Node.js: Es un entorno de ejecución de JavaScript para orientado a eventos asíncronos, o sea, el proceso principal de la ejecución no se detiene por un proceso en paralelo.

Express: Se utiliza para desarrollar servidores, en donde el código es mas conciso, es un framework web de node.js

Su propósito es ayudar a crear aplicaciones de node.js en donde sus características son:

1. Ofrece mecanismos para definir creación de rutas mas sencillo que con node sin utilizar verbos http.
2. Se enfoca al alto rendimiento desarrollando aplicaciones de node.js

Conceptos básicos de express:

1. CRUD : Acronimo que representan las operaciones básicas que podemos realizar con la información almacenada en la base de datos, la cual están relacionadas con los métodos HTTP y APIs (Create, Read, Update, Detelete)
2. Son equivalentes en verbos o métodos HTTP:

Create recurso → POST

Read recurso u obtener un recurso → GET

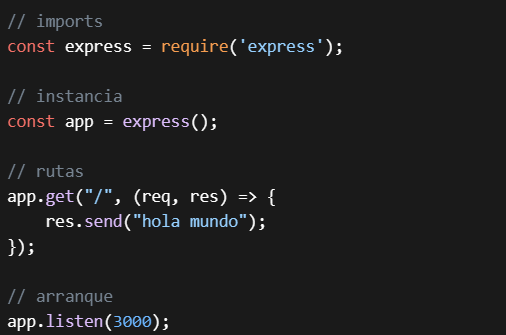
Update recurso → PUT

Delete recurso → DELETE

API = Interfaz de Programacion de aplicaciones: Permite que dos o más programas se comuniquen entre si, cliente con el servidor, son usadas por el programador que usa la API para implementar su programa.

La API solicita información al servicio que ofrece esa información, enviando esa información al cliente que lo solicite.

REST → Es un estilo de arquitectura de software para sistemas hipermedia distribuidos como la Internet.



1. **Imports**: Se importa el módulo de Express utilizando require.
2. **Instancia**: Se crea una instancia de una aplicación Express llamada app. Esta instancia se utiliza para configurar el servidor, rutas, middleware, etc.
3. **Rutas**: Se define una ruta para la raíz del sitio web ("/"). Cuando un cliente realiza una solicitud HTTP GET a la raíz, el servidor responderá con el texto “hola mundo”.
4. **Arranque**: Se inicia el servidor para que escuche en el puerto 3000. Esto significa que el servidor estará accesible a través de http://localhost:3000 en un navegador web o mediante una solicitud HTTP a ese puerto.

Con este código, has creado un servidor web simple que responde con “hola mundo” a las solicitudes en la raíz del sitio. Es un buen punto de partida para empezar a construir aplicaciones web más complejas con Express.

nodemon → dependencia de desarrollo para reiniciar el código por nosotros

script → "dev": "nodemon app.js" para que cada que haga un cambio en el código lo reinicie por si solo.

createPool → Es una función especial de conjunto de conexiones que podemos usar en producción fácilmente.