo de Infraestructura](#_17dp8vu) 27

[13. Crear usuario del aplicativo](#_86zl5romdevr) 29

[14. Iniciar los servicios de los módulos](#_3rdcrjn) 32

[15. Acceder a los módulos](#_26in1rg) 33

[A. Módulo sistema de captura](#_a1i7c4kumya) 33

[B. Módulo de visualización (dashboard)](#_35nkun2) 34

[Anexo. Errores comunes.](#_qatxi9pcyck4) 35

# 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.-** Adicionalmente se requiere que el servidor tenga salida a internet, esto por la descarga de dependencias que existen para el funcionamiento de las gráficas.

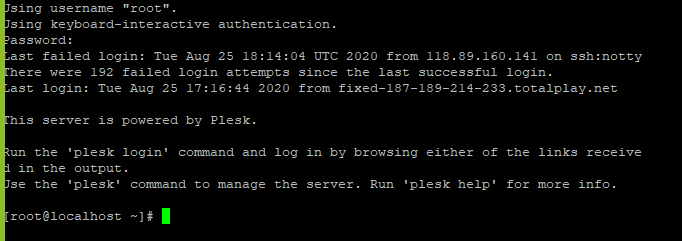
**2.**- El servidor debe permitir el acceso a través de los puertos 3000 y 4000.

**3.**- Se necesita tener instalado el software: **git** y **pm2**.

**4.-** Postgresql debe de configurar los métodos de conexión **local** e **IPv4** como **trust**

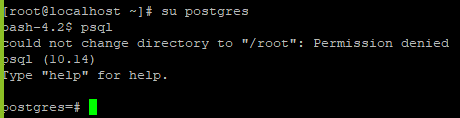
# 2. Creación y configuración de la base de datos

**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; |



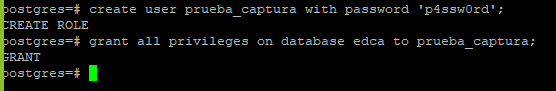
**Nota.** Los nombres de “usuario” y la “contraseña” para esta configuración son de ejemplo.

**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*; |

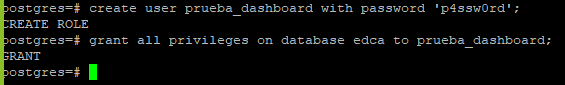


**Paso 5.** Crear el usuario que se asignará como propietario del esquema dashboard de la base y otorgarle todos los permisos sobre la base.

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

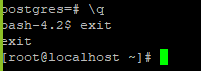
Se deben ejecutar los siguientes comandos:

|  |
| --- |
| 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:

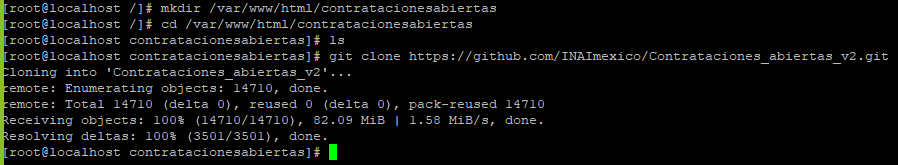
|  |
| --- |
| mkdir /var/www/html/contratacionesabiertas |



**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\_v2.git |





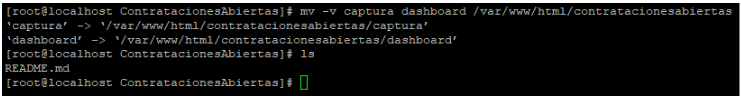
**Paso 3.** Acceder a la ruta que contiene los directorios *captura* y *dashboard,* ejecutando el siguiente comando:

|  |
| --- |
| cd Contrataciones\_abiertas\_v2/ContratacionesAbiertas |



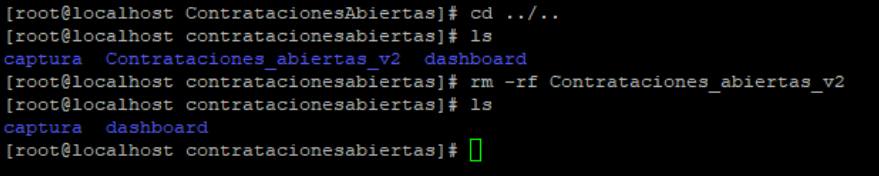
**Paso 4.** Mover los directorios *captura* y *dashboard* a la carpeta raíz, ejecutando el siguiente comando:

|  |
| --- |
| 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.

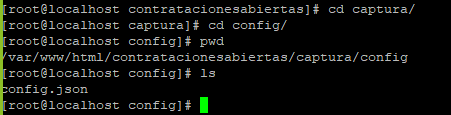
|  |
| --- |
| cd ../..  rm -rf Contrataciones\_abiertas\_v2 |

