

```
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
5 estudiantes = [
6     {'numero_control':1220100603, 'nombre':'ANZO AVALOS MARIA CITLALLI', 'edad':21, 'sexo':'F'},
7     {'numero_control':1219100524, 'nombre':'ARREDONDO GONZALEZ DANIEL ENRIQUE', 'edad':21, 'sexo':'M'},
8     {'numero_control':1220100317, 'nombre':'DUARTE VELAZQUEZ DANIEL', 'edad':21, 'sexo':'M'},
9     {'numero_control':1220100629, 'nombre':'GUTIERREZ MARTINEZ VALERIA IVONNE', 'edad':21, 'sexo':'F'},
10    {'numero_control':1220100632, 'nombre':'LUNA CANTERO ANGEL IVAN', 'edad':21, 'sexo':'M'},
11    {'numero_control':1220100209, 'nombre':'MARTINEZ RAMIREZ GUADALUPE MONSERRAT', 'edad':21, 'sexo':'F'},
12    {'numero_control':1220100053, 'nombre':'REYES MORALES SALVADOR', 'edad':21, 'sexo':'M'},
13    {'numero_control':1220100597, 'nombre':'SALAZAR LEON MARIA GUADALUPE', 'edad':21, 'sexo':'F'},
14    {'numero_control':1220100596, 'nombre':'TADEO MARTINEZ ALEJANDRO', 'edad':21, 'sexo':'M'},
15    {'numero_control':1220100075, 'nombre':'TORRES GARCIA JOSE ROGELIO', 'edad':21, 'sexo':'M'},
16    {'numero_control':1220100595, 'nombre':'TRANQUEÑO HERNANDEZ OSCAR ARMANDO', 'edad':21, 'sexo':'M'}
17 ]
```

Método GET – Obtener estudiantes

```
19 @app.get('/estudiante')
20 def get_estudiante():
21     return jsonify(estudiantes), 200
```

Automatización / 07/08/2023 / GET_Estudiante

GET http://127.0.0.1:5000/estudiante Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 54 ms Size: 1.47 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "edad": 21,
4     "nombre": "ANZO AVALOS MARIA CITLALLI",
5     "numero_control": 1220100603,
6     "sexo": "F"
7   },
8   {
9     "edad": 21,
10    "nombre": "ARREDONDO GONZALEZ DANIEL ENRIQUE",
11    "numero_control": 1219100524,
12    "sexo": "M"
13  },
14  {
15    "edad": 21,
16    "nombre": "DUARTE VELAZQUEZ DANIEL",
17    "numero_control": 1220100317,
18    "sexo": "M"
19  },
```

```
23 @app.route('/estudiantes/<int:numero_control>', methods=['GET'])
24 def get_students_id(numero_control):
25     student = next((stu for stu in estudiantes if stu['numero_control'] == numero_control)),
26     if student:
27         return jsonify(student)
28     else:
29         return jsonify({'message': 'El estudiante no fue encontrado'}), 415
```

Automatización / 07/08/2023 / GET_Estudiante

GET http://127.0.0.1:5000/estudiantes/1220100597 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 57 ms Size: 290 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "edad": 21,
3   "nombre": "SALAZAR LEON MARIA GUADALUPE",
4   "numero_control": 1220100597,
5   "sexo": "F"
6 }
```

Método POST – Agregar estudiantes

```
31 @app.post('/estudiantes')
32 def add_estudiante():
33     datos = request.get_json()
34     for i in estudiantes:
35         if i['numero_control'] == datos['numero_control']:
36             return {'error': 'Este usuario ya se encuentra registrado'}, 201
37     estudiantes.append(datos)
38     return {'Success': 'Registro agregado con éxito'}, 201
```

Automatización / Add_Estudiante

POST http://127.0.0.1:5000/estudiantes Send

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

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "numero_control": 1220100613,
3   "nombre": "VARGAS LÓPEZ JUAN",
4   "edad": 21,
5   "sexo": "M"
6 }
```

Body Cookies Headers (5) Test Results

Status: 201 CREATED Time: 51 ms Size: 222 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "Success": "Registro agregado con éxito"
3 }
```

Método DELETE – Eliminar estudiantes

```
40 @app.delete('/estudiantes/<numero_control>')
41 def delete_estudiantes(numero_control):
42     data = None
43     for i in estudiantes:
44         if i['numero_control'] == int(numero_control):
45             estudiantes.remove(i)
46             return {'status': 'ok', 'mensaje': 'Estudiante removido'}, 203
47     return {'error': 'Estudiante no encontrado'}, 415
```

Automatización / DELETE_Estudiante

DELETE http://127.0.0.1:5000/estudiantes/1220100603 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 203 NON AUTHORITY INFORMATION Time: 7 ms Size: 249 B Save as Example

Pretty Raw Preview Visualize JSON

```
1
2 "mensaje": "Estudiante removido",
3 "status": "ok"
4
```

Método PUT – Modificar estudiantes

```
49 @app.put('/estudiantes/<int:numero_control>')
50 def update_estudiantes(numero_control):
51     dates = request.json
52
53     for j in estudiantes:
54         if j['numero_control'] == int(dates['numero_control']):
55             j['nombre'] = dates['nombre']
56             j['edad'] = dates['edad']
57             j['sexo'] = dates['sexo']
58             return jsonify({'mensaje': 'Estudiante modificado'}), 200
59     return jsonify({'error': 'Estudiante no encontrado'}), 404
```

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

- URL:** `http://127.0.0.1:5000/estudiantes/1220100596`
- Method:** `PUT`
- Body (Request):**

```
1 {
2   "numero_control": 1220100597,
3   "nombre": "TADEO MARTINEZ ALEJANDRO",
4   "edad": 22,
5   "sexo": "M"
6 }
```
- Body (Response):**

```
1 {
2   "mensaje": "Estudiante modificado"
3 }
```
- Status:** `200 OK`, **Time:** `8 ms`, **Size:** `206 B`