MANUAL DE ACTUALIZACIÓN DE LA HERRAMIENTA

**CONTRATACIONES**

**ABIERTAS**

Contenido

[1.](#_1ksv4uv) Arquitectura. 3

[2.](#_44sinio) Preparación de la actualización. 3

[2.1](#_2jxsxqh) Conectarse al servidor 3

[2.2](#_z337ya) Realizar un backup de la base de datos edca. 3

[2.3](#_3j2qqm3) Detener los servicios de “Dashboard y Captura”. 4

[2.4](#_1y810tw) Se debe realizar “backup” de código de Dashboard y Captura 4

[2.5](#_4i7ojhp) Descargar el archivo del código. 5

[3.](#_2xcytpi) Ejecución de la actualización 6

[3.1](#_1ci93xb) Actualización de la base de datos 6

[3.2](#_3whwml4) Creación de los nuevos usuarios 9

[3.3](#_2bn6wsx) Actualizar usuario en la base de datos 11

[3.4](#_qsh70q) Importar el dump a la base de datos edca 13

[3.5](#_3as4poj) Actualización y ejecución de scripts 14

[4.](#_1pxezwc) Compilación y ejecución. 15

[4.1](#_49x2ik5) Error al ejecutar la instalación de Bower. 18

[4.2](#_2p2csry) Validación de conexión a base de datos. 19

[4.3](#_147n2zr) Asignación de variables para conexión a la API de PNT. 20

[4.4 Asignación de variables MAPTOKEN. 22](#_3o7alnk)

[4.5](#_23ckvvd) Iniciar servicio del “Sistema de captura” 25

[4.6](#_ihv636) Actualización de dashboard. 25

[4.7](#_32hioqz) Iniciar servicio del “dashboard - visualizador” 28

[5.](#_1hmsyys) Acceder a los módulos. 28

[5.1.](#_41mghml) Módulo de captura. 28

[5.2.](#_2grqrue) Módulo de visualización (dashboard). 30

# Arquitectura.

El servidor requiere de los siguientes requerimientos de hardware:

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

El servidor requiere de los siguientes requerimientos de software:

|  |  |
| --- | --- |
| Sistema Operativo | RedHat 7 |
| Base de datos relacional | PostgreSQL 10 |
| Base de datos de documentos | MongoDB v3 |
| Lenguaje de programación | Nodejs v9 |
| Servidor web | NGINX |

**Nota:**

**1.-** Adicionalmente se requiere que el servidor tenga salida a internet, esto por la descarga de dependencias que existen.

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

**3.**- Se necesita tener instalado git.

# Preparación de la actualización.

Para realizar la actualización del código, se deben de seguir los siguientes pasos:

# Conectarse al servidor

**Paso 1.** Ingresar al servidor con el usuario root.

# Realizar un backup de la base de datos edca.

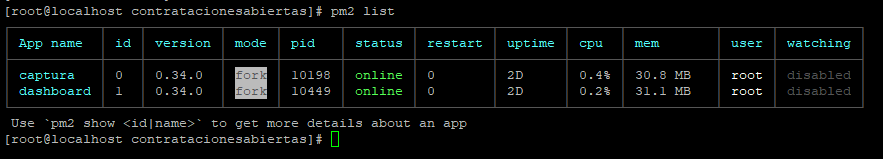
**Paso 1.** Ubicarse en la carpeta donde se desee guardar el archivo sql de respaldo. Ejecutar el siguiente comando para exportar la base de datos completa edca.

|  |
| --- |
| pg\_dump -U postgres edca > respaldo.sql |

# Detener los servicios de “Dashboard y Captura”.

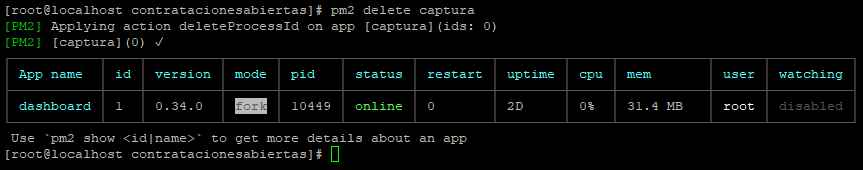
**Paso 1.** Ejecutar el siguiente comando para listar los servicios que se encuentran iniciados:

|  |
| --- |
| pm2 list |

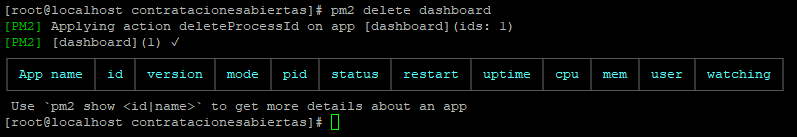


**Paso 2.** Borrar los servicios iniciados ejecutando un comando por servicio iniciado:

|  |
| --- |
| pm2 delete captura |



|  |
| --- |
| pm2 delete dashboard |



# Se debe realizar “backup” de código de Dashboard y Captura

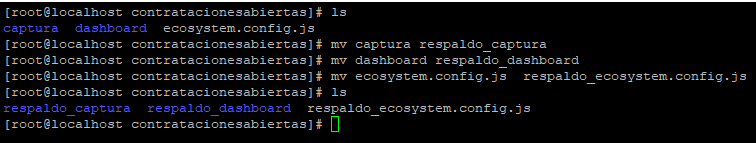
**Paso 1.** Posicionarnos en la ruta en donde se encuentra instalado el código, ejecutando el siguiente comando:

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

**Paso 2.** Ejecutar las siguientes instrucciones:

|  |
| --- |
| mv captura <Nombre del folder backup captura>  mv dashboard <Nombre del folder backup dashboard>  mv ecosystem.config.js <Nombre del archivo backup ecosystem.config.js> |

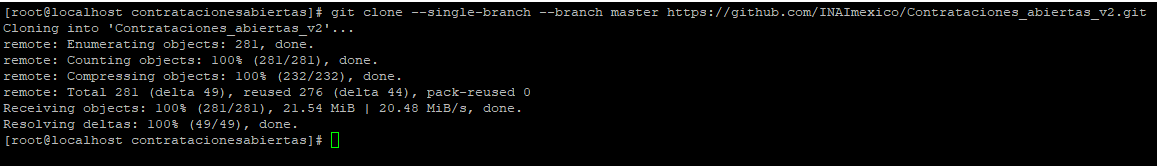
Como se muestra en el siguiente ejemplo:

****

# Descargar el archivo del código.

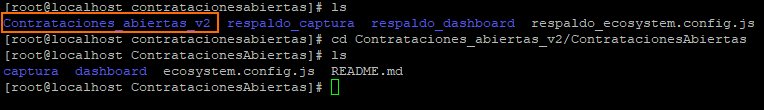
**Paso 1.** Clonar el repositorio GitHub con los archivos del código actualizado, ejecutando el siguiente comando:

|  |
| --- |
| git clone --single-branch --branch master https://github.com/INAImexico/Contrataciones\_abiertas\_v2.git |



