

Instalar Laragon

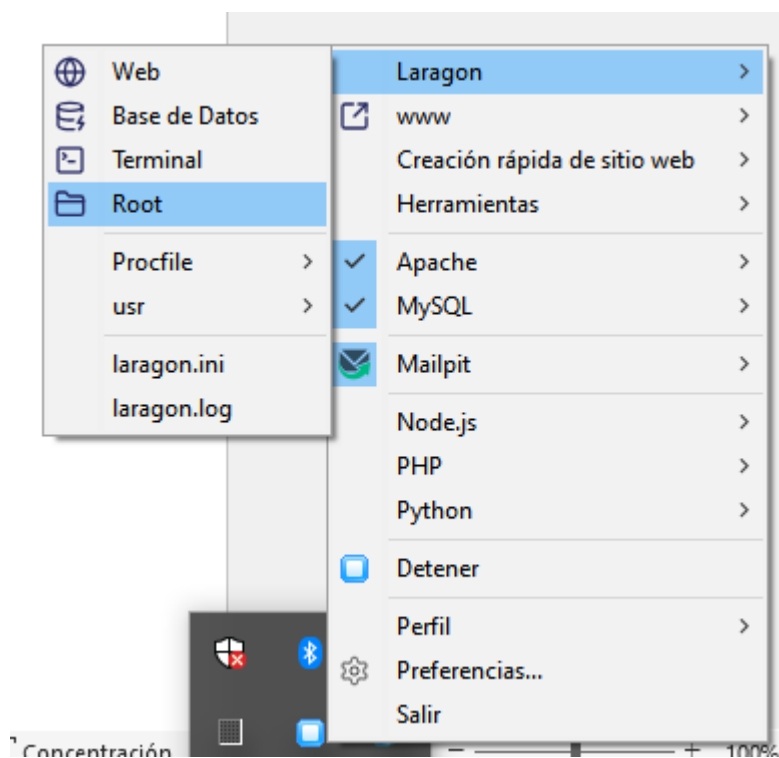
<https://github.com/leokhoa/laragon/releases/download/8.0.0/laragon-wamp.exe>

Ahora tenemos que tenemos que abrir la consola de comandos “terminal” en laragon



Escribir el comando

```
git clone https://github.com/MinoruMX/Chasap.git
```



Dentro de la carpeta “Chasap” encontraremos las siguientes carpetas

backend
frontend

Nos vamos a backend y editamos el archivo “package.json” y editamos las siguientes líneas:

```
"SCRIPTS": {  
  "START": "set NODE_OPTIONS=--openssl-legacy-provider && NODEMON DIST/SERVER.JS",  
}
```

Ahora hacemos lo mismo con la carpeta frontend y el archivo “package.json”

```
"scripts": {  
  "start": "set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start",  
  "build": "set NODE_OPTIONS=--openssl-legacy-provider && set \"GENERATE_SOURCEMAP=false\" && react-scripts build",  
},
```

Debe quedar así:

```
"scripts": {  
  "start": "set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start",  
  "build": "set NODE_OPTIONS=--openssl-legacy-provider && set \"GENERATE_SOURCEMAP=false\" && react-scripts build",  
}
```

Al terminar escribir

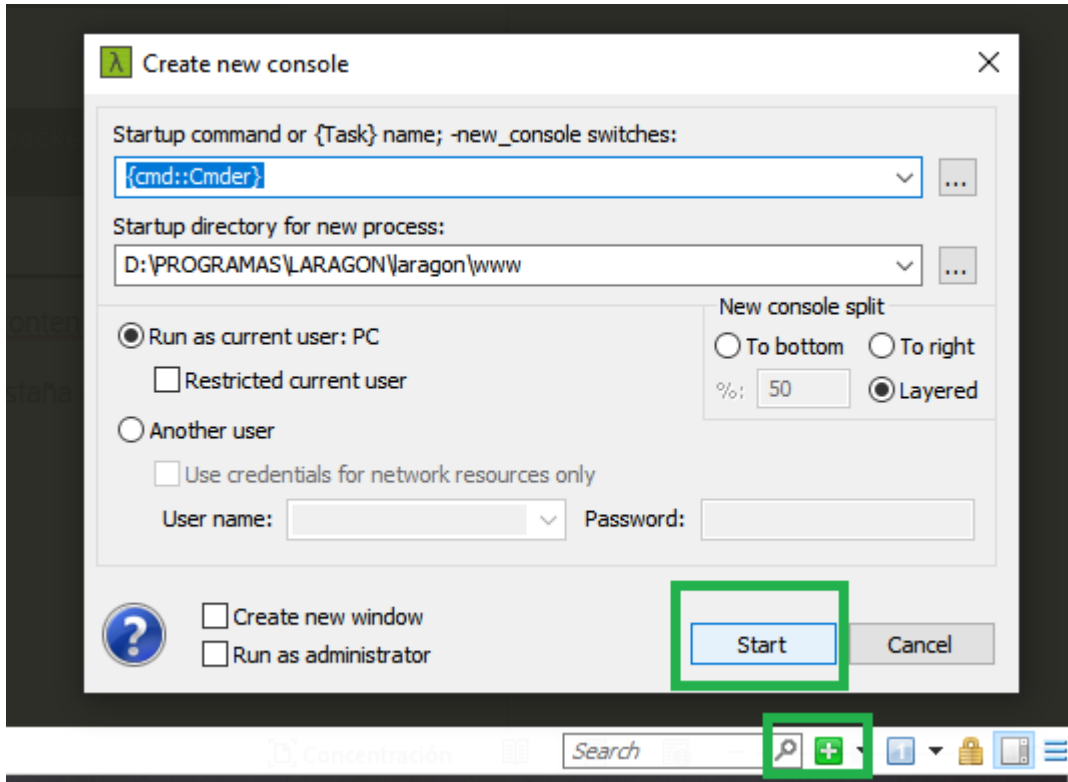


Escribir los siguientes comandos:

- cd Chasap
- cd frontend
- npm install

- npm run build
- npm run start

abrir una nueva ventana sin cerrar la anterior (deben estar corriendo ambos procesos en ventanas separadas)



cd Chasap

cd backend

npm install

npm run build

```
D:\PROGRAMAS\LARAGON\laragon\www\vipclub_5.3.5\backend (backend@1.0.0)
λ npm run start
```

Ahora debe empezar a ejecutarse el servidor.

Recuerda revisar el archivo .ENV

```
NODE_ENV=
BACKEND_URL=http://localhost:4000
FRONTEND_URL=http://localhost:3000
```

```
PROXY_PORT=443
PORT=4000
```

Ahora tenemos que hacer lo mismo de lo anterior pero con la carpeta "frontend"

Y revisar en el .ENV

```
REACT_APP_BACKEND_URL=http://localhost:4000
REACT_APP_HOURS_CLOSE_TICKETS_AUTO=24
```

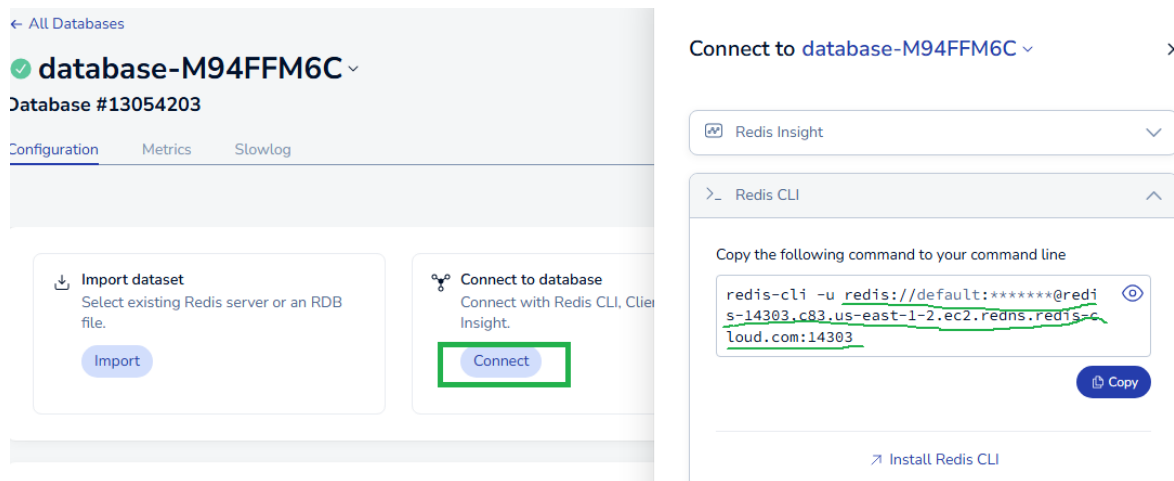
*** Recuerda nunca cerrar el servidor backend, debes abrir una nueva pestaña en frontend para que ambos servicios corran al mismo tiempo. Después solo deberas ir al navegador y escribir <http://localhost:3000/>

INICIAR REDIS:

(iniciar redis en Windows es un poco complicado, si solo requerimos hacer pruebas es mejor utilizar redis en la nube)

<https://cloud.redis.io/#/databases>

y el link resultante:



The image shows the Redis Cloud web interface. On the left, under 'All Databases', a database named 'database-M94FFM6C' is selected. Below it, there are two main options: 'Import dataset' and 'Connect to database'. The 'Connect to database' option is highlighted with a green box, and its 'Connect' button is also highlighted. On the right, a modal window titled 'Connect to database-M94FFM6C' is open. It shows the 'Redis CLI' tab selected. The command to connect is displayed: `redis-cli -u redis://default:*****@redis-14303.c83.us-east-1-2.ec2.redns.redis-cloud.com:14303`. The command is highlighted with a green box, and a 'Copy' button is visible next to it. At the bottom of the modal, there is a link to 'Install Redis CLI'.

pegarlo en background / .ENV

EJEMPLO:

```
REDIS_URI=redis://default:????????????????@redis-14303.c83.us-east-1-2.ec2.redns.redis-cloud.com:14303
```

NOTAS:

Recuerda verificar que el servidor postgres este corriendo y ejecutar los siguientes comandos

- Para crear la base de datos, utilizara el nombre del archivo ENV de la carpeta backend y la linea DB_NAME
`npx sequelize db:create`
- Para migrar todas las tablas utilizar
`npx sequelize db:migrate`
- Para rellenar con datos preestablecidos utilizar
`npx sequelize db:seed:all`
- Para borrar todos los datos de la base de datos:
`npx sequelize db:drop`