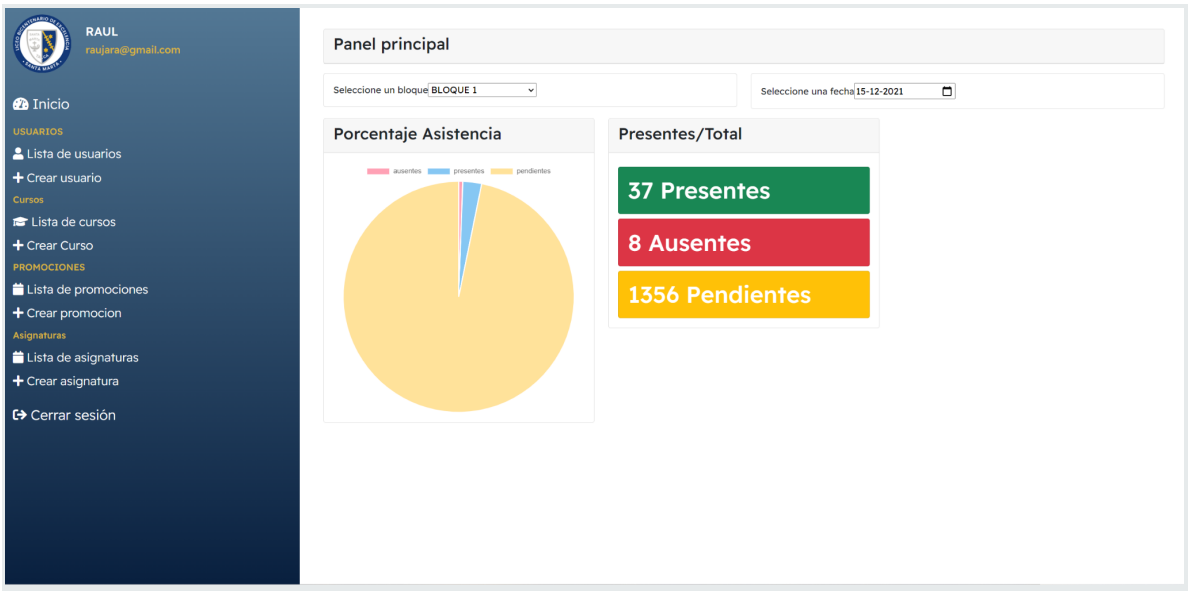
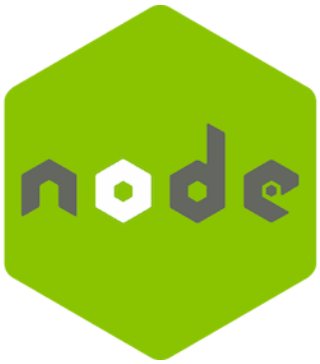
Manual de instalación sistema ASSIES
En linux



Instalación de Node.js en Linux



Para instalar Angular CLI, debe tener la última versión de Node.js y NPM instalada en su sistema Linux.

$ sudo curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash - [for Node.js version 12]

curl -sL https://deb.nodesource.com/setup_12.x | bash - [for Node.js version 12]

En versiones más nuevas de linux basta con ejecutar el comando:

sudo apt install install nodejs
sudo apt install npm

Además, para compilar e instalar complementos nativos de NPM, es posible que deba instalar herramientas de desarrollo en su sistema de la siguiente manera.

$ sudo apt install -y build-essential [On Debian/Ubuntu]
# yum install gcc-c++ make [On CentOS/RHEL]
# dnf install gcc-c++ make [On RHEL 8/Fedora 22+]

Instalación de Angular CLI en Linux

Una vez que tenga Node.js y NPM instalados, como se muestra arriba, puede instalar Angular CLI usando el administrador de paquetes npm de la siguiente manera (el indicador -g significa instalar la herramienta en todo el sistema para ser utilizada por todos los usuarios del sistema).

npm install -g @angular/cli
O
$ sudo npm install -g @angular/cli

```
[root@tecmint ~]# npm install -g @angular/cli
/usr/bin/ng -> /usr/lib/node_modules/@angular/cli/bin/ng

> @angular/cli@8.1.2 postinstall /usr/lib/node_modules/@angular/cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data with the Angular Team at Google under
Google's Privacy Policy at https://policies.google.com/privacy? For more details and
how to change this setting, see http://angular.io/analytics. No
+ @angular/cli@8.1.2
added 239 packages from 185 contributors in 411.91s
[root@tecmint ~]#
```