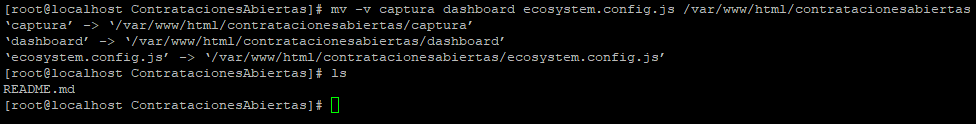
**Paso 2.** Una vez descargado el código, acceder a la ruta que contiene los directorios *captura*, *dashboard* y el archivo *ecosystem.config.js.*

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



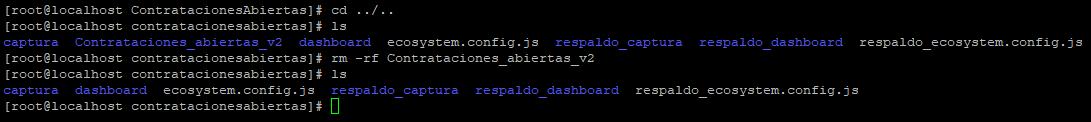
**Paso 3.** Mover los directorios *captura* y *dashboard* y el archivo *ecosystem.config.js* a la carpeta raíz, ejecutando el siguiente comando:

|  |
| --- |
| mv -v captura dashboard ecosystem.config.js /var/www/html/contratacionesabiertas |



**Paso 4.** Posicionarse en la carpeta donde se clonó el repositorio para eliminarlo, ejecutando los siguientes comandos:

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



# Ejecución de la actualización

Esta actualización contempla, la creación de usuarios y contraseñas de acuerdo a las políticas propias de seguridad (ejemplo: Longitud de contraseña, números, uso de mayúsculas, minúsculas, uso de caracteres especiales), un usuario es para el **esquema public** (Módulo Sistema de captura) y el otro usuario para el **esquema dashboard** (Módulo Dashboard - visualizador), y la configuración de los archivos de conexión a la base de datos con los usuarios recién creados.

**Nota.** Es importante mencionar que los usuarios a crear para los esquemas, deben de ser todo en minúsculas.

# Actualización de la base de datos

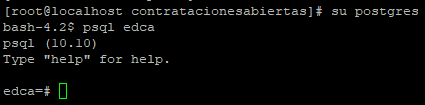
Tomando en cuenta que la base de datos contiene procedimientos de contratación registrados y los usuarios iniciales de instalación (tester y dashboard\_tester) son los propietarios de los esquemas public y dashboard, realizaremos los siguientes pasos para conservar los datos y actualizar los usuarios propietarios de cada esquema.

**Paso 1.** Ingresar al gestor de bases de datos postgresql, ejecutando el comando:

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

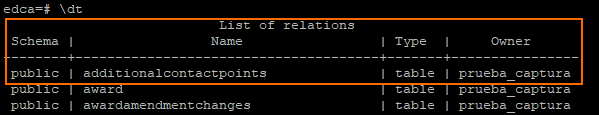
y para entrar directamente a la base de datos “edca”, ejecutar el comando:

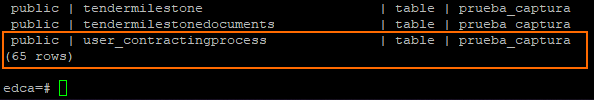
|  |
| --- |
| > psql edca |



**Paso 2.** Una vez dentro de la base, comprobar el usuario propietario del esquema public, ejecutando el comando:

|  |
| --- |
| > \dt |





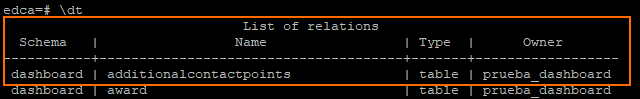
Al final de la tabla de relaciones se debe de mostrar el conteo de: 65 filas.

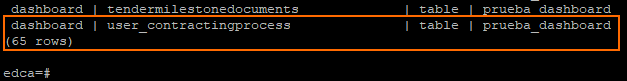
**Paso 3.** Para mostrar la tabla de relaciones del esquema dashboard, se deben de ejecutar los siguientes comandos para cambiar al esquema dashboard y luego mostrar la tabla:

|  |
| --- |
| > set search\_path to dashboard; |



|  |
| --- |
| > \dt |

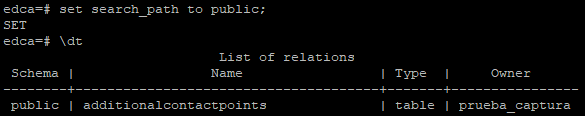




Al final de la tabla de relaciones se debe de mostrar el conteo de: 65 filas.

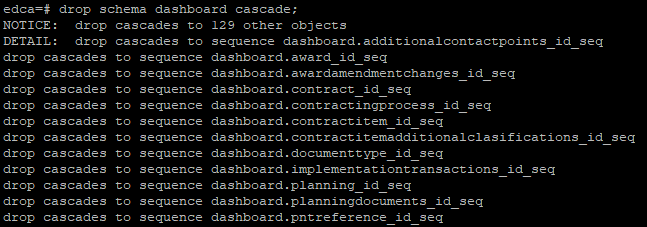
**Paso 4.** Para regresar a la tabla de relaciones del esquema public, se debe ejecutar el comando:

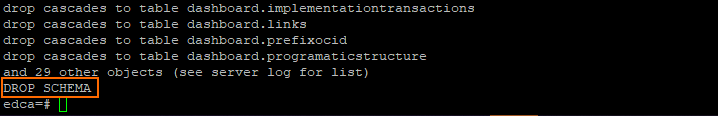
|  |
| --- |
| > set search\_path to public; |



**Paso 5.** Borrar el esquema dashboard, se debe ejecutar el comando:

|  |
| --- |
| > drop schema dashboard cascade; |





**Paso 6.** Corroborar que el esquema dashboard ya no contenga tablas, primero cambiar el esquema y luego ejecutar el comando \dt:

|  |
| --- |
| > set search\_path to dashboard; |

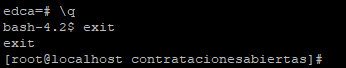


|  |
| --- |
| > \dt |



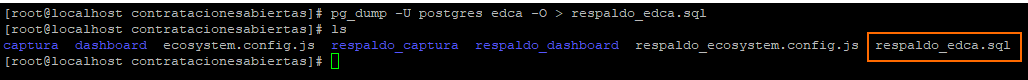
**Paso 7.** Salir de postgres ejecutando el comando:

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



**Paso 8.** Crear un dump de la base de datos edca sin usuarios propietarios, ejecutar el comando:

|  |
| --- |
| pg\_dump -U postgres edca -O > respaldo\_edca.sql |

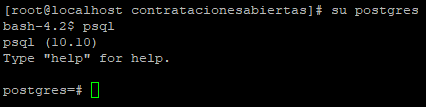


# Creación de los nuevos usuarios

**Nota: Los usuarios y password creados en este manual, son datos de prueba.**

**Paso 1.** Ingresar al gestor de bases de datos postgresql, ejecutando el comando:

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

****

**Paso 2.** Borrar la base de datos edca, ejecutando el comando:

|  |
| --- |
| > drop database edca; |

****

**Paso 3.** Crear la base de datos edca, ejecutando el comando:

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

****

**Paso 4.** Crear el nuevo usuario que será el propietario del **esquema public**, ejecutando el comando:

|  |
| --- |
| > create user nuevo\_captura with password 'nuevo\_password'; |

****

**Paso 5.** Asignar al usuario nuevo\_captura los permisos sobre la base de datos, ejecutando el comando:

|  |
| --- |
| > grant all privileges on database edca to nuevo\_captura; |

****

**Paso 6.** Crear el nuevo usuario que será el propietario del **esquema dashboard**, ejecutando el comando:

|  |
| --- |
| > create user nuevo\_dashboard with password 'nuevo\_password'; |

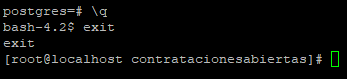
****

**Paso 7.** Asignar al usuario nuevo\_dashboard los permisos sobre la base de datos, ejecutando el comando:

|  |
| --- |
| > grant all privileges on database edca to nuevo\_dashboard; |

**Paso 8.** Salir de postgres ejecutando el siguiente comando:

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



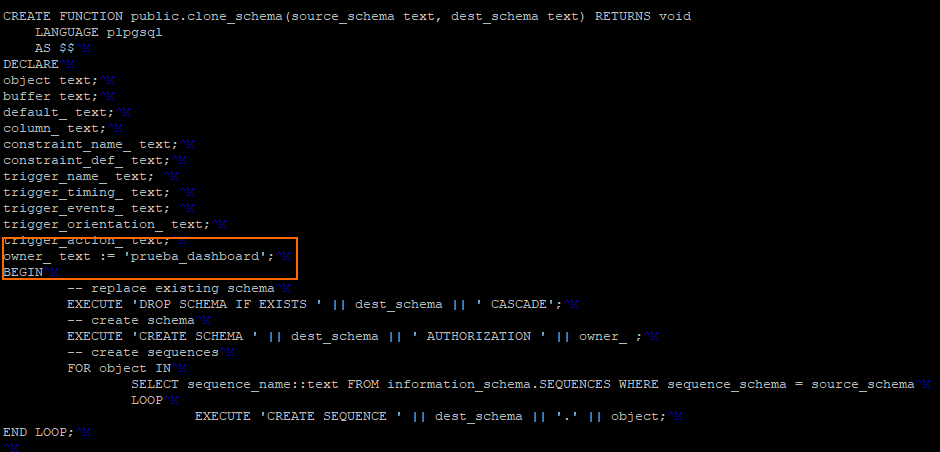
# Actualizar usuario en la base de datos

**Paso 1.** Abrir el archivo del dump de la base de datos que acabamos de crear ejecutando el comando:

|  |
| --- |
| vi respaldo\_edca.sql |



Y actualizar la variable “owner\_ text” con el usuario creado para el esquema dashboard:

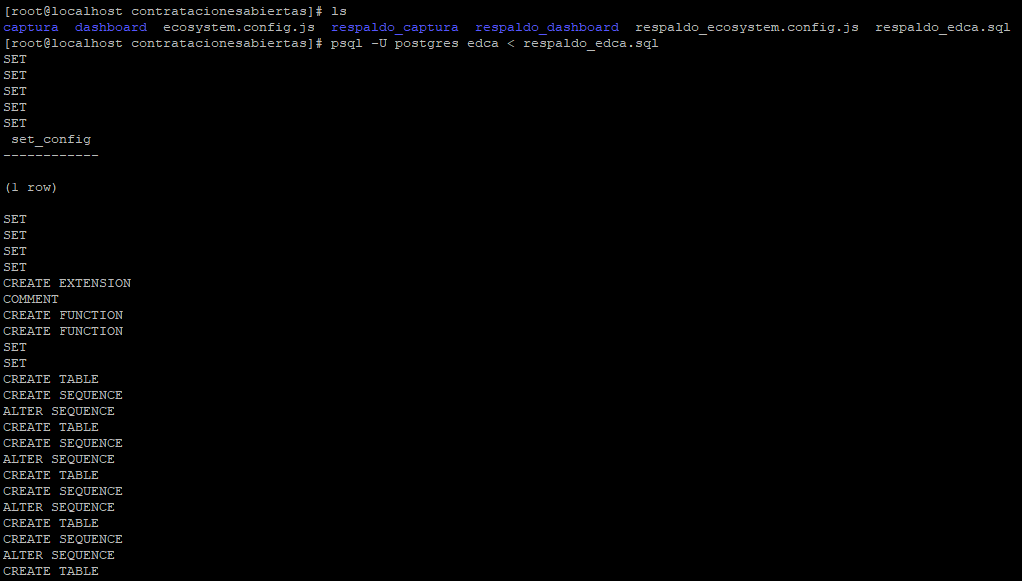


Guardar los cambios y salir.

# Importar el dump a la base de datos edca

**Paso 1.** Ejecutar el siguiente comando para cargar el respaldo (creado previamente en el apartado 2.1 Actualización de la base de datos – Paso 8) de la base de datos:

|  |
| --- |
| psql -U postgres edca < respaldo\_edca.sql |



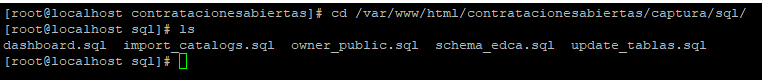
# Actualización y ejecución de scripts

En la carpeta “sql” se encuentran los archivos que se deben de actualizar con los usuarios creados para cada esquema:

* dashboard.sql
* owner\_public.sql
* update\_tablas.sql

**Paso 1.** Realizar la actualización y ejecución de los scripts, ingresar a la carpeta “sql” ejecutando el siguiente comando:

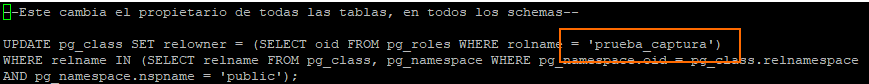
|  |
| --- |
| cd /var/www/html/contratacionesabiertas/captura/sql/ |



**Paso 2.** Abrir el archivo owner\_public.sql ejecutando el comando:

|  |
| --- |
| vi owner\_public.sql |

Modificar el usuario prueba\_captura por el usuario creado para el **esquema public**, guardar los cambios y salir.