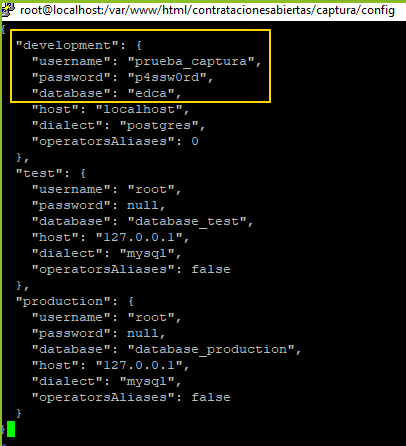
**

# 

# 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.



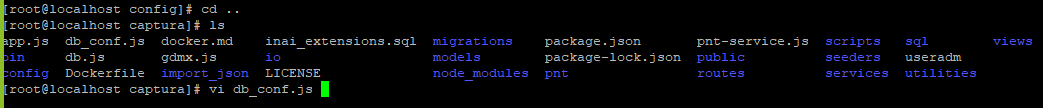


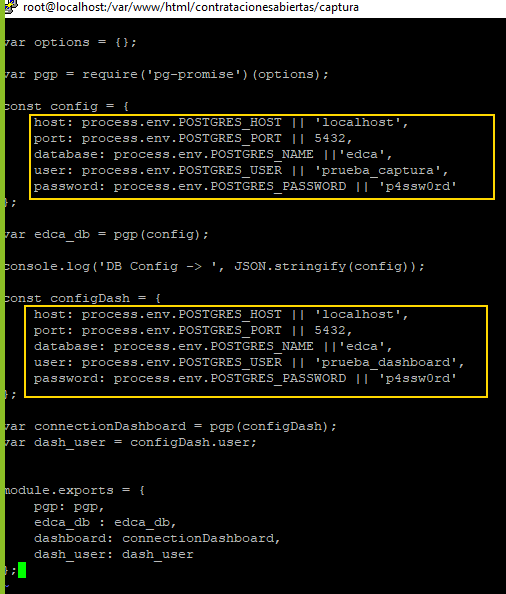
Salimos de la edición del archivo con el siguiente comando

*> :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, 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; ejecutando los comandos siguientes:

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



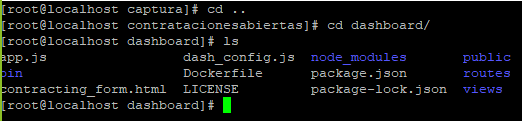


Actualizar, y guardar los cambios con el siguiente comando:

> *:wq*

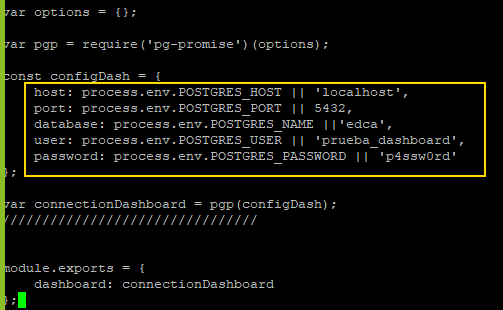
**Paso 3.** Salir de la carpeta /carpeta/sql/ e ingresar a la carpeta /dashboard/, ejecutar los siguientes comandos:

|  |
| --- |
| 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; ejecutar el comando siguiente:

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



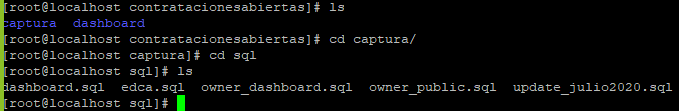
Actualizar, y guardar los cambios con el comando:

> *:wq*

# 5. 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:

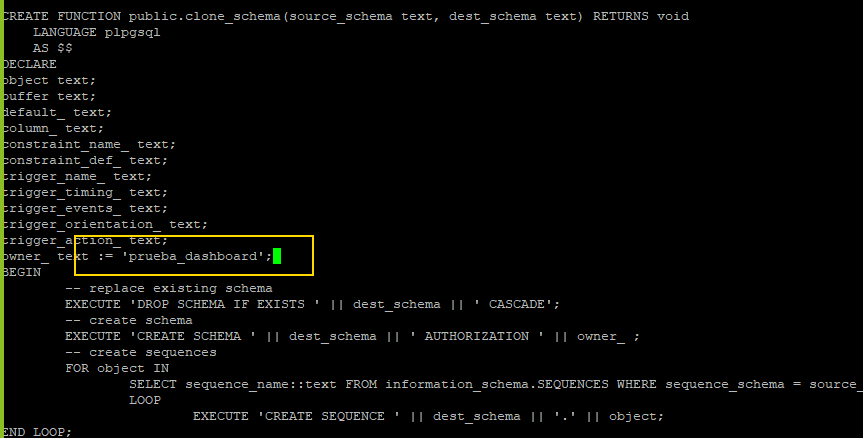
|  |
| --- |
| cd captura  cd sql |



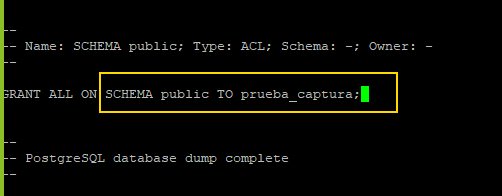
**Paso 2.** Editar el archivo para actualizar el nombre del usuario del esquema public y del esquema dashboard respectivamente, con los usuarios creados en el paso: **“2. Creación y configuración de la base de datos.”**

