Esquema de funcionamiento de la aplicación NodeJS

📌 Esquema de Funcionamiento de la Aplicación

```
📁 Proyecto
|── 📁 public
|   ├── index.html   → Vista principal de la aplicación (frontend)
|
|── 📁 node_modules  → Módulos instalados de Node.js
📁 database
|── database.js      → Conexión con SQLite y creación de la tabla 📁
routes
|── tasks.js         → Definición de las rutas (APIs)
|── index.js         → Archivo principal que inicia el servidor
|── package.json     → Dependencias y scripts de la aplicación
|── package-lock.json → Archivo de bloqueo de dependencias
```

🏗️ Flujo de Funcionamiento

1. **Inicio del Servidor (index.js)**
   - Carga los módulos (`express`, `sqlite3`, `cors`, etc.).
   - Configura `express`.
   - Importa `tasks.js` y lo usa como middleware de rutas.
   - Levanta el servidor en un puerto específico.
2. **Conexión a la Base de Datos (database.js)**
   - Utiliza `sqlite3` para conectarse a una base de datos SQLite.
   - Crea la tabla `tasks` si no existe.
   - Exporta la conexión a la base de datos para ser usada en `tasks.js`.
3. **Gestión de Rutas (tasks.js)**
   - Define endpoints para:
     - Obtener todas las tareas (`GET /`).
     - Agregar una nueva tarea (`POST /`).
     - Actualizar una tarea (`PUT /:id`).
     - Eliminar una tarea (`DELETE /:id`).
   - Usa la conexión de `database.js` para ejecutar consultas en SQLite.
4. **Interfaz de Usuario (index.html en /public)**
   - Se comunica con la API mediante `fetch` para obtener, agregar, actualizar y eliminar tareas.
   - Renderiza las tareas en la pantalla.
   - Puede tener eventos en botones para interactuar con el backend.

Esquema de funcionamiento de la aplicación NodeJS

🔄 Ejemplo de Flujo de una Petición

1. El usuario abre `index.html` en el navegador.
2. Se hace un `fetch` a `GET /` para obtener la lista de tareas y mostrarlas.
3. El usuario agrega una nueva tarea en el formulario.
4. Se envía un `POST /` con los datos de la nueva tarea.
5. `tasks.js` recibe la petición y la guarda en SQLite.
6. Se responde con la nueva lista de tareas actualizada.
7. La interfaz se actualiza en el navegador sin recargar la página.