**

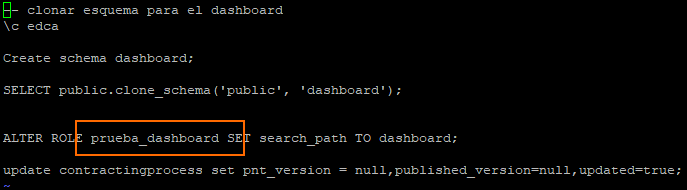
**Paso 3.** Ejecutar el comando

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

**Paso 4.** Abrir el archivo dashboard.sql ejecutando el comando:

|  |
| --- |
| vi dashboard.sql |

Modificar el usuario prueba\_dashboard por el usuario creado para el **esquema dashboard**, guardar los cambios y salir.



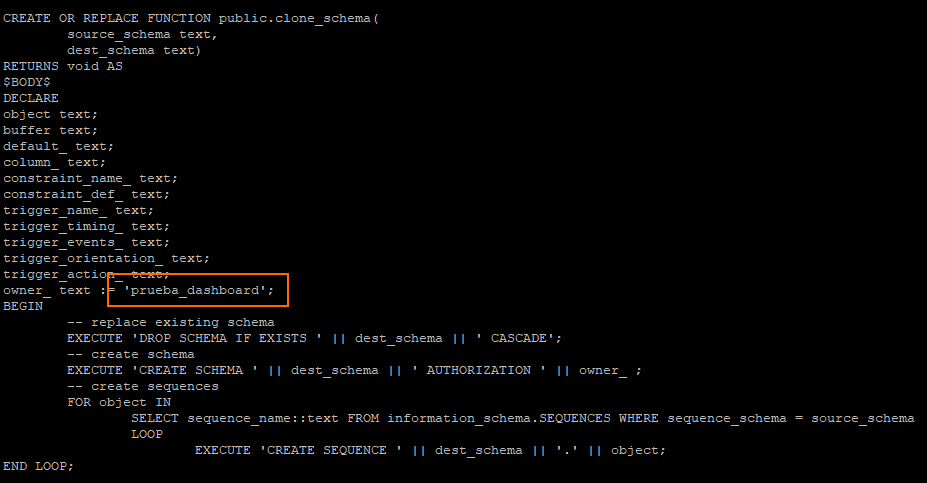
**Paso 5.** Ejecutar el comando

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

**Paso 6.** Abrir el archivo update\_tablas.sql ejecutando el comando:

|  |
| --- |
| vi update\_tablas.sql |

Modificar el usuario prueba\_dashboard por el usuario creado para el esquema dashboard, guardar los cambios y salir.



**Paso 7.** Ejecutar el comando

|  |
| --- |
| psql -U postgres edca < update\_tablas.sql |

# Compilación y ejecución.

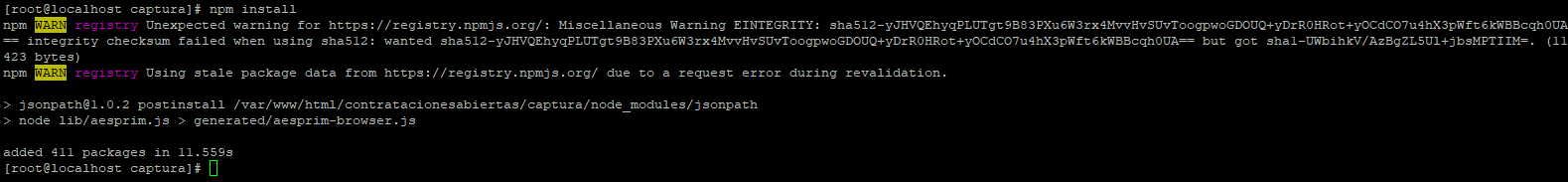
**Paso 1.** Realizar la compilación y ejecución del código, ingresar a la siguiente ruta:

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

**Paso 2.** Ejecutar el siguiente comando:

|  |
| --- |
| npm install |

Ejemplo:

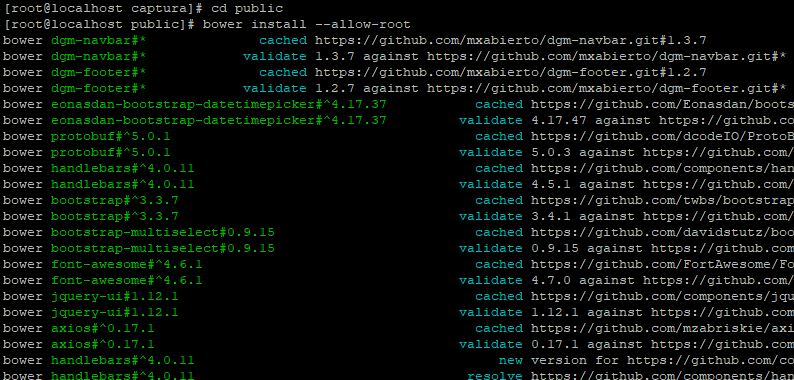


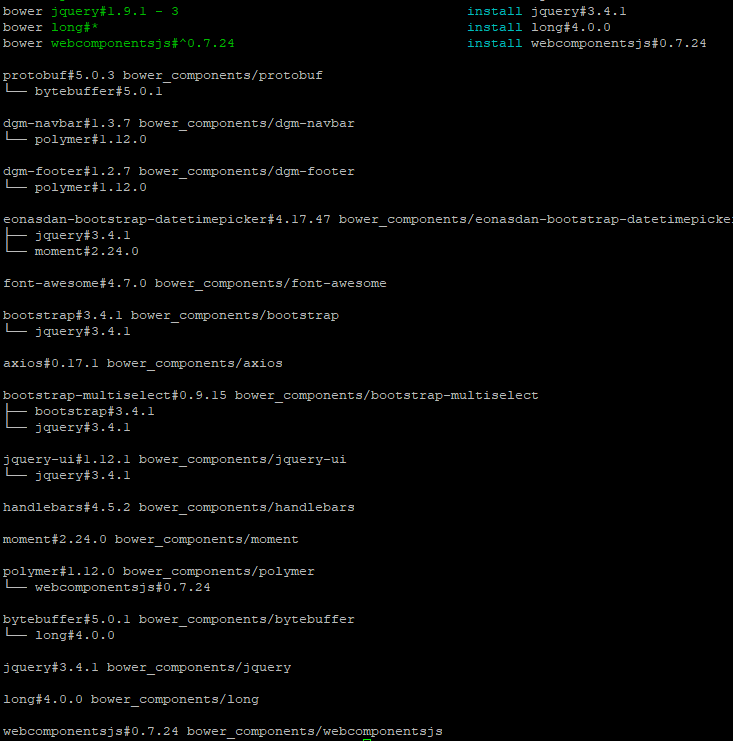
**Paso 3.** Ingresar a la carpeta “public” que se encuentra en la siguiente ruta:

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

**Paso 4.** Ejecutar el siguiente comando para la instalación de dependencias del sistema por medio de bower:

|  |
| --- |
| bower install --allow-root |

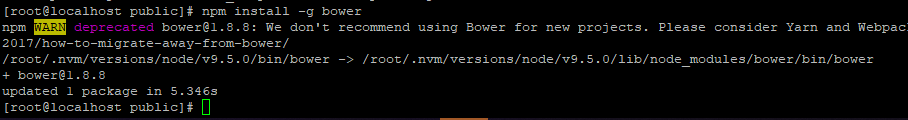




**Nota:** Si la ejecución del comando para instalar bower **muestra un error**, pasar al número “4.1” y seguir los pasos indicados. En caso de que **no se muestre error** ejecutar el siguiente paso: “Paso 5” y saltarse las indicaciones del número “4.1”

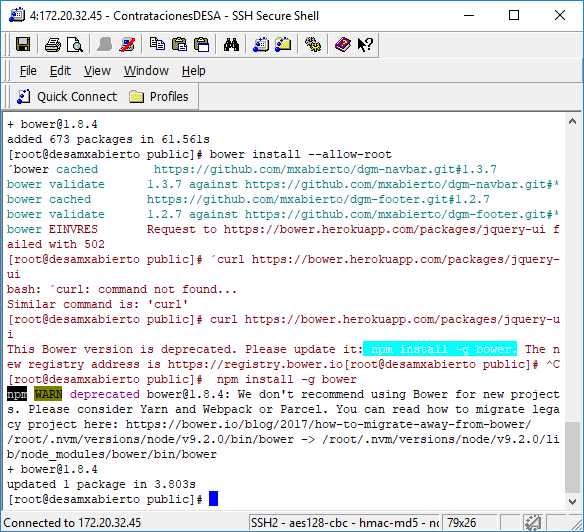
**Paso 5.** Ejecutar el siguiente comando de bower:

|  |
| --- |
| npm install -g bower |



# Error al ejecutar la instalación de Bower.

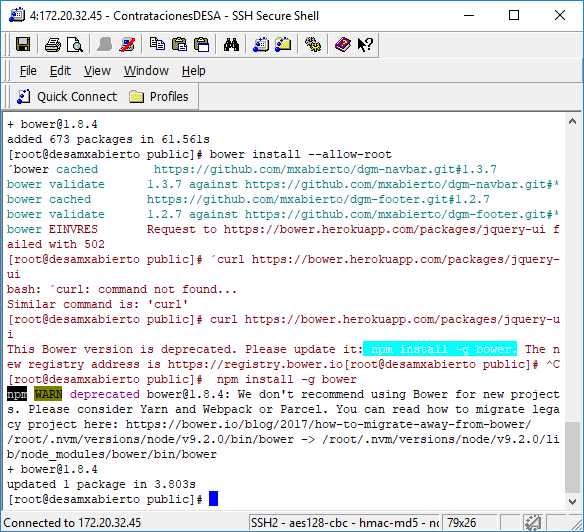
**Nota:** Si al ejecutar el comando “*bower install --allow-root*” se muestra un error:



Realizar los pasos siguientes:

**Paso 1.** Ejecutar el comando:

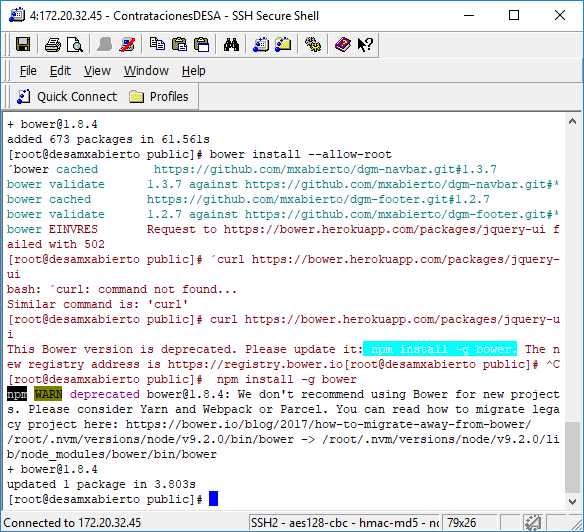
|  |
| --- |
| curl https://bower.herokuapp.com/packages/jquery-ui |



Si se muestra “*This bower versión is deprecated. Please update it: npm install -g bower*”

**Paso 2.** Ejecutar el comando:

|  |
| --- |
| npm install -g bower |



**Paso 3.** Para desinstalar el bower de /usr/bin, ejecutar el siguiente comando:

|  |
| --- |
| npm uninstall bower -g --prefix=/usr |

**Paso 4.** Ejecutar el comando:

|  |
| --- |
| bower install --allow-root |

**Paso 5.** Ejecutar el comando:

|  |
| --- |
| npm install -g bower |

**Paso 6.** Ejecutar el comando para conocer el repositorio raíz de npm:

|  |
| --- |
| npm list -g --depth=0 |

Este comando arrojara una salida como la siguiente:

**/home/inai/.npm-data/lib**

**├── bower@1.8.4**

**├── npm@6.5.0**

**└── pm2@3.2.2**

Sustituir la palabra ***lib*** del final, por **bin**

**/home/inai/.npm-data/bin**

Y agregar esa ruta al PATH con el siguiente comando:

|  |
| --- |
| export PATH=$PATH: /home/inai/.npm-data/bin |

Es importante mencionar que la ruta que se menciona en el paso anterior es de ejemplo, se tiene que dejar la que arroje de salida el comando npm list -g –depth=0

# Validación de conexión a base de datos.

Es necesario validar que la conexión a base de datos sea correcta.

