

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS – UNIMINUTO

Evidencias funcionamiento backend local
Aplicación de tareas

Duvan Alonso Leon Daza
Ingeniería Web II
NRC 45-55290

Nombre del tutor:
Jose Danilo Sanchez Torres

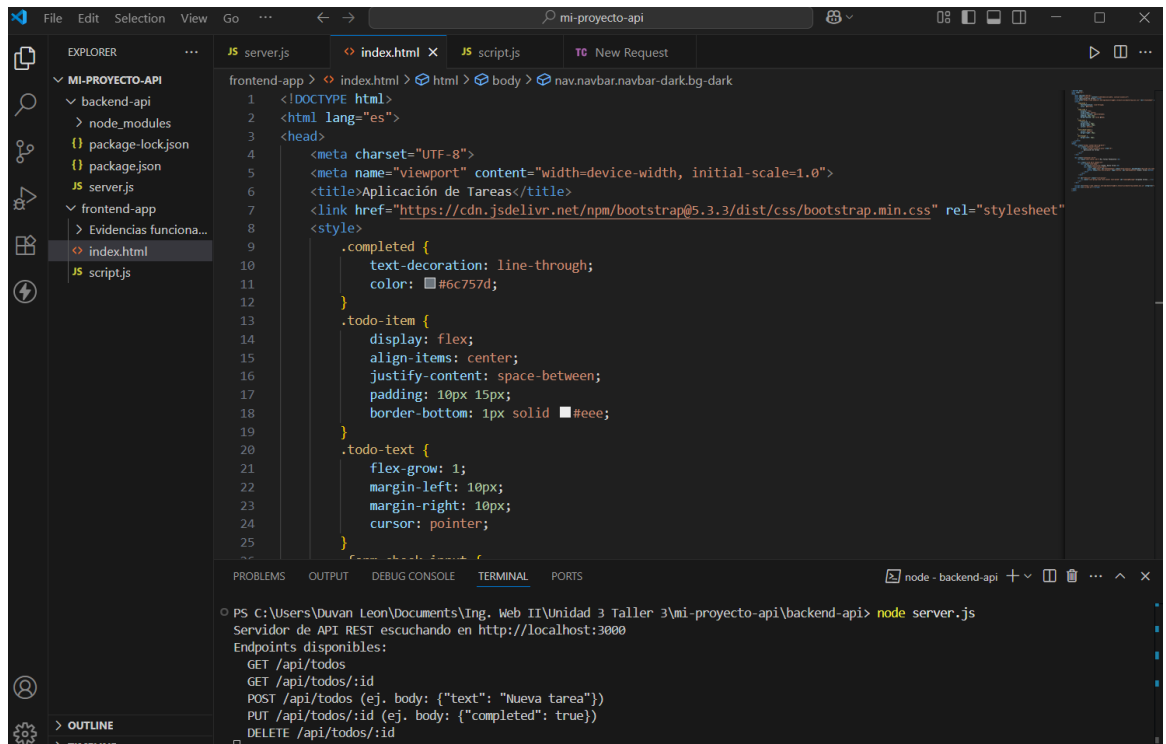
Bogotá – 15 de Junio 2025

Contexto

En GitHub Pages no es posible desplegar y ejecutar el back-end de Node.js directamente. GitHub Pages está diseñado para alojar sitios web estáticos (HTML, CSS, JavaScript del lado del cliente). Un back-end de Node.js es una aplicación de servidor que necesita un entorno de ejecución específico (Node.js) y un proceso que se mantenga activo escuchando solicitudes, algo que GitHub Pages no proporciona.

Ya que el proyecto está cargado completamente en github, desde este archivo envío las evidencias del funcionamiento del backend de forma local a través del servidor node.js y que la API responde a diferentes tipos de solicitudes HTTP (GET, POST, PUT, DELETE) en Thunder Client.