> vi edca.sql

Actualizar el nombre del usuario del esquema dashboard

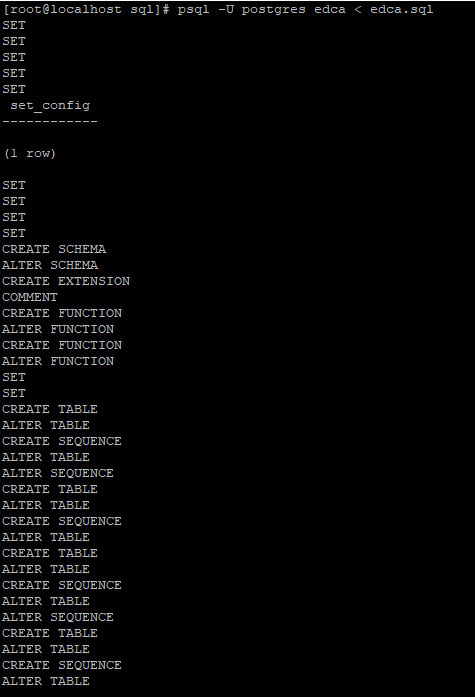


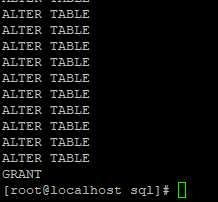
Actualizar el nombre del usuario del esquema public, que se encuentra en la parte final del documento.



**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* |



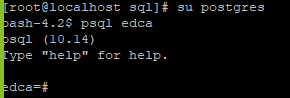


La ejecución de este script no debe de mostrar error.

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

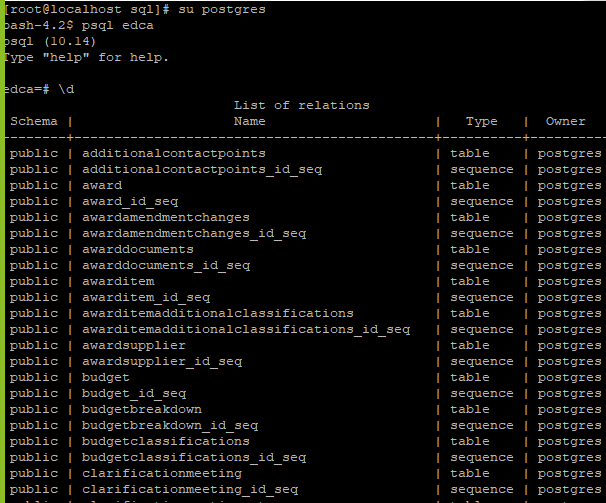
> su postgres

> psql edca



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

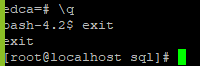
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

> exit

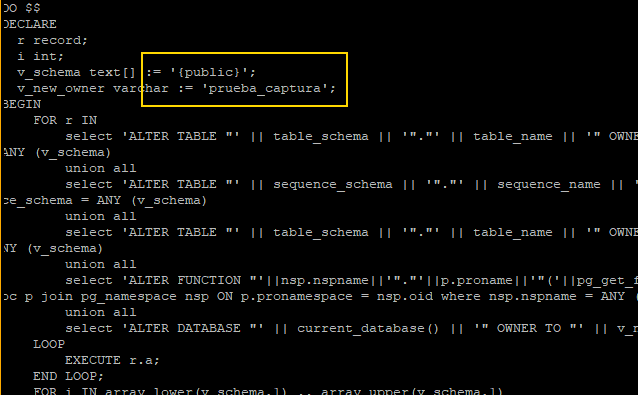


# 

# 6. 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

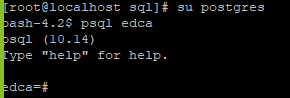


La ejecución de este script no debe de mostrar error.

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

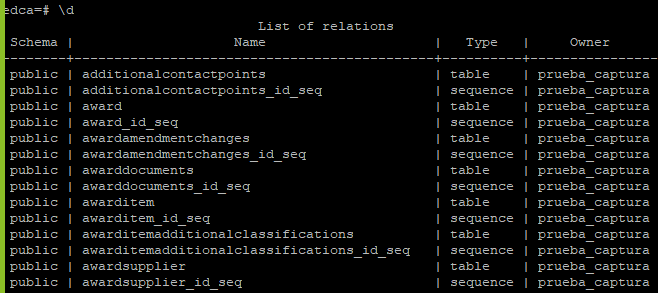
> su postgres

> psql edca



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

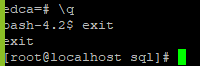
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

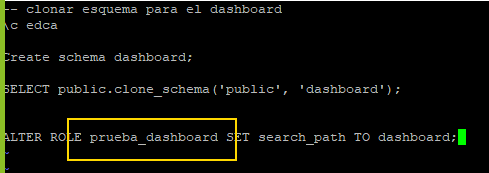
> exit



# 7. 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

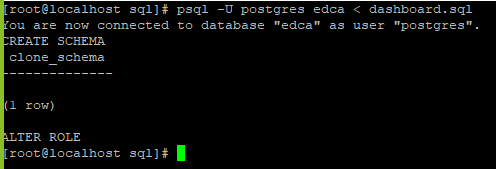


Guardar y salir con el comando:

> :wq

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

> psql -U postgres edca < dashboard.sql

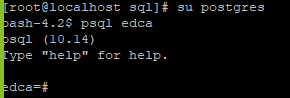


La ejecución de este script no debe de mostrar error.

