



## PROGRAMACIÓN PARA INTERNET

### ENTREGA DE PARCIAL 2

ALUMNO:

IGOR IMPROTA MARTINEZ DA SILVA

MAYO DE 2024

MAESTRO:

FABRICIO A. SUAREZ DOMINGUEZ

## Contenido

1.	Definición del Objetivo .....	3
2.	Base de Datos .....	3
3.	Funcionalidades.....	3
4.	Capturas de Pantalla de los Endpoints .....	4
5.	Repositorio GitHub .....	11

## 1. Definición del Objetivo

El objetivo de este proyecto es hacer el backend de mi página web “BlogMyMetal”, donde haré un foro con los comentarios de usuarios que frecuentan mi blog.

## 2. Base de Datos

El programa autogenera las 2 colecciones en MongoDB: “usuarios” y “comentarios”.

La colección “usuarios” tiene los datos de un usuario: nombre, e-mail, contraseña y un identificador de administrador del blog. Fue creado un administrador “Admin”, que sólo el programador tiene acceso.

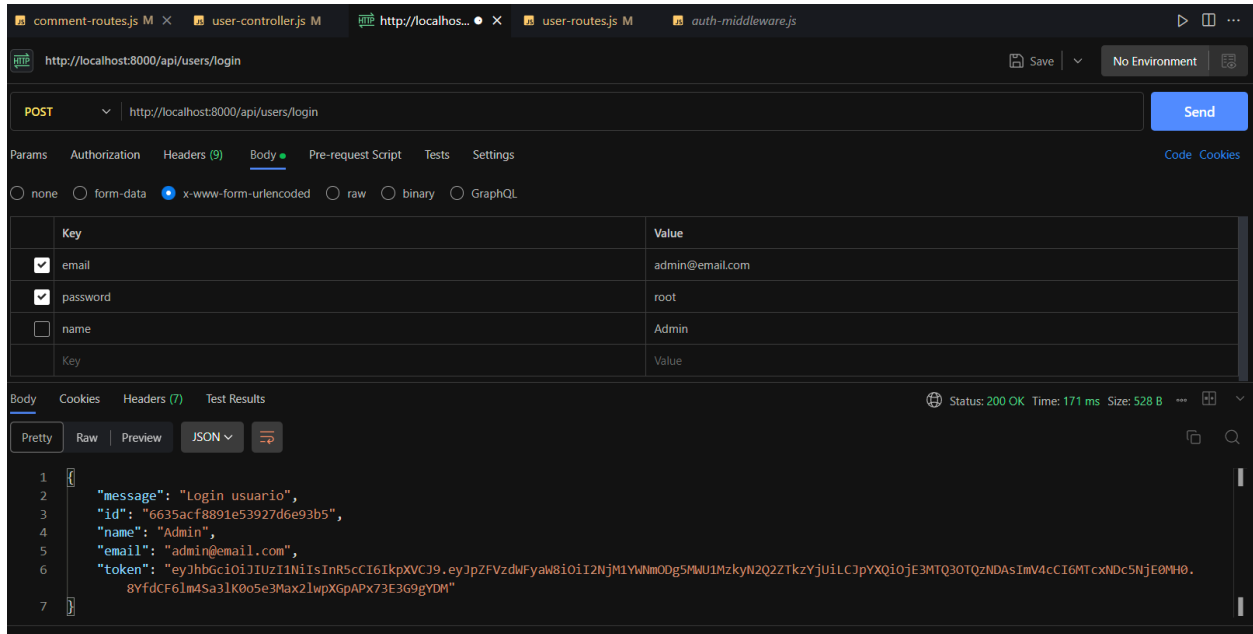
La colección “comentarios” tiene 2 atributos: “text” (el comentario) y un identificador de usuario, para que ese usuario o el administrador puedan borrar el comentario.