Ejecución exitosa de node server.js

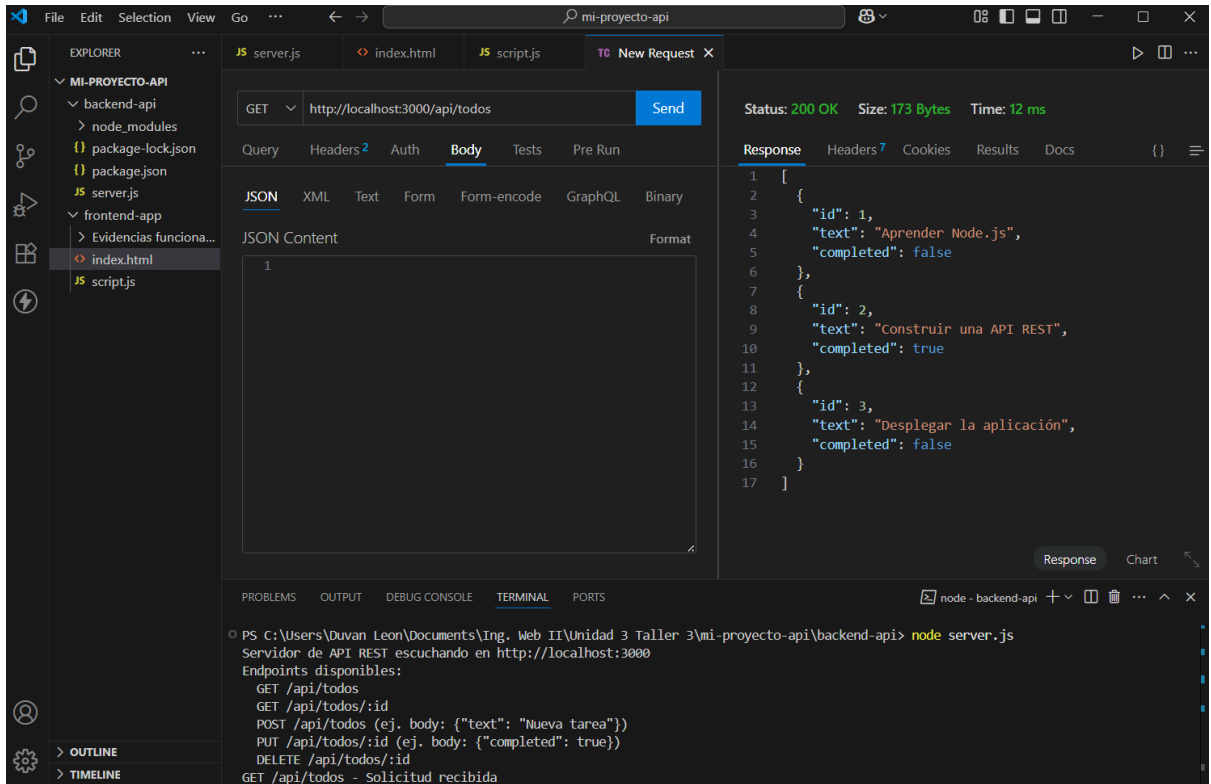


The screenshot shows the Visual Studio Code interface with a project named 'mi-proyecto-api'. The Explorer sidebar on the left shows the file structure, including 'backend-api', 'node_modules', 'package-lock.json', 'package.json', 'server.js', 'frontend-app', and 'index.html'. The main editor displays the 'index.html' file, which contains HTML boilerplate and CSS styles for a todo application. The terminal at the bottom shows the command 'node server.js' being executed in the 'backend-api' directory, resulting in a successful message: 'Servidor de API REST escuchando en http://localhost:3000'. Below this, the available endpoints are listed: GET /api/todos, GET /api/todos/:id, POST /api/todos (ej. body: {"text": "Nueva tarea"}), PUT /api/todos/:id (ej. body: {"completed": true}), and DELETE /api/todos/:id.

```
front-end-app > index.html > html > body > nav.navbar.navbar-dark.bg-dark
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Aplicación de Tareas</title>
7    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8  </head>
9  <body>
10   <div class="container">
11     <div class="row">
12       <div class="col">
13         <div class="card">