**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



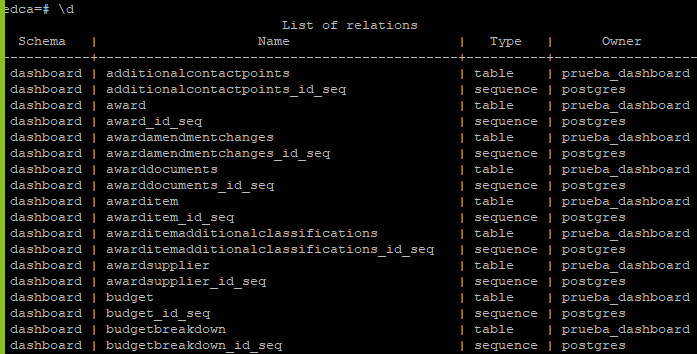
**Paso 4.** Cambiar el esquema de la base, ejecutando el comando:

> set search\_path to dashboard;



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

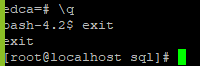
> \d



Para salir de postgres, ejecutamos los comandos:

> \q

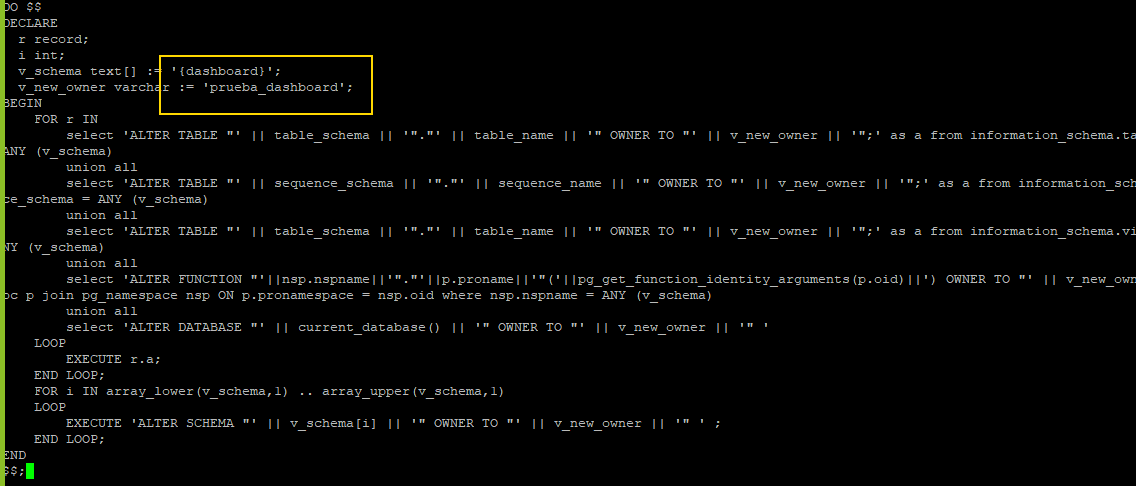
> exit



# 8. 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

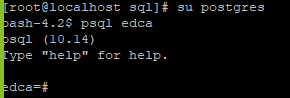


La ejecución de este script no debe de mostrar error.

**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



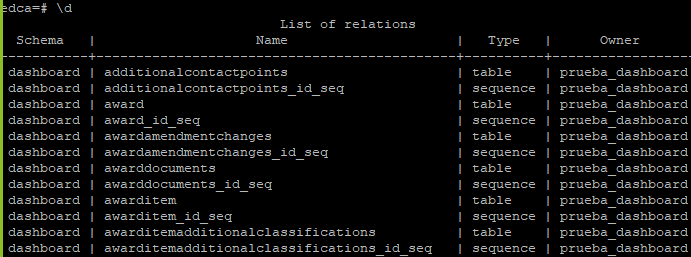
**Paso 4.** Cambiar el esquema de la base, ejecutando el comando:

> set search\_path to dashboard;



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

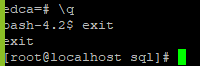
> \d



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 PNT

Es necesario validar que la conexión a la API de 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 conexión está configurada para enviar información del artículo 70, fracción 28 para los formatos A y B de la ley general de transparencia. El correo electrónico, URL de PNT y contraseña son accesos que deberán gestionarse con el órgano garante local.