## 3. Funcionalidades

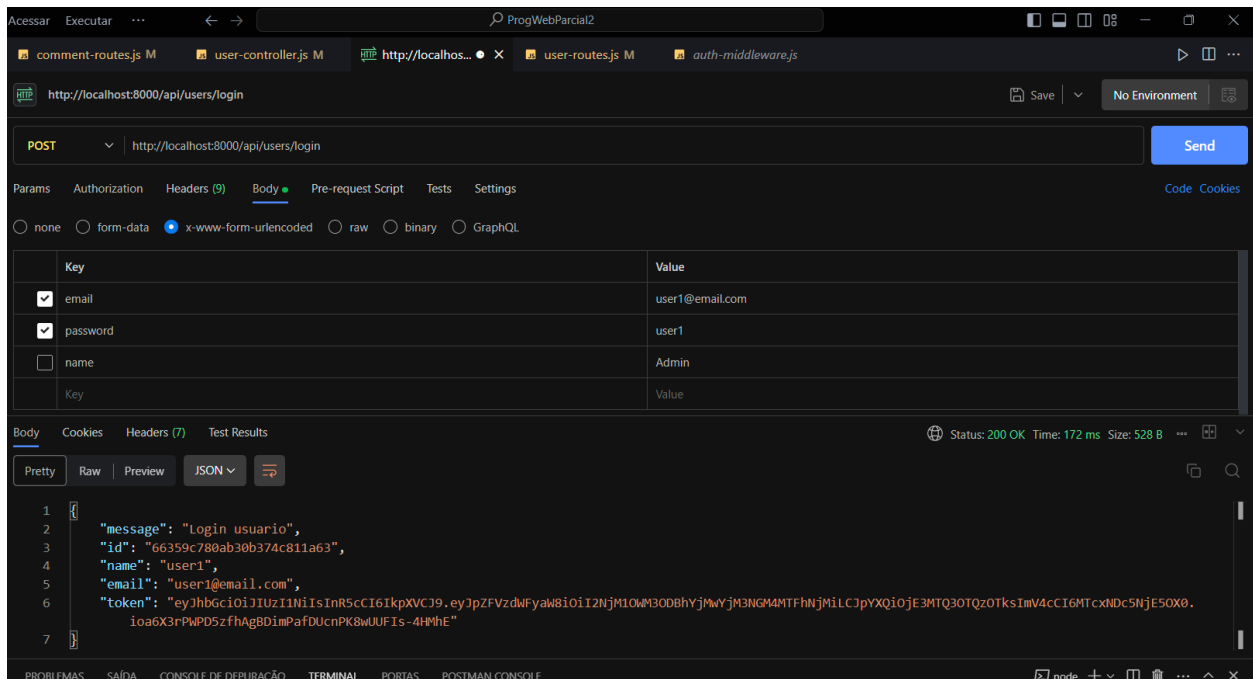
El programa permite que se generen usuarios nuevos, que se haga “login”, que se borre un usuario, que el usuario pueda comentar algo en el foro, modificar su comentario y borrar su comentario. Un administrador puede controlar todo de los usuarios y checar los comentarios por usuario. En seguida, serán mostradas pruebas de los *endpoints*.

#### 4. Capturas de Pantalla de los Endpoints

Login Admin



## Login Usuario común



## Creando comentario con User1

The screenshot shows the Postman interface for a POST request to `http://localhost:8000/api/comments`. The request is configured with the `x-www-form-urlencoded` body type. The body contains a single key-value pair: `text` with the value `primer comentario`. The response status is `201 Created`, with a time of `219 ms` and a size of `422 B`. The response body is displayed in JSON format:

```
{  "user": "66359c780ab30b374c811a63",  "text": "primer comentario",  "id": "6635b2ed58540864f9bc3b8d",  "createdAt": "2024-05-04T04:00:45.896Z",  "updatedAt": "2024-05-04T04:00:45.896Z",  "_v": 0}
```

## Checando comentario con User1

The screenshot shows the Postman interface for a GET request to `http://localhost:8000/api/comments`. The request is configured with the `x-www-form-urlencoded` body type. The response status is `200 OK`, with a time of `162 ms` and a size of `419 B`. The response body is displayed in JSON format:

```
{  "_id": "6635b2ed58540864f9bc3b8d",  "user": "66359c780ab30b374c811a63",  "text": "primer comentario",  "createdAt": "2024-05-04T04:00:45.896Z",  "updatedAt": "2024-05-04T04:00:45.896Z",  "_v": 0}
```

## Checando comentario del User 1 con Admin

HTTP GET http://localhost:8000/api/comments/66359c780ab30b374c811a63

Params Authorization Headers (8) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value
-----	-------

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {
2   {
3     "_id": "6635b2ed58540864f9bc3b8d",
4     "user": "66359c780ab30b374c811a63",
5     "text": "primer comentario",
6     "createdAt": "2024-05-04T04:00:45.896Z",
7     "updatedAt": "2024-05-04T04:00:45.896Z",
8     "__v": 0
9   }
10 }
```

## Checando todos los comentarios con Admin

comment-routes.js M comment-model.js M user-controller.js M http://localhost:8000/api/comments Save No Environment

HTTP GET http://localhost:8000/api/comments Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Code Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value
-----	-------

Body Cookies Headers (7) Test Results Status: 200 OK Time: 162 ms Size: 606 B

Pretty Raw Preview JSON

