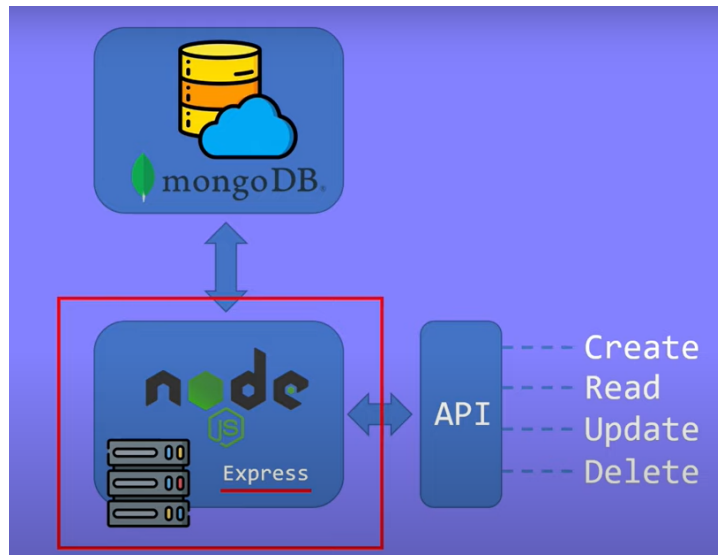


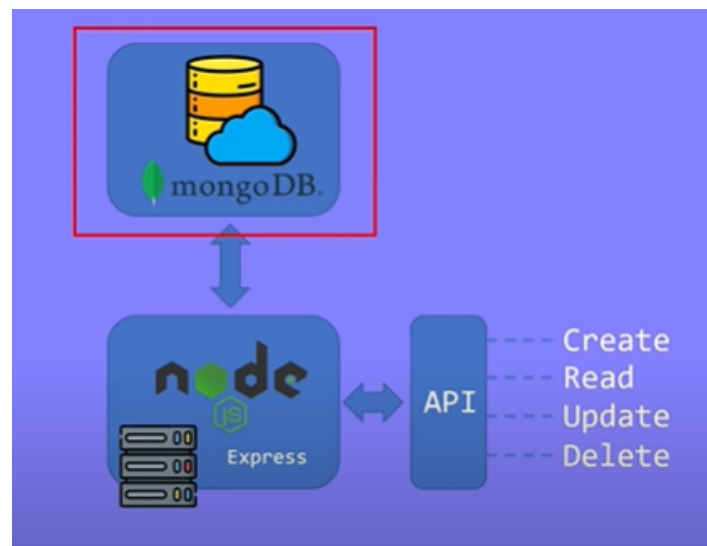
## Sesión # 13 Componente Práctico

### Bases de datos no relacionales

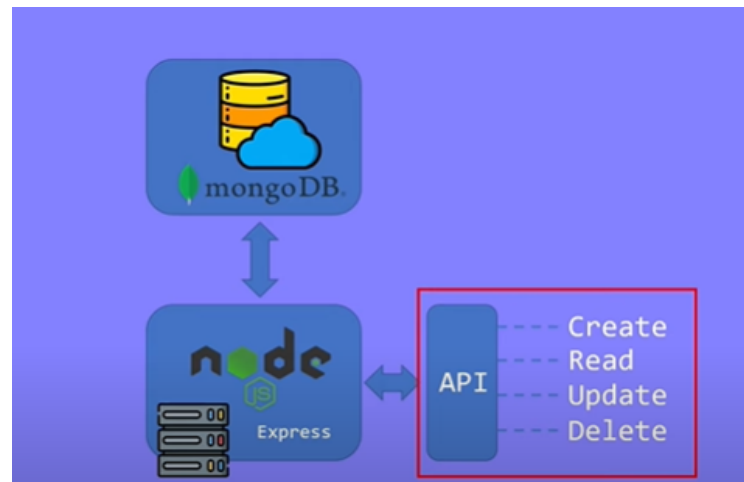
En esta sesión crearemos un base de datos mongodb y haremos la parte inicial para la construcción de una API que realizará consultas hacia la misma.:



Estaremos usando express (framework de node.js) que como ya es conocido por sesiones anteriores, nos permite levantar un servidor con pocas líneas de código.



