Que es Node? Ejecuta del lado del servidor. Lenguaje de Backend. Corre sobre el motor v8 de google. Permite el acceso al sistema de archivos del equipo.

Que puedo hacer con Node? Uso de Sockets para una comunicacion real Cliente-Servidor. Manejo de archivos en FileSystem, cargas simultaneas. Servidores locales y remotos. Conexiones de base de datos. Creacion de servicios REST

Por que es tan popular? Sus entradas y salidas no realizan bloqueos.

Quienes lo usan? Miles de companias

Blocking vs NonBlocking => Las ejecuciones bloquantes demoran mucho mas tiempo. Las no bloqueantes que reciben como argumento un callback (una funcion como parametro).Nosotros vamos a trabajar de forma no bloquante ya que se ejuta mas rapido

NPM INIT => Lo que crea es un package.json de todos las dependencias (paquetes necesarios que tiene nuestra aplicacion) que tiene la aplicacion

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Hay que usar la libreria de node => <https://nodejs.org/es/>

<https://nodejs.dev/en/learn/writing-files-with-nodejs/>

fs.writeFile => Para escribir archivos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Instalar packetes en nuestra aplicacion. Siempre es mejor trabajar con packetes que tengan alto nivel de descarga.

npm init -y ===> Crea el package.json. Es el punto inicial de cualquier proyecto de node. A partir de ahi conocemos como correr la aplicacion o que comandos usar.

Cuando intalamos cualquier cosa se instalan en dependencia. Son paquetes necesario para que la aplicacion funcione. Ejemplo npm i colors te instala una dependencia de colores y que es necesaria en la aplicacion que creaste.

Para versiones especificas => npm i colors@1.0.0

Para actualizarla ==> npm update ====> Lo que hace es revisar las dependencias y las actualiza. Es preferible actualizar las dependencias manualmente

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

npm i yargs ===> Es para ver cuestiones de consola

MONGODB => <https://www.mongodb.com/atlas/database>

1. SIGN IN
2. INGRESAR CON GOOGLE
3. ME TENGO QUE GENERAR UN NUEVO PROYECTO (new proyect)
4. ALL CLUSTERS
5. INGRESO AL PROYECTO
6. +CREATE A DATA BASE
7. ELEGIR LA GRATUITA => SHARED
8. CAMBIAR NOMBRE DEL CLUSTER
9. SECURITY => DATABASE ACCESS
10. ADD NEW DATABASE USER => agrego el nombre y la constraseña
11. VOY AL CÓDIGO Y EN .ENV => PEGO USER: Y PASSWORD:
12. BUILD AND ROLL => USER PRIVILEGS => READ AND WRITE ANY DATABASE
13. ADD USER
14. VOY A DATABASE
15. AGREGO EL IP
16. CONECT => MONGO DB COMPASS
17. I HAVE MONGODB COMPASS INSTALLED
18. COPIAMOS LA CONECTION STRING
19. LO PEGO EN EL .ENV => MONGODB\_CNN=
20. REEMPLAZO EL USER NAME Y EL PASSWORD

DEBERIA QUEDAR ASI => PORT = 8080

USER = user\_node\_cafe

PASSWORD = gtwA8x\_pg\_BCTLj

MONGODB\_CNN = mongodb+srv://user\_node\_cafe:gtwA8x\_pg\_BCTLj@cluster0.c9anw7o.mongodb.net/

1. DESPUES PUEDO PORRAR EL USER Y EL PASSWORD (ME QUEDARIA EL PORT Y EL MONGODB\_CNN
2. LE AGREGO AL FINAL DE TODO => después del .net/café\_DB
3. VOY AL MONGODB COMPASS Y PEGO LA CONECTION STRING Y YA ESTA LA BASE DE DATOS CONECTADA
4. UTILIZAMOS <https://mongoosejs.com/>
5. NPM I MONGOOSE
6. NUEVO DIRECTORIO => DATABASE => config.js =>
7. const mongoose = require('mongoose')
8. const dbConnection = async() =>{
9. try {
10. await mongoose.connect(process.env.MONGODB\_CNN, {
11. useNewUrlParser: true,
12. useUnifiedTopology: true,
13. useCreateIndex: true,
14. useFindAndModify: false
15. })
16. } catch (error) {
17. console.log(error)
18. throw new Error('Error en base de datos')
19. }
20. }
21. module.exports = {
22. dbConnection
23. }
24. LLAMO DENTRO DE models => serve.js => esa función =>

async conectarDb(){

    await dbConnection

  }

Texto

Descripción generada automáticamente