```
1 {
2   {
3     "_id": "6635b2ed58540864f9bc3b8d",
4     "user": "66359c780ab30b374c811a63",
5     "text": "primer comentario",
6     "createdAt": "2024-05-04T04:00:45.896Z",
7     "updatedAt": "2024-05-04T04:00:45.896Z",
8     "__v": 0
9   },
10  {
11    "_id": "6635b39f58540864f9bc3b96",
12    "user": "6635b35858540864f9bc3b92",
13    "text": "Comentario del user 2",
14    "createdAt": "2024-05-04T04:03:43.853Z",
15    "updatedAt": "2024-05-04T04:03:43.853Z",
16    "__v": 0
17  }
18 }
```

PROBLEMAS SAÍDA CONSOLE DE DEPURACIÓN TERMINAL PORTAS POSTMAN CONSOLE node

## Modificando el comentario del User 1 con User 1

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** `http://localhost:8000/api/comments/6635b2ed58540864f9bc3b8d`
- Body Type:** x-www-form-urlencoded
- Status:** 200 OK
- Time:** 200 ms
- Size:** 453 B

The response body is displayed in JSON format:

```
1 {
2   "commentUpdated": {
3     "_id": "6635b2ed58540864f9bc3b8d",
4     "user": "66359c780ab30b374c811a63",
5     "text": "Intentando modificar un comentario",
6     "createdAt": "2024-05-04T04:00:45.896Z",
7     "updatedAt": "2024-05-04T04:17:13.222Z",
8     "__v": 0
9   }
10 }
```

## Intentando modificar el comentario del User 1 con el User 2

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** `http://localhost:8000/api/comments/6635b2ed58540864f9bc3b8d`
- Body Type:** x-www-form-urlencoded
- Body Fields:**

Key	Value
<input type="checkbox"/> email	user1@email.com
<input type="checkbox"/> password	user1
<input checked="" type="checkbox"/> text	Intentando modificar un comentario
- Status:** 400 Bad Request
- Time:** 41 ms
- Size:** 303 B

The response body is displayed in JSON format:

```
1 {
2   "message": "Acceso no autorizado, usuario no autenticado."
3 }
```

## Borrando el comentario del User 2 con User 2

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/api/comments/6635bfb694d83d8b67bdaae0`
- Method:** `DELETE`
- Body:** `{ "commentDeleted": { "_id": "6635bfb694d83d8b67bdaae0", "user": "6635b35858540864f9bc3b92", "text": "comentario del user 2", "createdAt": "2024-05-04T04:55:18.908Z", "updatedAt": "2024-05-04T04:55:18.908Z", "__v": 0 } }`
- Status:** 203

## Borrando el comentario del User 1 con User 2

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/api/comments/6635b2ed58540864f9bc3b8d`
- Method:** `DELETE`
- Body:** `{ "message": "No tienes permiso para eliminar este comentario.", "stack": null }`



## Borrando el comentario del User 1 con el Admin

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** `http://localhost:8000/api/comments/6635b2ed58540864f9bc3b8d`
- Authorization:** Bearer Token `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Status:** 203 Non-Authoritative Information (since HTTP/1.1)
- Response Body (JSON):**

```
1 {
2   "commentDeleted": {
3     "_id": "6635b2ed58540864f9bc3b8d",
4     "user": "66359c780ab30b374c811a63",
5     "text": "Intentando modificar un comentario",
6     "createdAt": "2024-05-04T04:00:45.896Z",
7     "updatedAt": "2024-05-04T04:18:57.275Z",
8     "__v": 0
9   }
10 }
```

## Verificando User 1 con User 1

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/api/users/data`
- Method:** `GET`
- Body Type:** `x-www-form-urlencoded` (selected)
- Body:** (Empty)
- Response Body:** A JSON object representing a user:

```
{  "_id": "66359c780ab30b374c811a63",  "name": "user1",  "email": "user1@email.com",  "esAdmin": false,  "createdAt": "2024-05-04T02:24:56.293Z",  "updatedAt": "2024-05-04T02:24:56.293Z",  "__v": 0}
```

## Verificando User 1 con Admin

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8000/api/users/data/66359c780ab30b374c811a63`
- Method:** `GET`
- Authorization:** Bearer Token `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`
- Body:** Pretty JSON view showing the response:

```
1 {
2   "user": {
3     "_id": "66359c780ab30b374c811a63",
4     "name": "user1",
5     "email": "user1@email.com",
6     "password": "$2a$10$kIVY2jjVuG/5Bj52NRgTZ.8csDjZYhgoamld9kMN9Ny6pM5NqDav0",
7     "esAdmin": false,
8     "createdAt": "2024-05-04T02:24:56.293Z",
9     "updatedAt": "2024-05-04T02:24:56.293Z",
10    "_v": 0
11  },
12  "comments": []
13 }
```
- Status:** 200

## 5. Repositorio GitHub

<https://github.com/igor-ims/ProgWebParcial2.git>