También usaremos mongoDB como base de datos a la cual nos vamos a conectar con nuestra aplicación de node js para almacenar datos.



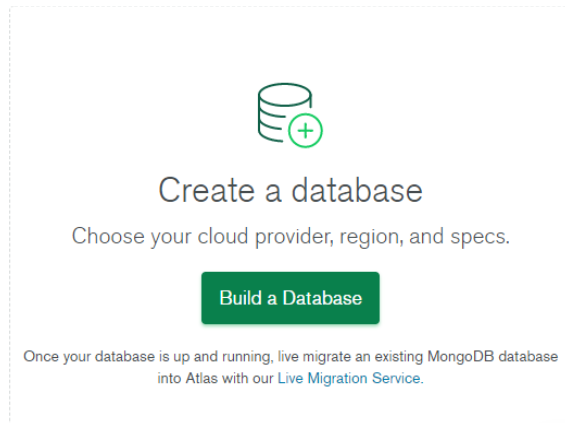
Y por último tendremos que desde nuestra API crearemos las consultas que le haremos a la base de datos.

## Paso a paso para configurar y crear base de datos mongoDB

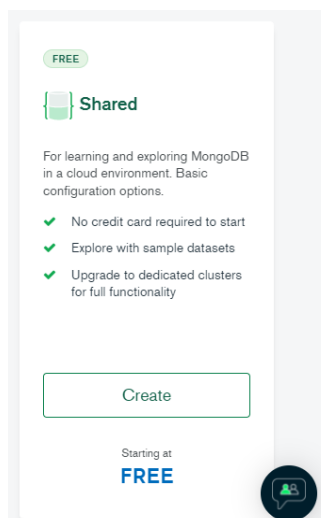
1. Crea un cuenta en mongoDB de manera gratuita ([Registro en mongo db](#))
2. Nos mostrará está vista donde nos registramos con nuestros datos.

La imagen muestra la interfaz de usuario para crear una cuenta en MongoDB Atlas. El encabezado dice "MongoDB Atlas". Debajo, se indica "Get started free" y "No credit card required". El formulario contiene los siguientes campos: "First Name" (con el valor "Jorge"), "Last Name" (con el valor "Riaño"), "Your Company (optional)", "Email Address" (con el valor "jriano23@gmail.com") y "Password (8 characters minimum)" (con caracteres ocultos por puntos). Hay un campo de verificación "I agree to the terms of service and privacy policy." con una casilla marcada. Al final, hay un botón verde que dice "Create account".

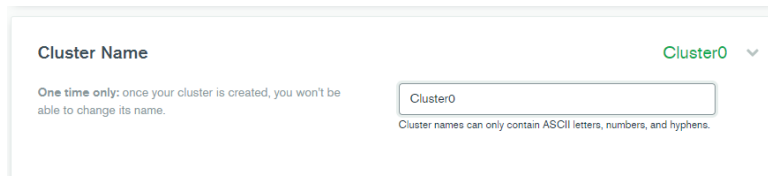
3. Luego verificamos en nuestro email la cuenta creada y listo de esa manera podemos [iniciar sesión](#) en la página.
4. Luego de iniciar sesión , crearemos nuestra primera base de datos



5. Escogemos la opción Free presionado sobre create.



6. La configuración por defecto la dejaremos igual, la única que modificaremos es la casilla llamada cluster name donde colocaremos el nombre que queramos.



7. Luego presionamos sobre Create Cluster.

8. Ahora haremos unas configuraciones adicionales y para eso presionamos sobre Database Access.

9. Luego crea un Database User el cual tendrá acceso a nuestra base de datos.



## Create a Database User

Set up database users, permissions, and authentication credentials in order to connect to your clusters.