14           <div class="card-body">
15             <div class="text-center">
16               <h2>Aplicación de Tareas</h2>
17             </div>
18             <div class="list-group">
19               <div class="list-item">
20                 <div class="d-flex justify-content-between align-items-center">
21                   <div class="flex-grow 1">
22                     <div class="text-decoration: line-through; color: #6c757d; font-weight: bold; font-size: 1.2em; margin-bottom: 5px;">
23                       <span>Tarea pendiente</span>
24                     </div>
25                   </div>
26                 </div>
27               </div>
28             </div>
29           </div>
30         </div>
31       </div>
32     </div>
33   </div>
34 </body>
35 </html>
```

```
PS C:\Users\Duvan Leon\Documents\Ing. Web II\Unidad 3 Taller 3\mi-proyecto-api\backend-api> node server.js
Servidor de API REST escuchando en http://localhost:3000
Endpoints disponibles:
GET /api/todos
GET /api/todos/:id
POST /api/todos (ej. body: {"text": "Nueva tarea"})
PUT /api/todos/:id (ej. body: {"completed": true})
DELETE /api/todos/:id
```

1. Prueba 1: Obtener Todas las Tareas (GET)



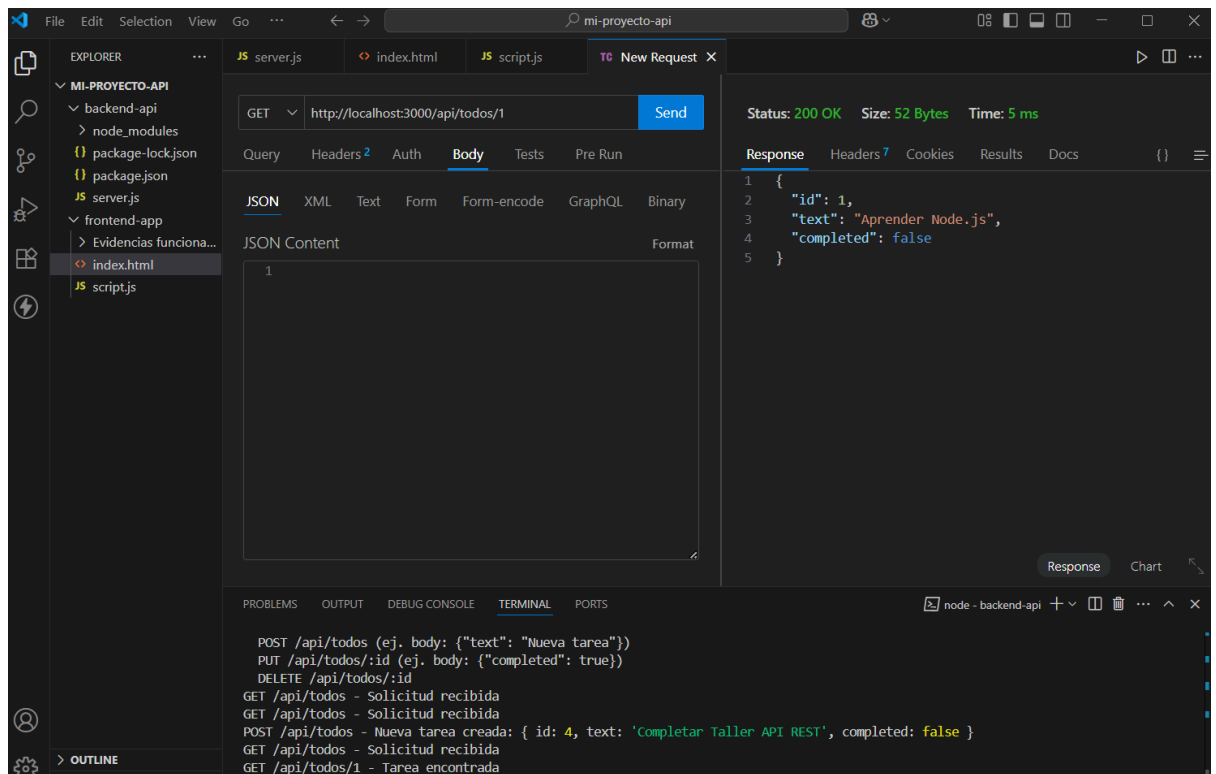
2. Prueba 2: Crear una Nueva Tarea (POST)

