

**Escuela Politécnica Nacional**

## Informe 7

## Capturas de los métodos HTTP para json-server

Primero se hace la consulta con GET para poder obtener todos los usuarios, y un usuario importante para poder observar que el PUT que es para actualizar funcione correctamente es el usuario 1 Xavier Anatoa.

The screenshot shows the Postman interface with the following details:

- Request Method:** POST
- URL:** http://localhost:8000/usuarios
- Body:** raw JSON (shown in the code block)
- Response Status:** 201 Created
- Response Body:** A JSON object representing the newly created user, identical to the one sent in the body.

```

1  {
2   "id": "3",
3   "nombre": "Angelo Conterón",
4   "email": "angelo.conteron@epn.edu.ec",
5   "contra": "12345",
6   "preferencias": {},
7   "intereses": {
8     "tags": []
9   },
10  "notificaciones": {}

```

Ahora para poder ingresar un usuario tenemos que observar el ultimo id ingresado para usuarios y se ingresa en el body todos los campos respectivos para usuario.

The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** http://localhost:8000/usuarios
- Headers:** (7)
- Response Status:** 200 OK
- Response Body:** A JSON array containing the user object from the previous screenshot.

```

1  [
2   {
3    "id": "3",
4    "nombre": "Angelo Conterón",
5    "email": "angelo.conteron@epn.edu.ec",
6    "contra": "12345",
7    "preferencias": {},
8    "intereses": {
9      "tags": []
10     },
11    "notificaciones": {
12      "alerta24h": true,
13      "alerta1h": false,
14      "cambios": true,
15      "cercaos": true
16    },
17    "permisoUbicacion": false
18  }
19 ]

```

Ahora con el método GET vamos a observar que se haya ingresado correctamente el usuario.

The screenshot shows the Postman interface with the following details:

- Overview**: GET Untitled R | PUT Projeto A | GET Ver usuario | POST Agregar | PUT Actualiza | DEL Borrar usu | + | No environment
- HTTP**: Projeto AWM / Actualizar usuarios
- Method**: PUT | http://localhost:8000/usuarios/1
- Body**: raw JSON (selected)
- Body Content** (Raw JSON):

```

1  {
2   "id": "1",
3   "nombre": "Jose Avilez",
4   "email": "angelo.conteron@epn.edu.ec",
5   "contra": "12345",
6   "preferencias": {},
7   "intereses": {
8     "tags": []
9   },
10  "notificaciones": {}

```
- Response**: 200 OK | 11 ms | 630 B | Save Response

Con PUT y la dirección URL respectiva para usuarios se ingrese el primer parámetro que es la id para poder actualizar los datos del primer usuario.

The screenshot shows the Postman interface with the following details:

- HTTP**: Projeto AWM / Ver usuarios
- Method**: GET | http://localhost:8000/usuarios
- Params**: (selected)
- Query Params**
- Body**: raw JSON (selected)
- Body Content** (Raw JSON):

```

1  [
2   {
3     "id": "1",
4     "nombre": "Jose Avilez",
5     "email": "angelo.conteron@epn.edu.ec",
6     "contra": "12345",
7     "preferencias": {},
8     "intereses": {
9       "tags": []
10    },
11     "notificaciones": {
12       "alerta24h": true,
13       "alertah": false,
14       "cambios": true,
15       "cercaos": true
16     },
17     "permisoUbicacion": false
18   }

```
- Response**: 200 OK | 5 ms | 1.33 KB | Save Response

Y con el método GET observamos que el usuario se actualizo correctamente el usuario con lo puesto en el body en el método PUT.