Puede iniciar Angular CLI usando el ejecutable ng que ahora debería estar instalado en su sistema. Ejecute el siguiente comando para verificar la versión de Angular CLI instalada.

ng –version

```
[root@tecmint ~]# ng --version

     _                      _                 ____ _     ___
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / △ \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI: 8.1.2
Node: 10.14.1
OS: linux x64
Angular:
...

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.801.2
@angular-devkit/core         8.1.2
@angular-devkit/schematics   8.1.2
@schematics/angular          8.1.2
@schematics/update           0.801.2
rxjs                         6.4.0

[root@tecmint ~]#
```

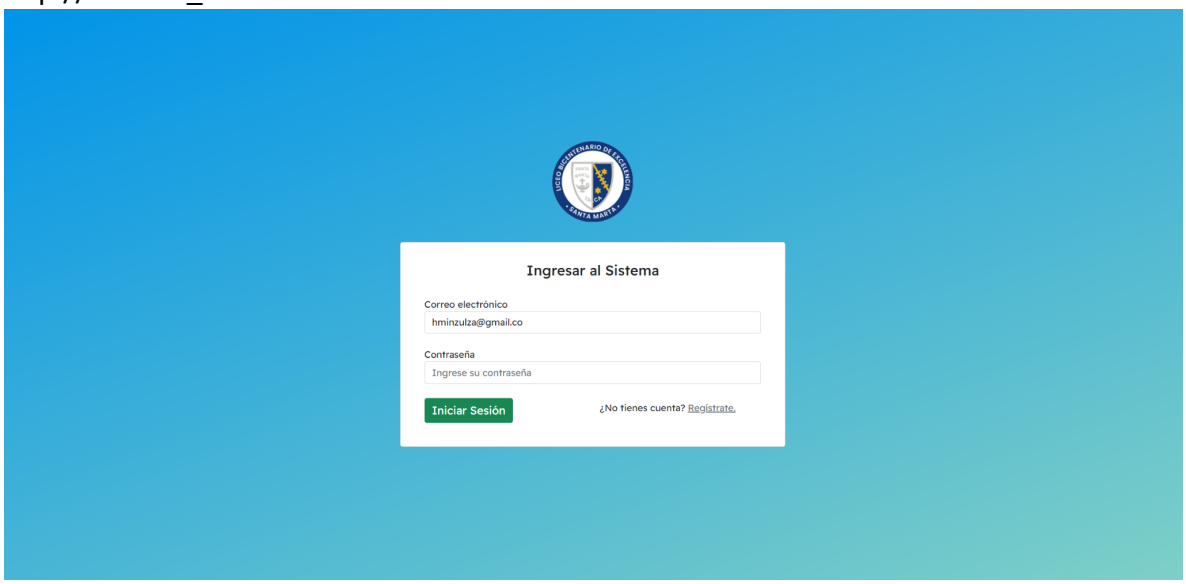
PERMITIR APP A TRAVÉS DE FIREWALL

Permitimos conexión en el firewall a través del puerto 4200

$ sudo ufw allow 4200/tcp
$ sudo ufw reload

Ahora puede abrir un navegador web y navegar usando la siguiente dirección para ver cómo se ejecuta la nueva aplicación como se muestra en la siguiente captura de pantalla.

http://localhost:4200/
or
http://SERVER_IP:4200



Si usa el comando ng serve para construir una aplicación y servirla localmente, como se muestra arriba, el servidor reconstruye automáticamente la aplicación y recarga la (s) página (s) web cuando cambia cualquiera de las fuentes. archivos.

Para obtener más información sobre la herramienta ng, ejecute el siguiente comando.

ng help

La página de inicio de Angular CLI: https://angular.io/cli
En este artículo, hemos mostrado cómo instalar Angular CLI en diferentes distribuciones de Linux. También cubrimos cómo construir, compilar y servidor una aplicación angular básica en un servidor de desarrollo. Para cualquier consulta o pensamiento que desee compartir con nosotros, utilice el formulario de comentarios a continuación.

Instalación de MySQL

Primero que nada debemos abrir la terminal, ya sea desde el menú de aplicaciones, o bien utilizando la combinación de teclas Ctrl+Alt+T.

