

Alumno: Luna Cantero Angel Ivan.

GRUPO: GRIC3091

Datos

```
'''
Nombre: Luna Cantero Angel Ivan
ejercicio: repaso estudiantes
fecha: 07/08/2023
'''

from flask import Flask, jsonify, request

app = Flask(__name__)

estudiantes = [
    {'numero_control':1220100603, 'area':'TI', 'nombre':'ANZO AVALOS MARIA CITLALLI', 'edad':20},
    {'numero_control':1219100524, 'area':'TI', 'nombre':'ARREDONDO GONZALEZ DANIEL ENRIQUE', 'edad':23},
    {'numero_control':1220100317, 'area':'TI', 'nombre':'DUARTE VELAZQUEZ DANIEL', 'edad':21},
    {'numero_control':1220100629, 'area':'TI', 'nombre':'GUTIERREZ MARTINEZ VALERIA IVONNE', 'edad':21},
    {'numero_control':1220100632, 'area':'TI', 'nombre':'LUNA CANTERO ANGEL IVAN', 'edad':21},
    {'numero_control':1220100209, 'area':'TI', 'nombre':'MARTINEZ RAMIREZ GUADALUPE MONSERRAT', 'edad':21},
    {'numero_control':1220100053, 'area':'TI', 'nombre':'REYES MORALES SALVADOR', 'edad':20},
    {'numero_control':1220100597, 'area':'TI', 'nombre':'SALAZAR LEON MARIA GUADALUPE', 'edad':20},
    {'numero_control':1220100596, 'area':'TI', 'nombre':'TADEO MARTINEZ ALEJANDRO', 'edad':24},
    {'numero_control':1220100075, 'area':'TI', 'nombre':'TORRES GARCIA JOSE ROGELIO', 'edad':21},
    {'numero_control':1220100595, 'area':'TI', 'nombre':'TRANQUEÑO HERNANDEZ OSCAR ARMANDO', 'edad':20}
]
```

