

UNIVERSIDAD DE INGENIERÍA Y  
TECNOLOGÍA



**UTECH**  
UNIVERSIDAD DE INGENIERÍA  
Y TECNOLOGÍA

## EJERCICIO PROPUESTO

CLOUD COMPUTING

**GitHub Repository :**

<https://github.com/KarTaGo124/api-books>

Jose Guillermo Galvez Pacori

## 1 Cloning the repository and installing packages

```
:~ $ pwd
/home/ubuntu
:~ $ ls
environment node_modules package-lock.json package.json python3 utec
:~ $ mkdir node
:~ $ cd node
:~/node $ git clone https://github.com/KarTaGo124/api-books.git
Cloning into 'api-books'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 0), reused 9 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 24.16 KiB | 4.03 MiB/s, done.
:~/node $ ls
api-books
:~/node $ cd api-books/
:~/node/api-books (main) $ ls
README.md books.js db.js index.js package-lock.json package.json postman.json
```

```
:~/node/api-books (main) $ npm install express sqlite3
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated npmlog@6.0.2: This package is no longer supported.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm warn deprecated are-we-there-yet@3.0.1: This package is no longer supported.
npm warn deprecated gauge@4.0.4: This package is no longer supported.

added 189 packages, and audited 190 packages in 6s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 10.8.2 -> 10.8.3
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.3
npm notice To update run: npm install -g npm@10.8.3
npm notice
```

## 2 Opening port 8001

i-05cfd2e353319fb54 (MV desarrollo)

▼ Reglas de entrada

Q Filtrar reglas

Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen	Grupos de seguri
-	sgr-0b8716d05ae093726	80	TCP	0.0.0.0/0	<a href="#">launch-wizard-1</a>
-	sgr-0af76cd247343458e	8001	TCP	0.0.0.0/0	<a href="#">launch-wizard-1</a>
-	sgr-08bc29eada9d657f	22	TCP	0.0.0.0/0	<a href="#">launch-wizard-1</a>
-	sgr-0bd19cf2dc3f88623	8000	TCP	0.0.0.0/0	<a href="#">launch-wizard-1</a>

## 3 Running API

```
:~/node/api-books (main) $ node index.js
Server is running on port 8001
Conectado a la base de datos SQLite.
Tabla "books" lista.
[]
```

## 4 Testing CRUD in Postman

Books API

GET	Get All Books
GET	Get Book by ID
POST	Add New Book
PUT	Update Book
DEL	Delete Book

## 4.1 POST

The screenshot shows a REST client interface for a "Books API". The request is a POST to "http://(IP):8001/books". The body is a JSON object:

```
{  "title": "New Book",  "author": "Author Name",  "year": 2024}
```

The response is a 201 Created status with a JSON body:

```
{  "id": 1,  "title": "New Book",  "author": "Author Name",  "year": 2024}
```

## 4.2 GET ALL

The screenshot shows a REST client interface for a "Books API". The request is a GET to "http://(IP):8001/books". The response is a 200 OK status with a JSON array of two book objects:

```
[  {    "id": 1,    "title": "New Book",    "author": "Author Name",    "year": 2024  },  {    "id": 2,    "title": "New Book 2",    "author": "Author Name 2",    "year": 2021  }]
```

## 4.3 GET BY ID

The screenshot shows a Postman interface for a GET request named "Get Book by ID". The URL is `http://(IP):8001/books/id`. A path variable `id` is defined with the value `1`. The response is a 200 OK status with a 100 ms response time and 297 B of data. The JSON body is as follows:

```
1 {
2   "id": 1,
3   "title": "New Book",
4   "author": "Author Name",
5   "year": 2024
6 }
```

## 4.4 PUT

The screenshot shows a Postman interface for a PUT request named "Update Book". The URL is `http://(IP):8001/books/id`. The request body is a JSON object with the following details:

```
1 {
2   "id": "1",
3   "title": "Updated Book",
4   "author": "Updated Author",
5   "year": 2025
6 }
```

The response is a 200 OK status with a 221 ms response time and 306 B of data. The JSON body is as follows:

```
1 {
2   "id": "1",
3   "title": "Updated Book",
4   "author": "Updated Author",
5   "year": 2025
6 }
```

## 4.5 DELETE

The screenshot shows a REST client interface with a dark theme. At the top, the tab is labeled "DEL Delete Book". The URL bar shows "http://{{IP}}:8001/books/id" with a "Send" button. A dropdown menu is open for the "IP" variable, showing "INITIAL 44.211.222.103", "CURRENT 44.211.222.103", and "SCOPE Collection". Below the URL bar, there are sections for "Params", "Authorization", "Path Variables", and "Body". The "Path Variables" section contains a table with one row: "id" with value "1". The "Body" section shows a "204 No Content" response with a status of "204 No Content", a response time of "267 ms", and a size of "134 B". The response is displayed in "Pretty" format.

**DELETE** ⌵ http://{{IP}}:8001/books/id Send ⌵

**Params** • **Authorization** ⌵ Settings Cookies

Key	Value	Description	⋮	Bulk Edit
Key				

**Path Variables**

Key	Value	Description	⋮	Bulk Edit
id	1	Description		

**Body** Cookies Headers (4) Test Results 204 No Content • 267 ms • 134 B ⌵ Save Response ⋮

**Pretty** Raw Preview Visualize Text ⌵ ⌵ ⌵

1