Posteriormente recomiendo actualizar los paquetes del sistema, que es una buena práctica que deben hacer de manera periódica, con los siguientes comandos:

```
$sudo apt-get update
$sudo apt-get upgrade
$sudo apt-get autoremove
```

Después de esto comenzamos instalando MySQL server, que es en sí, lo que nos permitirá utilizar Workbench, ya que si instalamos Workbench sin haber instalado MySQL server no nos permitirá usarlo.

- **Instalar Mysql**

Para instalarlo usamos el comando:

```
$sudo apt-get install mysql-server
```

Una vez que se haya instalado podemos revisar que todo esté bien usando el comando:

```
$sudo mysql
```

Si hemos asignado una contraseña

Nos deberá desplegar, entre varias cosas, la versión de MySQL y previo al cursor dirá mysql>, esto quiere decir que estamos dentro de la aplicación de mysql en la terminal. Para salir de ella podemos escribir exit.

- **Instalar Workbench**

Bien, ahora que ya tenemos MySQL server en nuestra máquina procederemos a instalar MySQL Workbench, que es el entorno gráfico que nos ayudará en este curso.

```
sudo snap install mysql-workbench-community
```

Luego en la consola, se debe ejecutar los siguientes comandos, con el propósito de cambiar el plugin de auth_socket a password_native y así poder acceder con contraseña.

```
mysql_native_password:

sudo mysql -u root -p
mysql> use mysql
mysql> SELECT User, Host, plugin FROM mysql.user;
mysql> UPDATE user SET plugin='mysql_native_password' WHERE User='root';
mysql> FLUSH PRIVILEGES;
```

Revisamos que los cambios se hayan efectuado:

```
mysql> SELECT User, Host, plugin FROM mysql.user;
```

Finalmente escribimos exit y presionamos enter.

Cabe aclarar que solo te deja usar el comando sudo mysql -u root -p una vez, ya que si lo usas por segunda vez te pedirá un password que no has generado, por lo que hay que generarlo, con el siguiente comando:

```
mysqladmin -u root password tupassword
```

Después de hacer esto ya podrás acceder al usuario root con ese password.

Finalmente si al momento de abrir el Local instance 3306 aparece el error Cannot Connect to Database Server, es debido a que workbench usa conexiones ssh y Password Manager para funcionar correctamente.
Por lo que hay que otorgar los permisos con los siguientes comandos en la terminal:

```
$ snap connect mysql-workbench-community:password-manager-service
$ snap connect mysql-workbench-community:ssh-keys
```

Importar Base de Datos

Para comenzar a utilizar el sistema, ingresamos una base de datos de demostración que está contenida en un archivo de extensión SQL, este archivo se llama asistencia.sql y se puede descargar del repositorio https://github.com/jorgemaldonado87/asistencia_api.

```
21
22  DROP TABLE IF EXISTS `asignaturas`;
23  /*!40101 SET @saved_cs_client     = @@character_set_client */;
24  /*!50503 SET character_set_client = utf8mb4 */;
25  CREATE TABLE `asignaturas` (
26    `id` int(11) NOT NULL AUTO_INCREMENT,
27    `nombre` varchar(45) DEFAULT NULL,
28    PRIMARY KEY (`id`)
29  ) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
30  /*!40101 SET character_set_client = @saved_cs_client */;
31
32  --
33  -- Dumping data for table `asignaturas`
34  --
35
36  LOCK TABLES `asignaturas` WRITE;
```