Codigo de Visual.

```
@app.get('/estudiantes')
def get_estudiantes():
    return jsonify(estudiantes), 200

@app.get('/estudiantes/<no_control>')
def get_estudiantes_id(no_control):
    for estudiante in estudiantes:
        if estudiante['numero_control'] == int(no_control):
            return jsonify(estudiante)
    return jsonify({'mensaje': 'Estudiante no encontrado.'}), 404

@app.post('/estudiantes')
def add_estudiantes():
    datos= request.get_json()
    for i in estudiantes:
        if i['numero_control'] == datos['numero_control']:
            return {'error':'Numero de contro existente'}, 201
    estudiantes.append(datos)
    return {'succes':'Usuario agregado'}, 201
```

```
@app.delete('/estudiantes/<no_control>')
def delete_estudiante(no_control):
    data = None
    for elem in estudiantes:

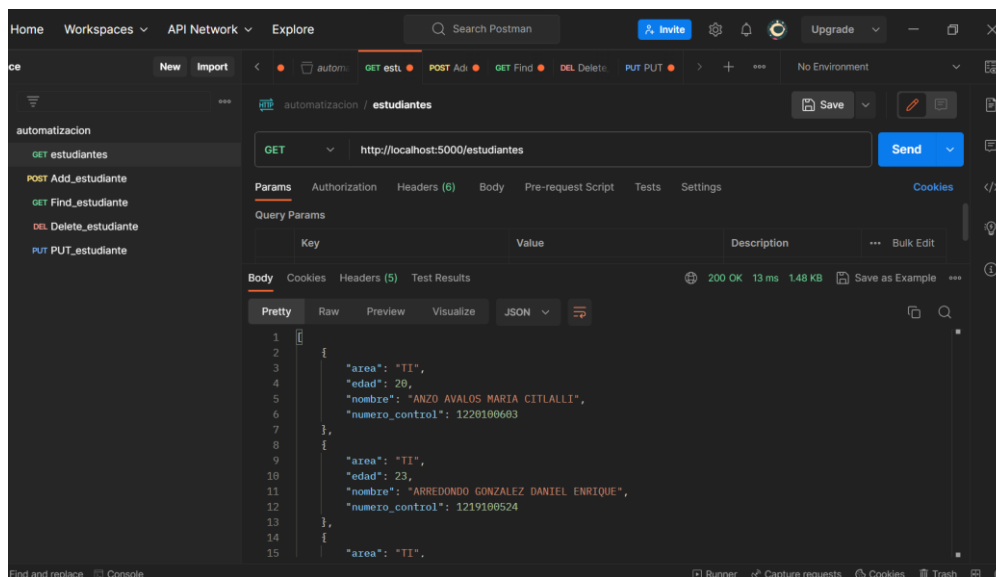
        #Verifica coincidencia de numero de control
        if elem['numero_control'] == int(no_control):
            estudiantes.remove(elem)
            return {'status':'ok', 'mensaje':'El estudiante ha sido removido'}, 200
        #En caso que no se encuentre numero de control.
    return {'error':'Estudiante no encontrado'}, 415
```

```
@app.put('/estudiantes')
def update_estudiante():
    #Extrae el dato del cuerpo de la solicitud
    data = request.get_json()
    for d in estudiantes: #Itera por cada elemento de lista

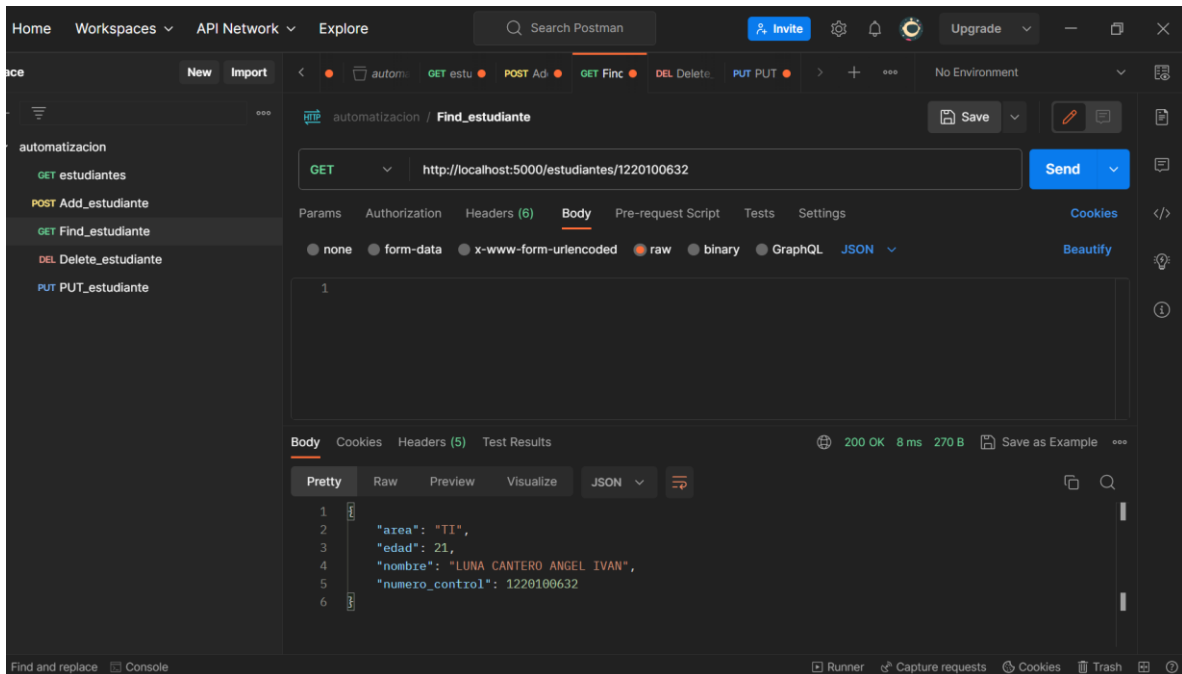
        #Compara los numeros de control
        if d['numero_control'] == int(data['numero_control']):
            d['area'] = data['area'] #modificar la area
            d['nombre'] = data['nombre'] #modificar la nombre
            d['edad'] = data['edad'] #modificar la edad
            return {'status':'ok', 'mensaje':'Se ha actualizado el registro'}, 200
    return {'error':'No se ha encontrado registro de estudiante'}, 415
```