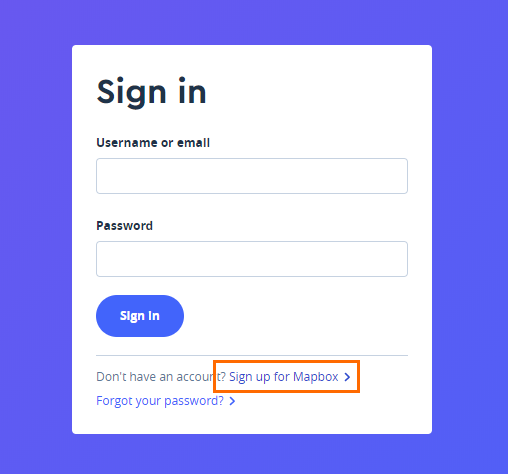
# 11. Asignación de variables MAPTOKEN

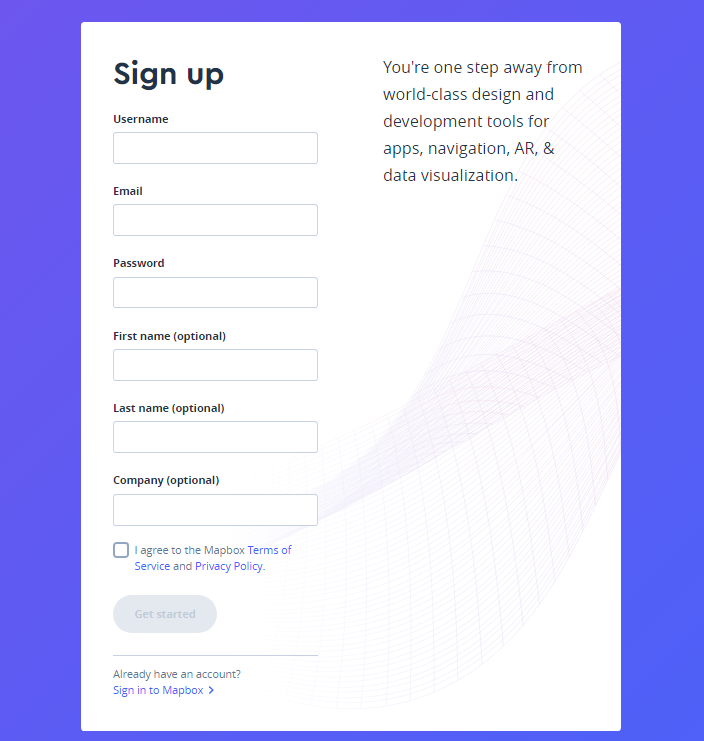
Para el servicio de Mapas es necesario configurar lo siguiente:

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

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

**Es necesario registrarse.**

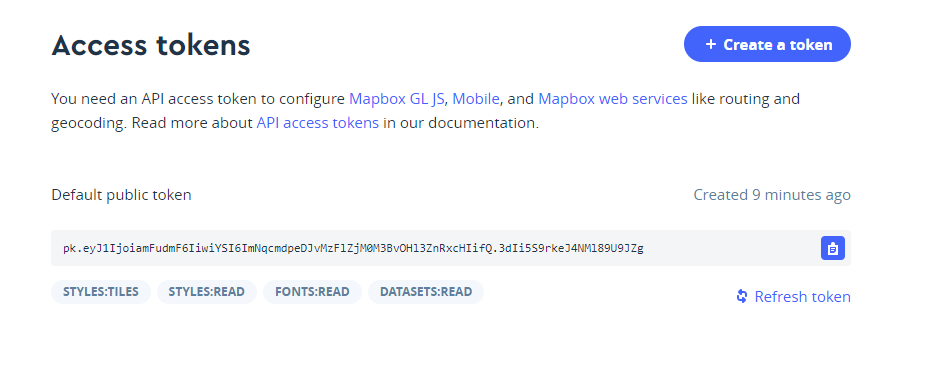




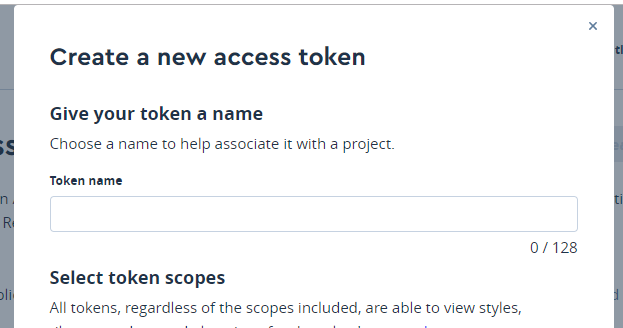
**Paso 2.** Una vez realizado el registro, ingresar a la página y le damos clic en la opción:



Debemos ver la siguiente pantalla:

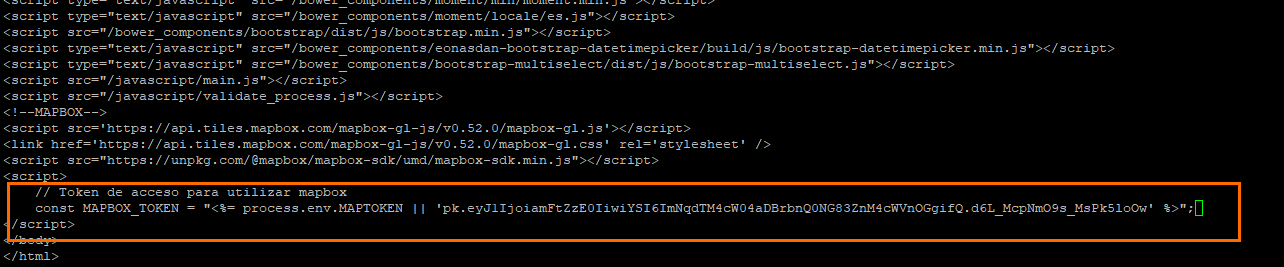


**Paso 3.** Copiar el valor del token que se encuentra ahí o crear uno asociándolo a un nombre



**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 |



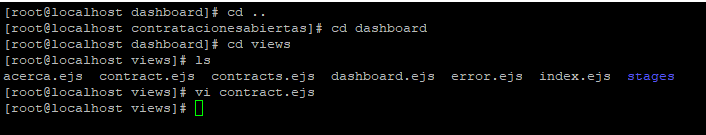
Sustituir con el valor del token

|  |  |
| --- | --- |
| **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.** Abrir el archivo “contracts.ejs” para editar el valor CAPTURE\_URL, utilizando el siguiente comando:

|  |
| --- |
| vi contracts.ejs |

****

Sustituir con la IP del servidor del módulo de captura, debe tener el puerto 3000.