Extracto archivo asignaturas

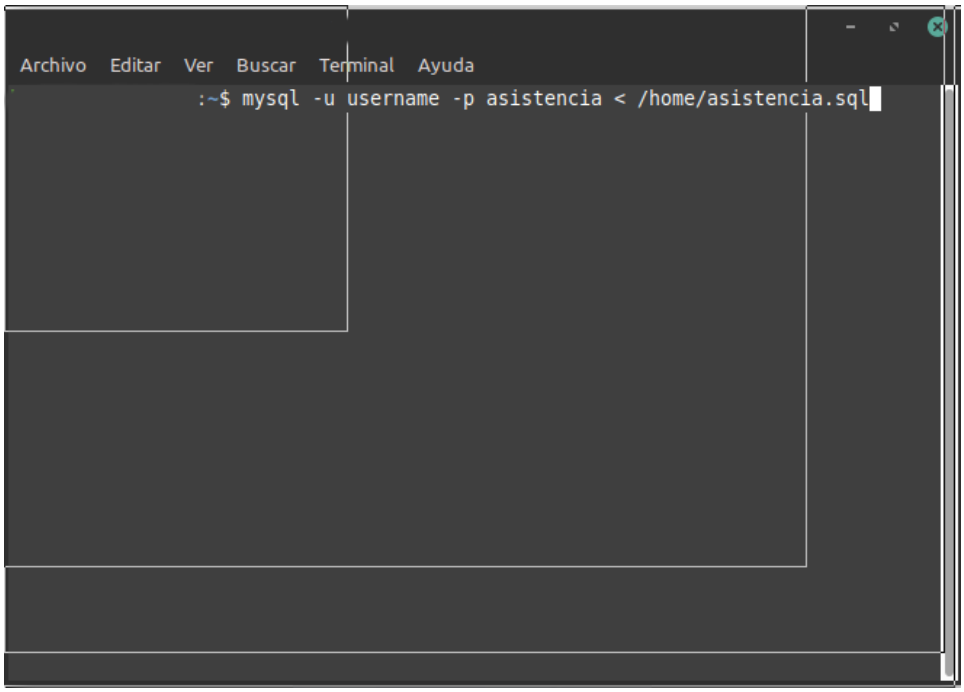
Para comenzar la importación de la base de datos podemos realizar dos procedimientos diferentes.

Consola

Para instalar a través de comandos de consola, debemos contar con **la ubicación de nuestro archivo de respaldo asistencia.sql** y con **mysql-client** instalado en nuestro servidor de base de datos.

Con lo anterior, el siguiente comando debería bastar para crear nuestra base de datos a partir de un archivo .sql

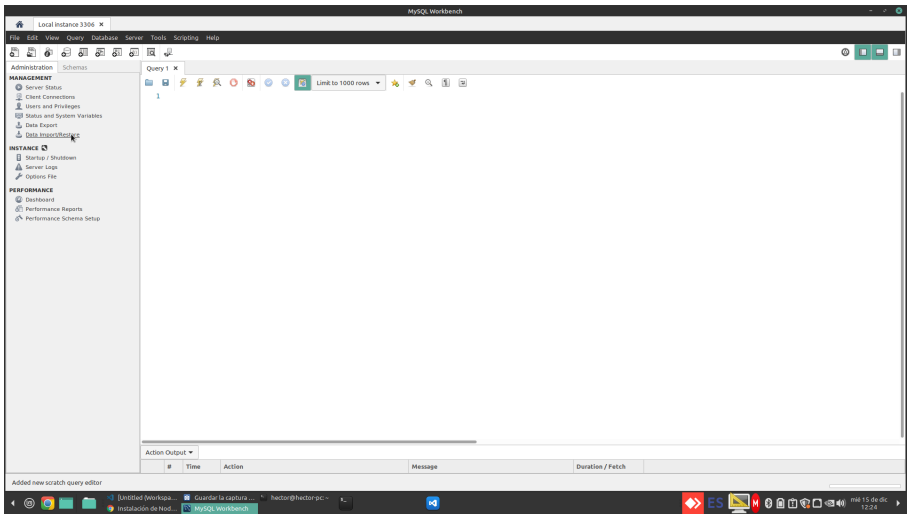
mysql -u username -p asistencia < /home/asistencia.sql