The top part of the image shows a VS Code editor with a REST client request configured. The request is a POST to `http://localhost:3000/api/todos` with a JSON body: `{ "text": "Completar Taller API REST" }`. The response status is 201 Created, with a size of 61 Bytes and a time of 110 ms. The response body is: `{ "id": 4, "text": "Completar Taller API REST", "completed": false }`. The bottom part of the image shows a web browser at `127.0.0.1:5500/frontend-app/index.html` displaying the "Aplicación de Tareas" interface. The page title is "Mis Tareas Pendientes". It features a form to "Añadir Nueva Tarea" and a list of tasks:

- ☐ Aprender Node.js
- ☒ Construir una API REST
- ☐ Desplegar la aplicación
- ☐ Completar Taller API REST

Each task has an "Eliminar" button next to it.

3. Prueba 3: Obtener una Tarea por ID (GET :id)



4. Prueba 4: Actualizar una Tarea (PUT):

The image shows a development environment with VS Code and a web browser. In VS Code, a REST client request is being sent to `http://localhost:3000/api/todos/1` using the PUT method. The request body is a JSON object: `{ "completed": true }`. The response status is 200 OK, with a size of 51 Bytes and a time of 15 ms. The response body is: `{ "id": 1, "text": "Aprender Node.js", "completed": true }`. The terminal shows the following log output:

```
PUT /api/todos/:id (ej. body: {"completed": true})
DELETE /api/todos/:id
GET /api/todos - Solicitud recibida
GET /api/todos - Solicitud recibida
POST /api/todos - Nueva tarea creada: { id: 4, text: 'Completar Taller API REST', completed: false }
GET /api/todos - Solicitud recibida
GET /api/todos/1 - Tarea encontrada
PUT /api/todos/1 - Tarea actualizada: { id: 1, text: 'Aprender Node.js', completed: true }
```

The web browser shows the application at `127.0.0.1:5500/frontend-app/index.html`. The page title is "Aplicación de Tareas".

Mis Tareas Pendientes

Añadir Nueva Tarea

Añadir Tarea

☒ Aprender-Node.js

☒ Construir una API-REST

☐ Desplegar la aplicación

☐ Completar Taller API REST

Eliminar

Eliminar

Eliminar

Eliminar

5. Prueba 5: Eliminar una Tarea (DELETE):

The screenshot shows the VS Code interface with a REST client request configured for a DELETE operation on `http://localhost:3000/api/todos/4`. The request body is a JSON object: `{ "completed": true }`. The status bar indicates a successful response: `Status: 204 No Content`, `Size: 0 Bytes`, and `Time: 15 ms`.

The terminal window displays the following log output:

```
GET /api/todos - Solicitud recibida
GET /api/todos - Solicitud recibida
POST /api/todos - Nueva tarea creada: { id: 4, text: 'Completar Taller API REST', completed: false }
GET /api/todos - Solicitud recibida
GET /api/todos/1 - Tarea encontrada
PUT /api/todos/1 - Tarea actualizada: { id: 1, text: 'Aprender Node.js', completed: true }
GET /api/todos - Solicitud recibida
DELETE /api/todos/4 - Tarea eliminada
```

The web browser shows the application running at `127.0.0.1:5500/frontend-app/index.html`. The page title is "Aplicación de Tareas" and the main heading is "Mis Tareas Pendientes".

Mis Tareas Pendientes

Añadir Nueva Tarea

Añadir Tarea

☒ Aprender Node.js

☒ Construir una API REST

☐ Desplegar la aplicación

Eliminar

Eliminar

Eliminar