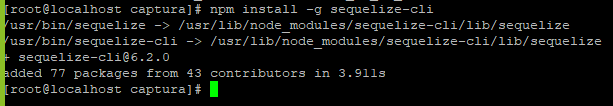
|  |  |
| --- | --- |
| **Variable** | **Valor** |
| CAPTURE\_URL | ip\_servidor:3000 |

Guardar el cambio y salir.

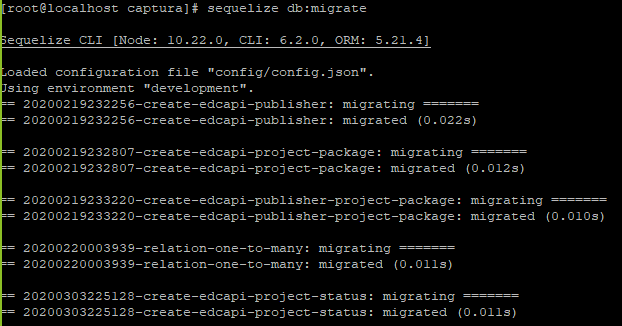
# 12. Instalación de ORM, para migración de tablas del módulo de Infraestructura

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

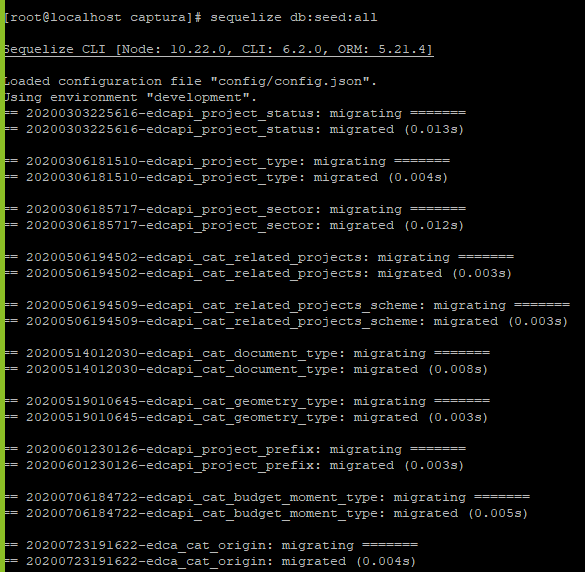
> npm install -g sequelize-cli



> 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 |

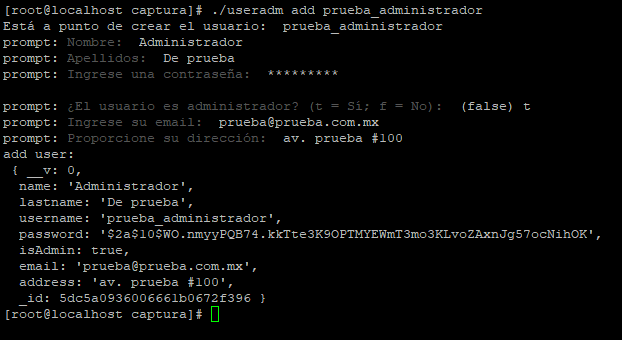
**Nota:** Los campos deben cumplir las siguientes características:

* 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.

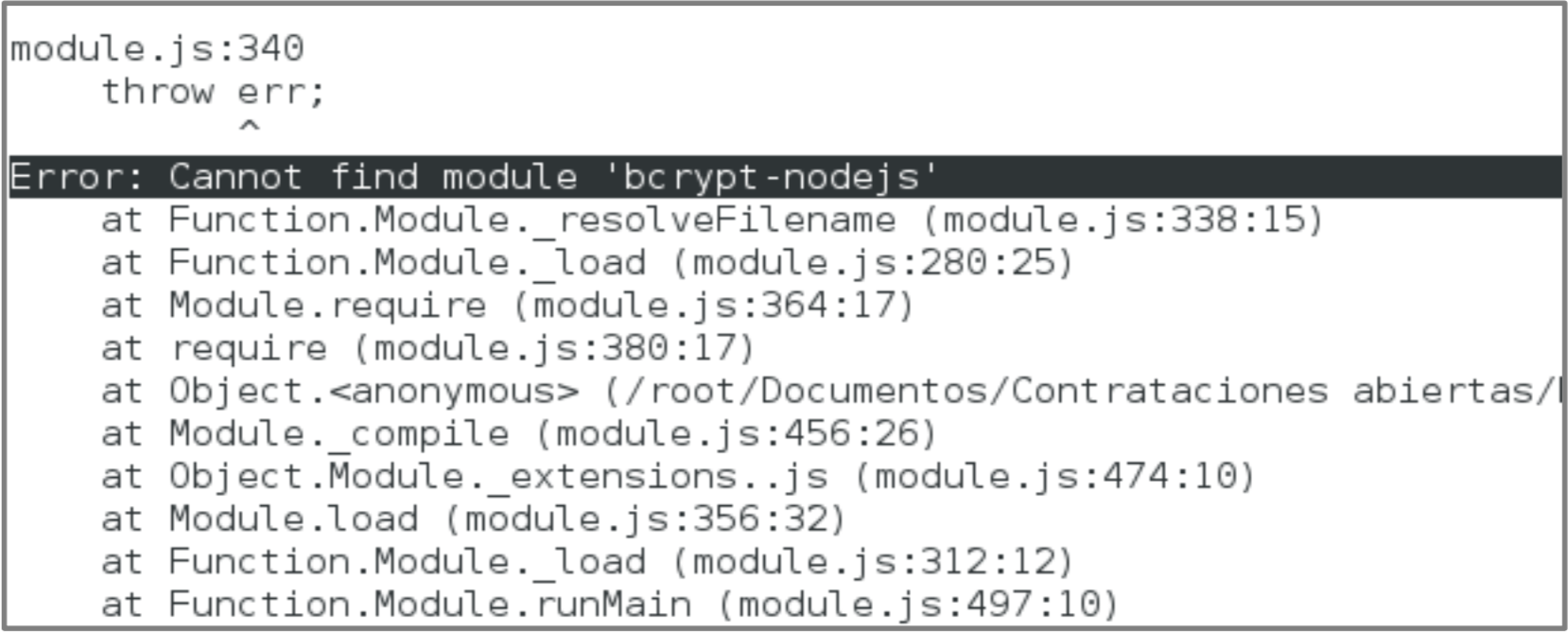
1) El usuario administrador tiene acceso a funcionalidades de creación de usuarios desde la herramienta, puede capturar información, eliminar usuarios y asignar permisos sobre procesos de contratación.