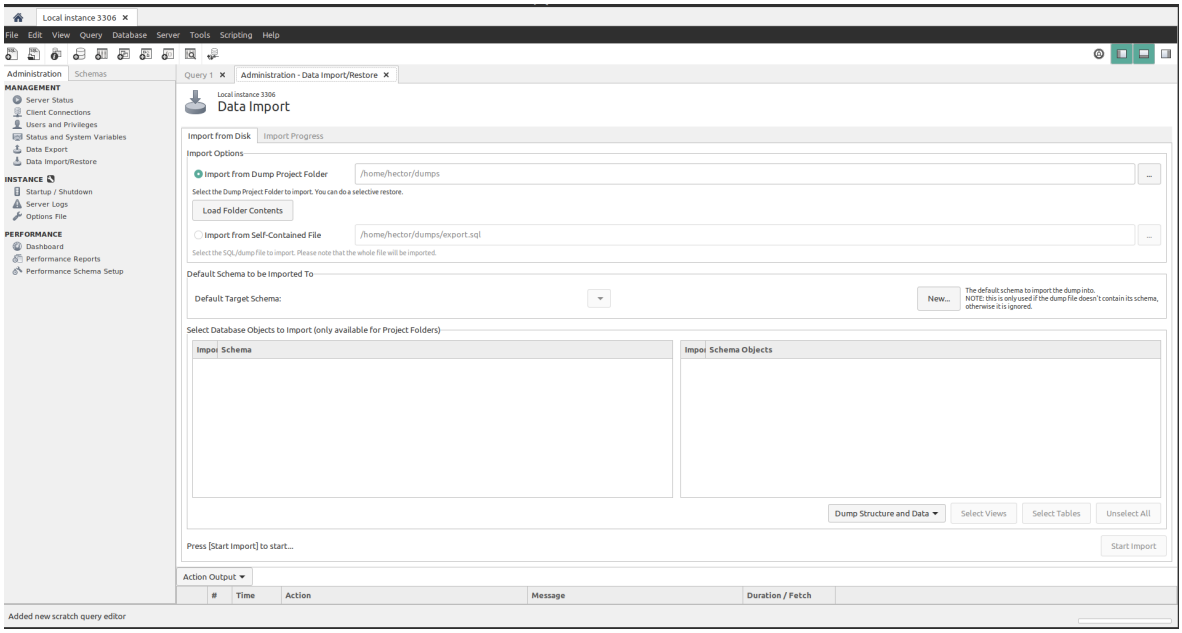
En el comando anterior, tendríamos:

- username: nombre de usuario
- asistencia: nombre base de datos
- /home/asistencia.sql ruta archivo a importar

Importar Base de Datos



Ir a la sección administración en la barra lateral izquierda, posterior a esto hacer click en Data Import / Importar datos según corresponda.



Seleccionamos el archivo asistencia.sql previamente descargado junto a la opción **Import From Self Contained File (sql).**

Import From Self Contained File (sql.)

Instalar proyectos

- **Instalar GIt**

Para instalar los respectivos proyectos, primero nos aseguraremos de tener una carpeta en donde clonarlos directamente desde nuestro repositorio Github. Así mismo, asegurarnos de tener instalado previamente GIT.

Para instalar GIT, ejecutamos el comando.

sudo apt install git

Opcionalmente se puede registrar el nombre de usuario, email y nombre en GIT para los respectivos commit y pulls que realizaremos haremos.

```
git config --global user.name "Gonzalo Maldonado"
git config --global user.name "jorgemaldonado87"
git config --global user.name "jorge.maldonado@alu.ucm.cl"
```

- **Clonar repositorios**

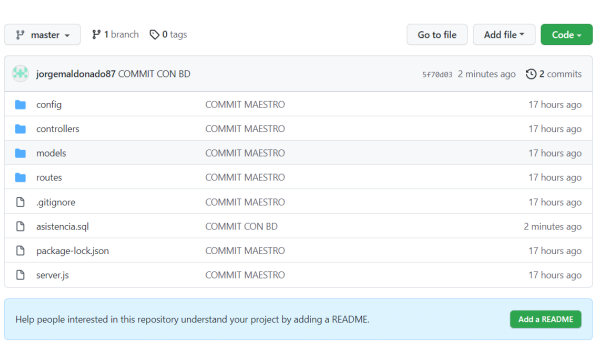
Para dejar ambos proyectos en una carpeta común y de fácil acceso, crearemos una carpeta asistencia en el proyecto raíz.

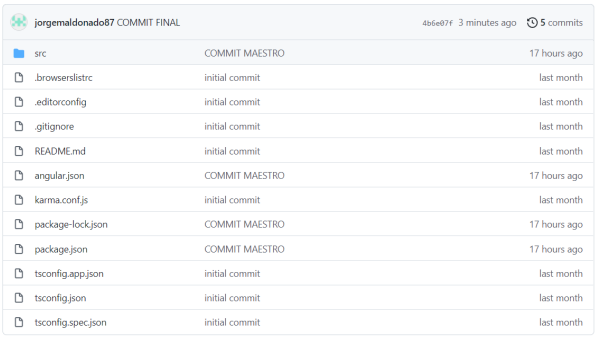
```
cd
sudo mkdir asistencia
cd asistencia
```

Ahora clonaremos los repositorios de nuestros sistemas de GitHub en nuestro computador, básicamente es como descargar el archivo zip y descomprimirlo pero lo hacemos a través del gestor oficial de GIT.

```
git clone https://github.com/jorgemaldonado87/asistencia_api asistencia_api
git clone https://github.com/jorgemaldonado87/asistencia asistencia
```

Ahora tendremos una carpeta asistencia y asistencia_api en nuestra carpeta asistencia, las cuales son nuestro proyecto de angular y express respectivamente.