Add New Database User

[Learn more](#)

10. Escoge como método de autenticación Password , ingresa user y password o auto generamos el password , y las demás opciones las dejamos tal como están, y presionamos sobre add user.

Authentication Method

Password	Certificate	AWS IAM (MongoDB 4.4 and up)
----------	-------------	---------------------------------

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

[SHOW](#)

11. Luego vamos a la opciones Network Access que se encuentra en la opciones que se despliegan el parte izquierda de la pantalla.



## Add an IP address

Configure which IP addresses can access your cluster.

Add IP Address

[Learn more](#)

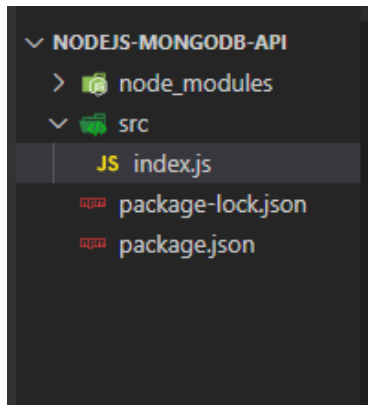
12. Luego presionamos sobre confirma.

ALLOW ACCESS FROM ANYWHERE

y luego

## Paso a paso creación de app Node.js

1. Crea un folder de proyecto y abrelo en VS code.
2. Abre una terminal en VS code y colocamos el comando `npm init --yes`.
3. Como necesitamos levantar un servidor web , utilizaremos el framework express de node js , el cual vamos a instalar en la terminal con el comando `npm i express`
4. Luego en la raíz del proyecto crea una carpeta llamada src y dentro de ella creamos un archivo js llamado index.js



5. Allí dentro del archivo index.js coloca las siguientes líneas de código

```
1 const express=require('express');| File is a CommonJS module; it may
2 const app =express();
3 const port = process.env.PORT||9000;
4 app.listen(port, ()=> console.log('server listening on port', port));
```

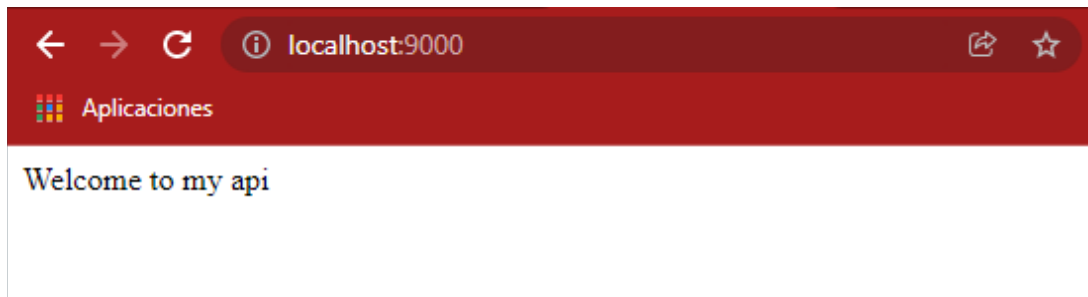
6. Luego vamos a instalar Nodemon por la terminal con el comando `npm i nodemon -D` para que cada vez que hagamos un cambio al código este se renderice de manera automática.
7. Después en el archivo package.json dentro del script colocaremos lo siguiente.

```
6 "scripts": {
7   "start": "nodemon src/index.js"
8 },
```

8. Luego para correr nuestra aplicación por la terminal usaremos el comando `npm run start` o `npm start`
9. Para probar luego si nuestro servidor responde a peticiones de un usuario colocaremos la siguiente línea de comando

```
4 // routes
5 app.get('/', (req,res)=>{ 'req' is dec
6   res.send('Welcome to my api')
7 }
8 );
```

10. Luego después abrimos el navegador y colocamos `localhost:9000` como url y debe salir lo siguiente



11. Listo, eso es todo por el momento. Ya nuestra API cuenta con conexión y tenemos configurada nuestra base de datos en MongoDB, en la siguiente sesión continuaremos con este proyecto e implementaremos métodos CRUD para hacer consultas.