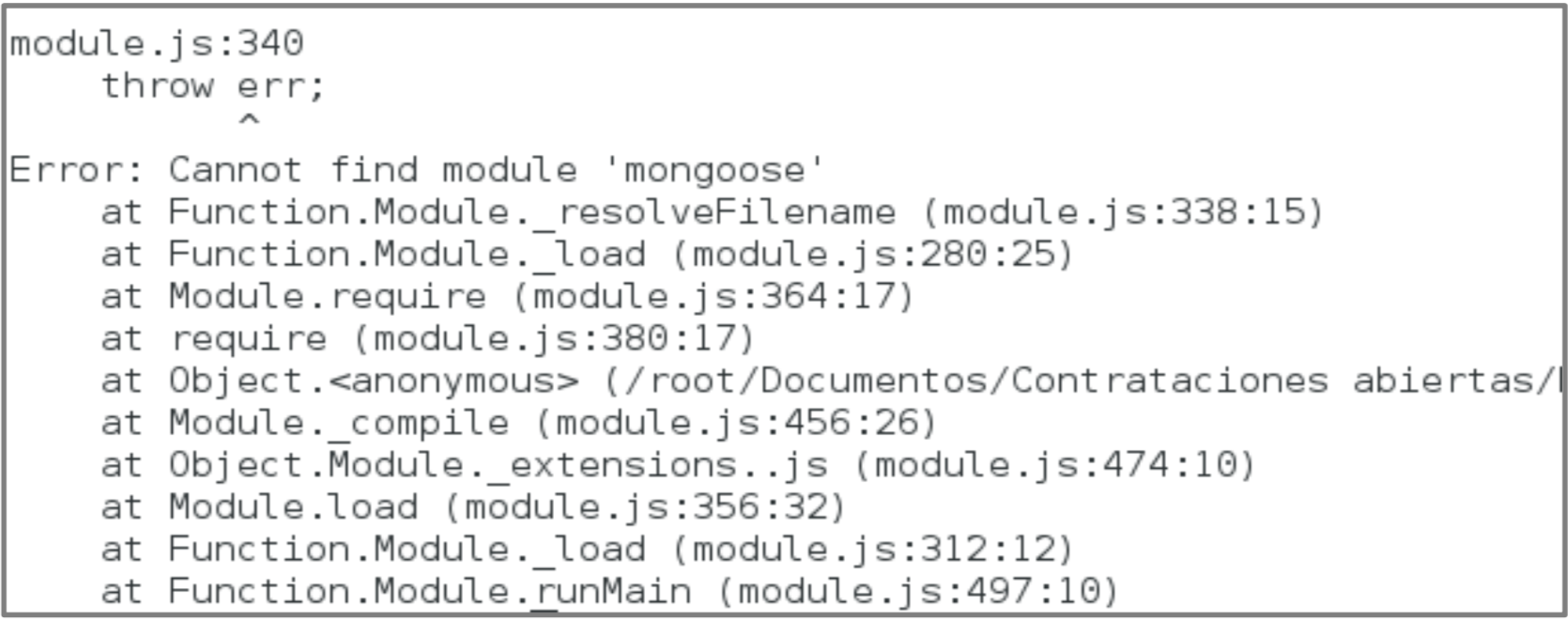
2) El usuario del sistema solo tiene acceso para capturar información.