- **Inicializar proyecto de angular**

Para inicializar el proyecto de angular, utilizaremos el gestor de paquetes NPM, el cual se encargará de descargar e instalar todos los módulos y componentes necesarios para que nuestra aplicación trabaje, estos paquetes se encuentran en el archivo packages.json


```
npm install
```

Con todos los módulos ya instalados, procederemos a cambiar el archivo environment.ts en la carpeta src/environments, para actualizar la ruta de la API donde estará nuestro servidor de express.

- **Inicializar proyecto de express (Node JS)**

Primero que todo cambiamos de carpeta a nuestra carpeta asistencia_api a través de los siguientes comandos.

```
cd ..
cd asistencia_api
```

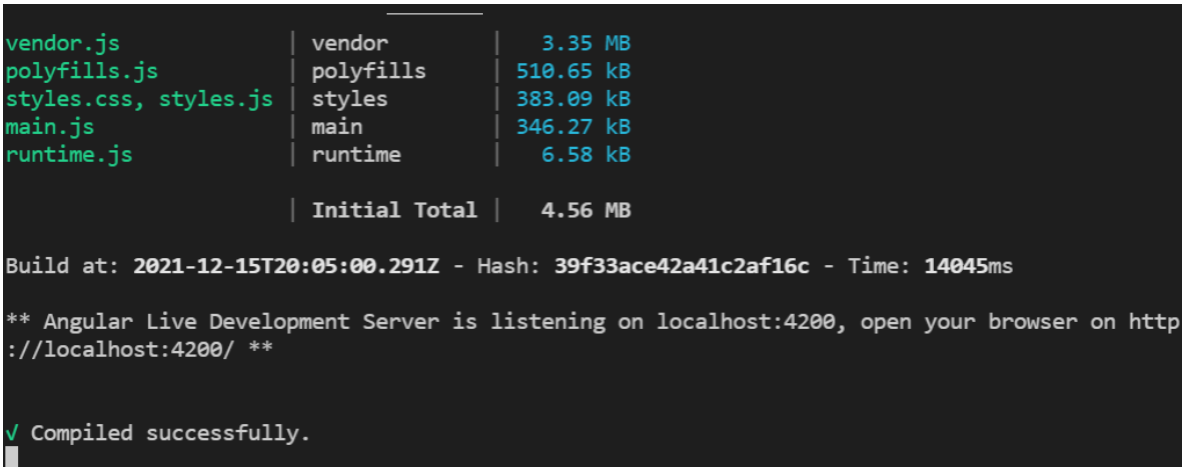
Al igual que en el proyecto anterior, ejecutaremos ahora un NPM install pra poder instalar los paquetes necesarios, además instalaremos manualmente los paquetes: express, sequelize, mysql-2, cors y json-decode, los cuales son necesarios para trabajar con nuestra aplicación de express.

```
npm install sequelize, express, mysql2, cors, json-decode
```