**Paso 1.** Ingresar a la carpeta “captura” en la siguiente ruta:

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

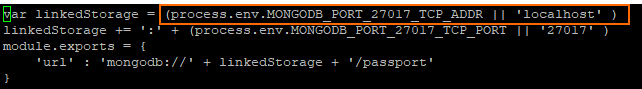
**Paso 2.** Abrir el archivo **db.js**



La variable linkedStorage debe estar de la siguiente manera:

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

Ejemplo:



# Asignación de variables para conexión a la API de PNT.

Es necesario validar que la conexión a la API de PNT sea correcta, esto se realiza en los archivos:

* process-pnt.js
* api-pnt.js

**Paso 1.** Ingresar a la siguiente ruta:

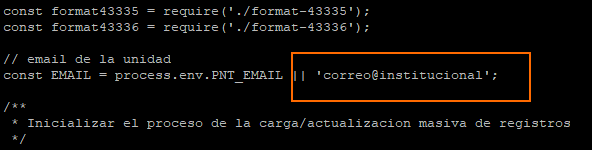
|  |
| --- |
| cd /var/www/html/contratacionesabiertas/captura/pnt/ |

**Paso 2.** Abrir el archivo process-pnt.js

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

La variable EMAIL se debe de configurar con el email de la unidad, dejarlo como se ve en la siguiente línea:

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



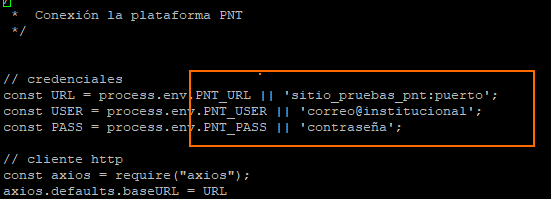
**Paso 4.** Guardar los cambios y salir del archivo.

**Paso 5.** Abrir el archivo api-pnt.js

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

En las siguientes variables se configuran las credenciales correspondientes de acuerdo a sus datos propios de configuración:

|  |
| --- |
| **const** **URL = process.env.PNT\_URL || 'sitio\_pruebas\_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 su órgano garante local.

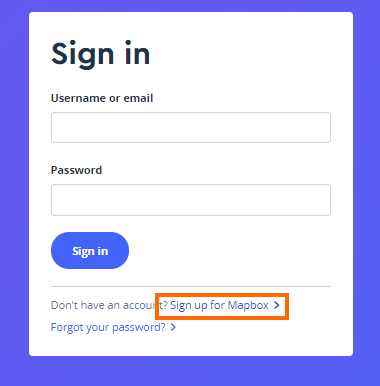
# 4.4 Asignación de variables MAPTOKEN.

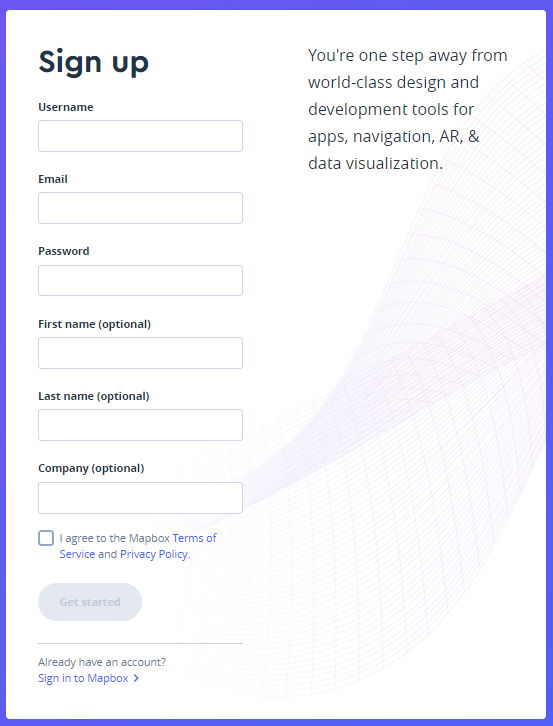
Para el servicio de Mapas es necesario configurar lo siguiente:

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

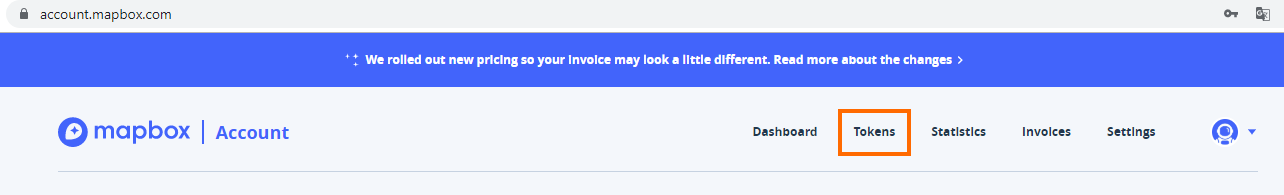
<https://www.mapbox.com/signup/>

Donde hay que registrarse

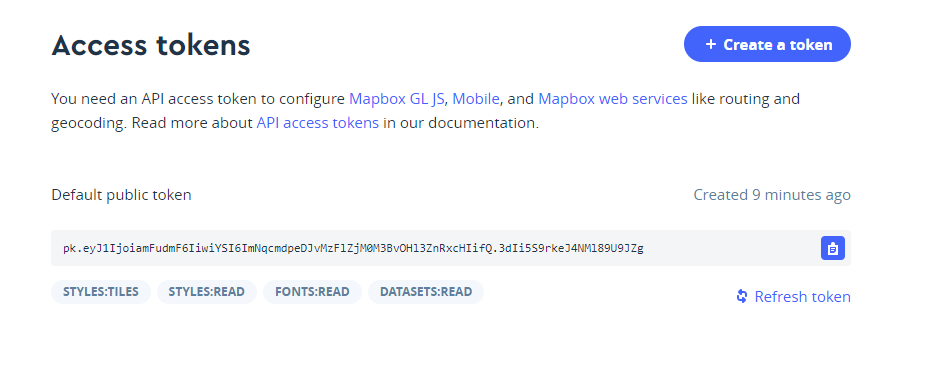




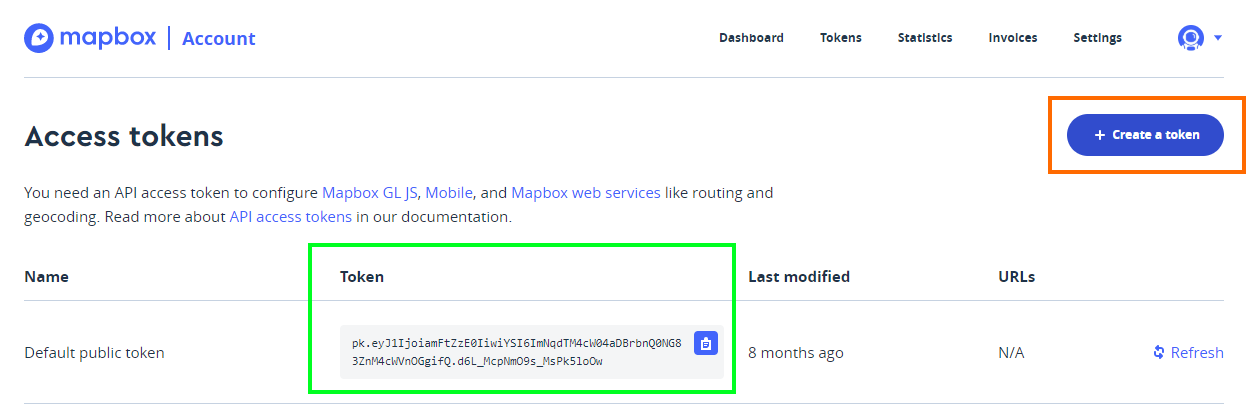
**Paso 2.** Una vez realizado el registro, ingresar a mapbox y dar clic en la opción “Tokens”:

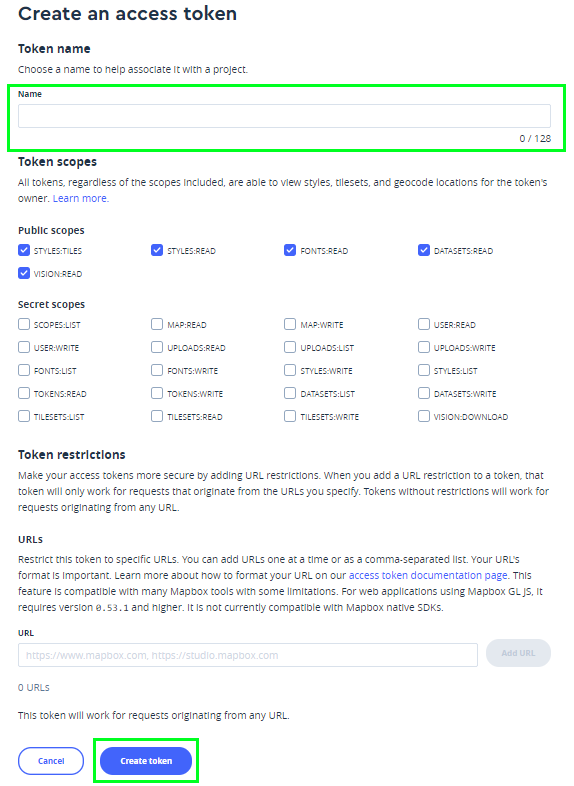


Y se visualiza la siguiente pantalla:



Se puede utilizar el valor del token que se encuentra por default o crear uno, asociándole un nombre:





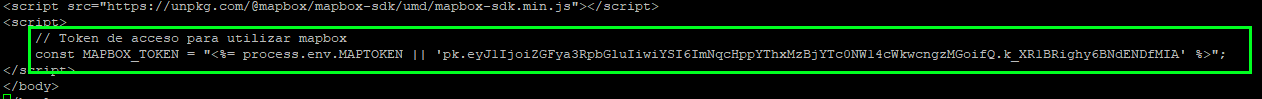
**Paso 3.** Una vez que se obtenga el token, ingresar a la siguiente ruta:

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

**Paso 4.** Abrir el archivo main.js

|  |
| --- |
| vi main.js |

Y buscar en el archivo el siguiente valor: MAPBOX\_TOKEN y actualizar el valor del token



**Paso 5.** Guardar cambios y salir.

# Iniciar servicio del “Sistema de captura”

**Paso 1.** Ingresar a la siguiente ruta:

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

**Paso 2.** Ejecutar el siguiente comando, para iniciar el servicio:

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

# Actualización de dashboard.

Ahora se tienen que ejecutar los siguientes pasos para la actualización de código del módulo dashboard - visualizador.

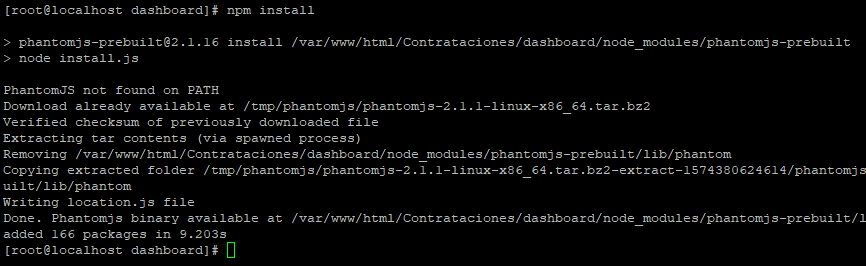
**Paso 1.** Ingresar a la siguiente ruta:

|  |
| --- |
| cd /var/www/html/contratacionesabiertas/dashboard |

**Paso 2.** Realizar la ejecución del comando:

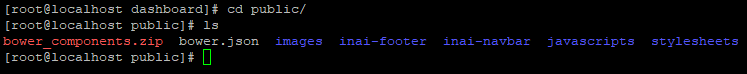
|  |
| --- |
| npm install |

La ejecución del comando anterior no debe de marcar ningún error, el resultado debe de ser similar a:

****

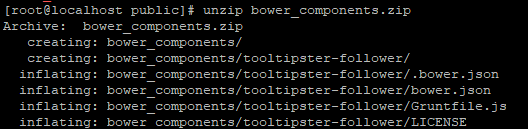
**Paso 3.** Ingresar a la carpeta “public” para instalar bower, ejecutando:

|  |
| --- |
| cd public |



**Paso 4.** Descomprimir el archivo llamado bower\_componentes.zip ejecutando el siguiente comando:

|  |
| --- |
| unzip bower\_components.zip |



**Paso 5.** Ingresar a la siguiente ruta:

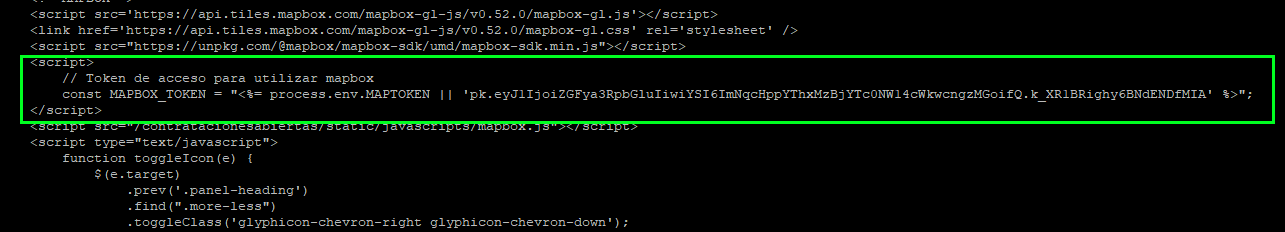
|  |
| --- |
| cd dashboard/views/ |

**Paso 6.** Abrir el archivo contract.ejs ejecutando el siguiente comando:

|  |
| --- |
| vi contract.js |

****

**Paso 6.** Actualizar el valor MAPBOX\_TOKEN por el token obtenido en el paso número 4.4

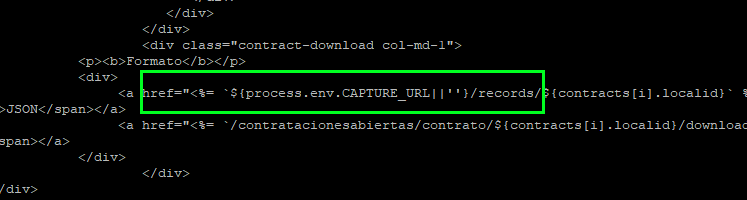


**Paso 7.** Guardar cambios y salir.

**Paso 8.** Abrir el archivo contracts.ejs ejecutando el siguiente comando:

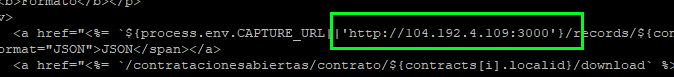
|  |
| --- |
| vi contracts.js |

**Paso 9.** Buscar el siguiente valor CAPTURE\_URL:



Enseguida de la variable: **process.env.CAPTURE\_URL** entre las comillas simples **||''}/**

Se debe de ingresar la URL del sistema de captura de contrataciones, ejemplo:



**Paso 10.** Guardar cambios y salir.

# Iniciar servicio del “dashboard - visualizador”

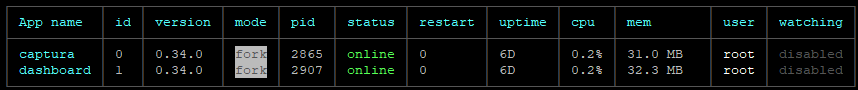
**Paso 1.** Ingresar a la siguiente ruta:

|  |
| --- |
| cd /var/www/html/contratacionesabiertas/dashboard |

**Paso 2.** Ejecutar el siguiente comando, para iniciar el servicio:

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

**Paso 3.** Ejecutamos el comando pm2 list para ver el estado de los servicios y que estén correctos.



# Acceder a los módulos.

Una vez iniciado los servicios de los dos módulos, para ingresar a cada uno de ellos, ejecutamos los siguientes pasos.

# Módulo de captura.

**Paso 1.** Ingresar a la siguiente dirección:

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



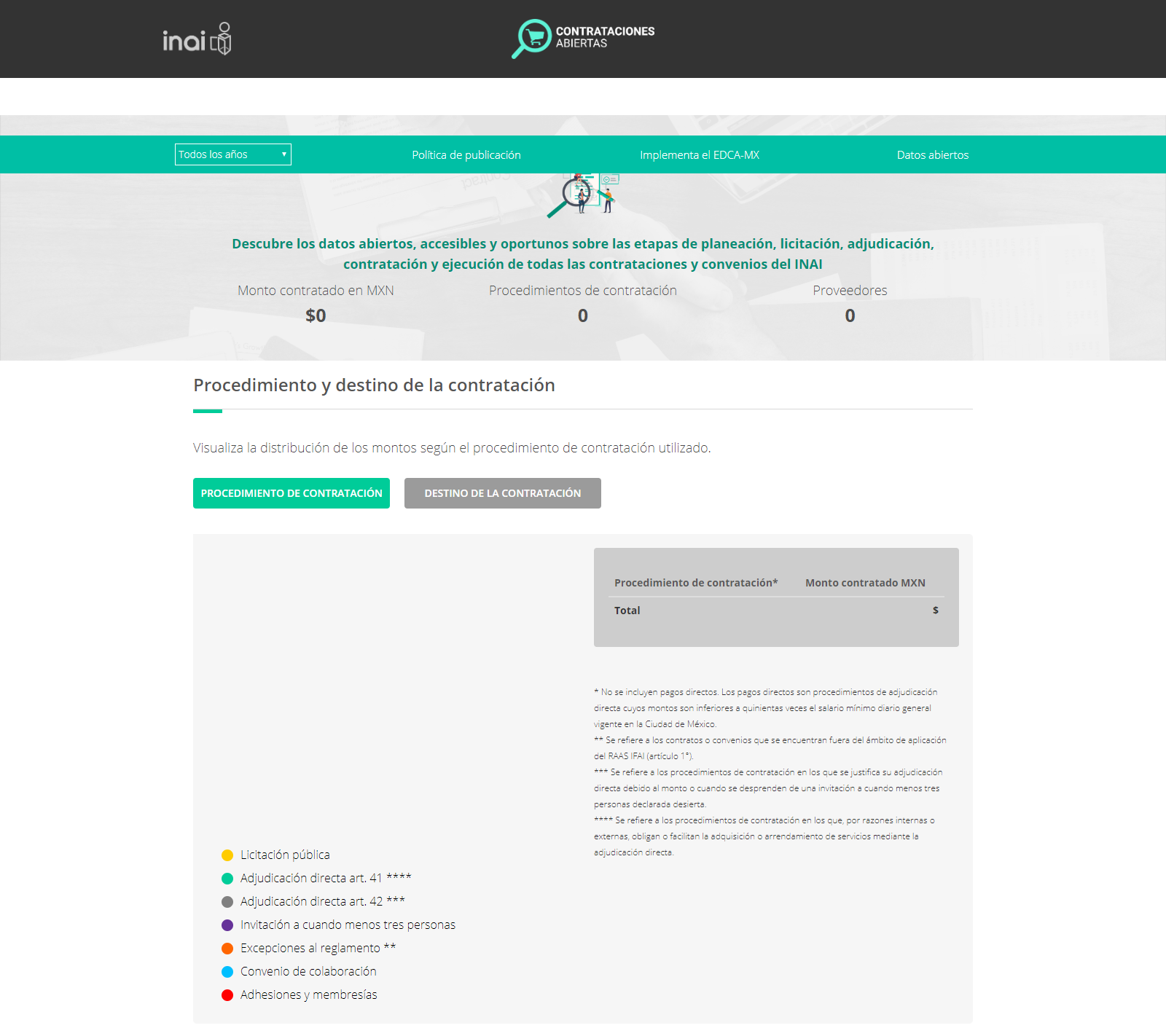
**Paso 2.** Ingresar el usuario y contraseña, creados desde la primera instalación.



# Módulo de visualización (dashboard).

**Paso 1.** Ingresar a la siguiente dirección:

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



Debido a que se borró el esquema dashboard y posteriormente con la actualización se volvió a crear, el módulo visualizador no mostrará información. Hasta que se validen y publiquen los procedimientos de contratación.

Con esto se finaliza la actualización del código.