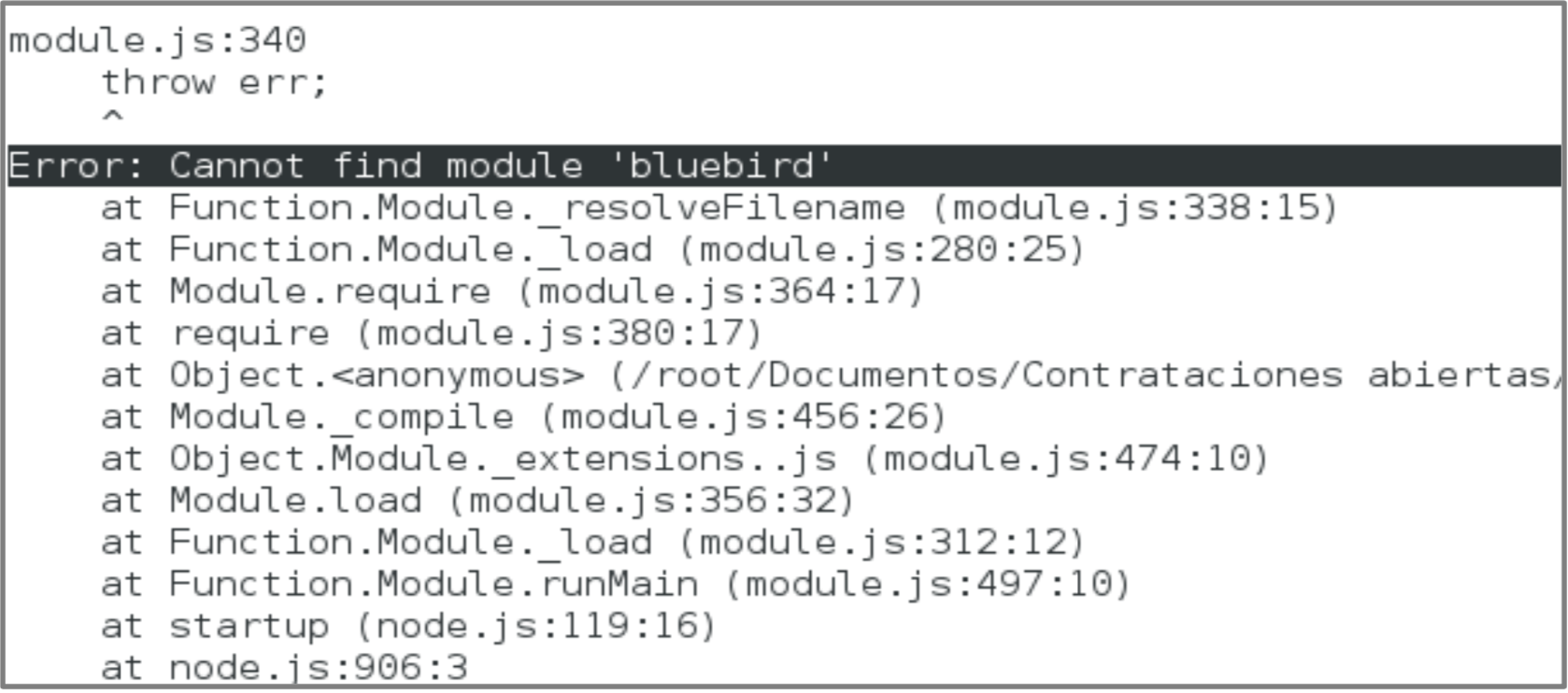
* Correo: debe presentar la estructura de un correo válido como la que se muestra a continuación; user@example.com
* Dirección: se permite ingresar la dirección completa incluyendo calle, número, municipio, etc.



**Importante:** 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 |

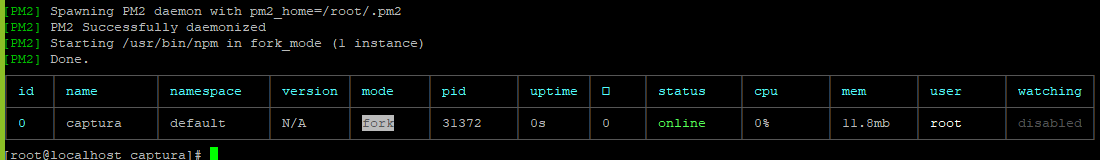
# 14. Iniciar los servicios de los módulos

**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 |

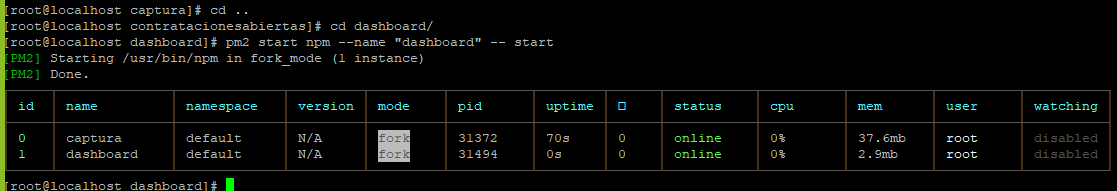


**Paso 3.** Salir de la carpeta captura e ingresar a la carpeta dashboard, ejecutando los siguientes comandos:

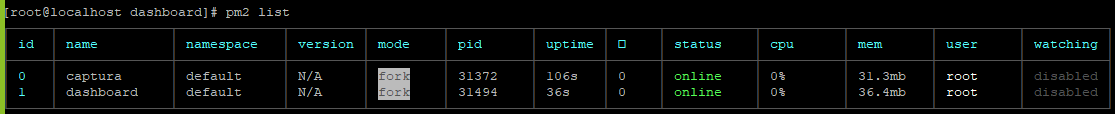
|  |
| --- |
| cd ..  cd dashboard |

**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 |



# Anexo. 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*