Iniciar proyectos

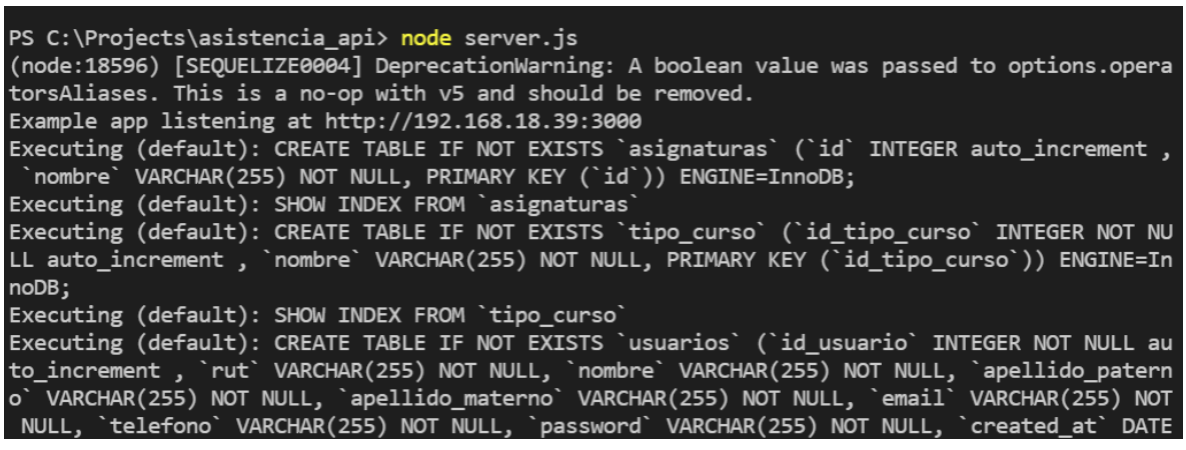
- Angular: Para iniciar el proyecto de angular se realiza con el comando ng serve

  `ng serve`



- Node JS Express: Para iniciar el proyecto de express a través de Node JS, tenemos un archivo server.js en nuestra carpeta principal, el cual configura puerto, ip y elementos necesarios, basta con pedirle a Node JS que ejecute nuestro archivo y corra nuestro servidor express con el comando node.

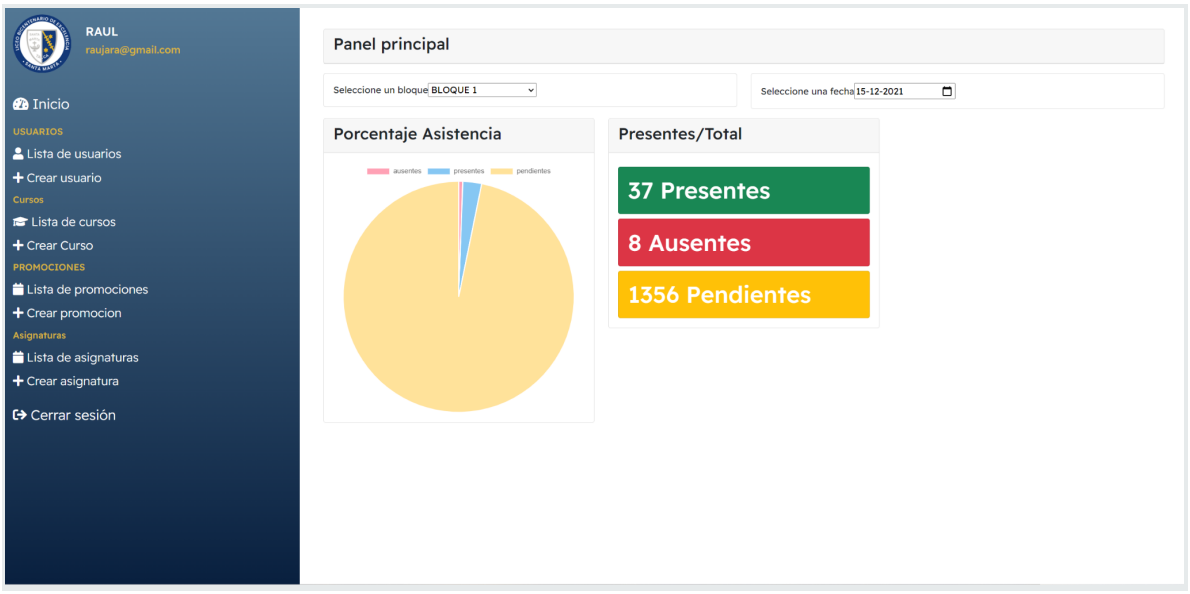
  `node server.js`



Aplicaciones

Con los comandos anteriores, ya tendríamos nuestra aplicación corriendo y funcionando para trabajar, siempre y cuando se hayan cumplido todos los pasos de este manual de manera adecuada.

`Si entramos a https://localhost:4200 veremos la aplicación principal`



`Nuestra api funciona, lo podemos ver obeniendo lista de usuarios a través de http://localhost:3000/asignaturas`


```
[
  {
    "id": 3,
    "nombre": "Música"
  },
  {
    "id": 4,
    "nombre": "Lenguaje"
  },
  {
    "id": 5,
    "nombre": "Matemáticas"
  },
```