Pruebas en PostMan

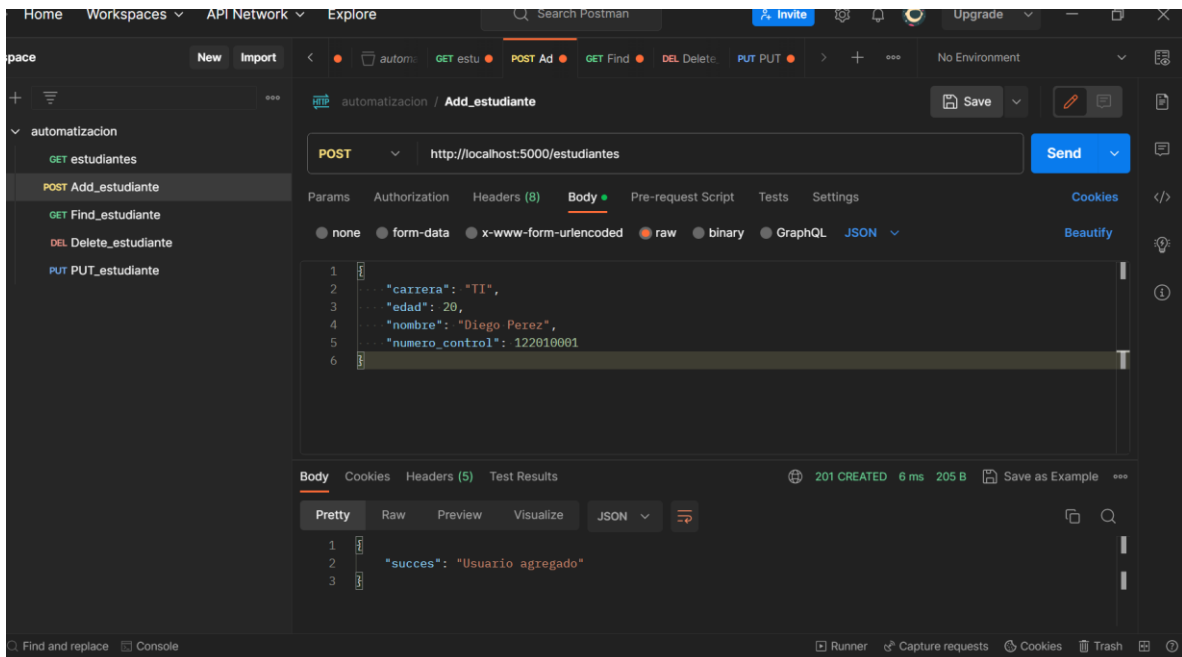
Obtener estudiantes

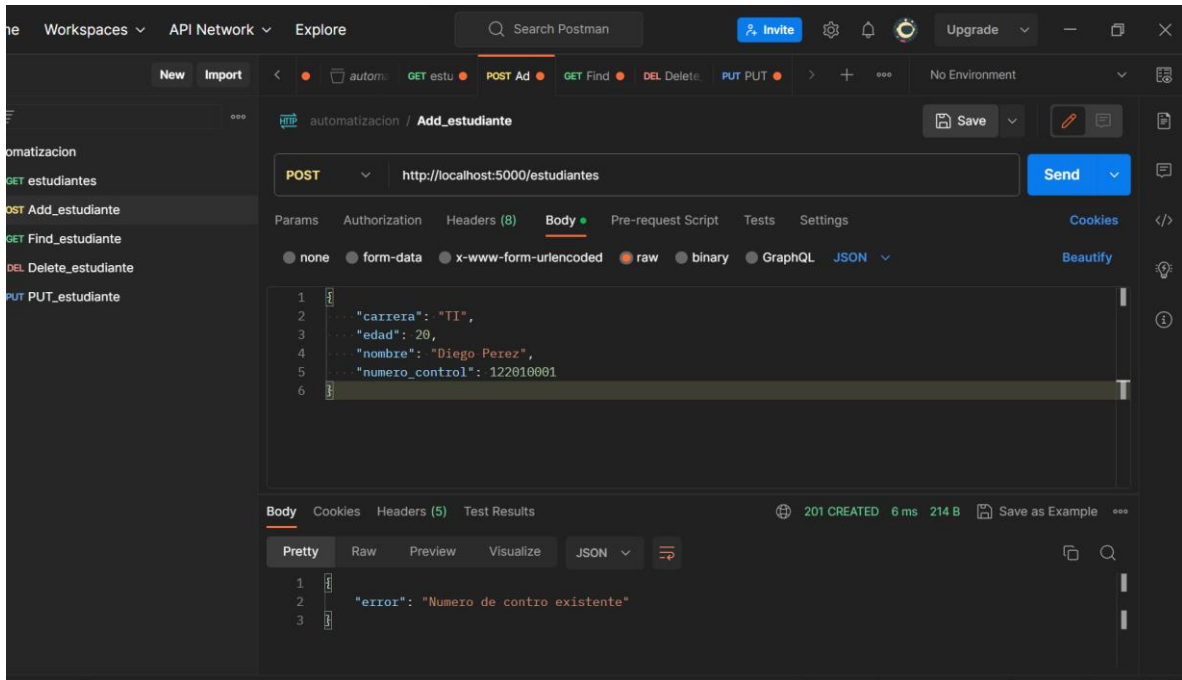


Buscar estudiantes por número de control:

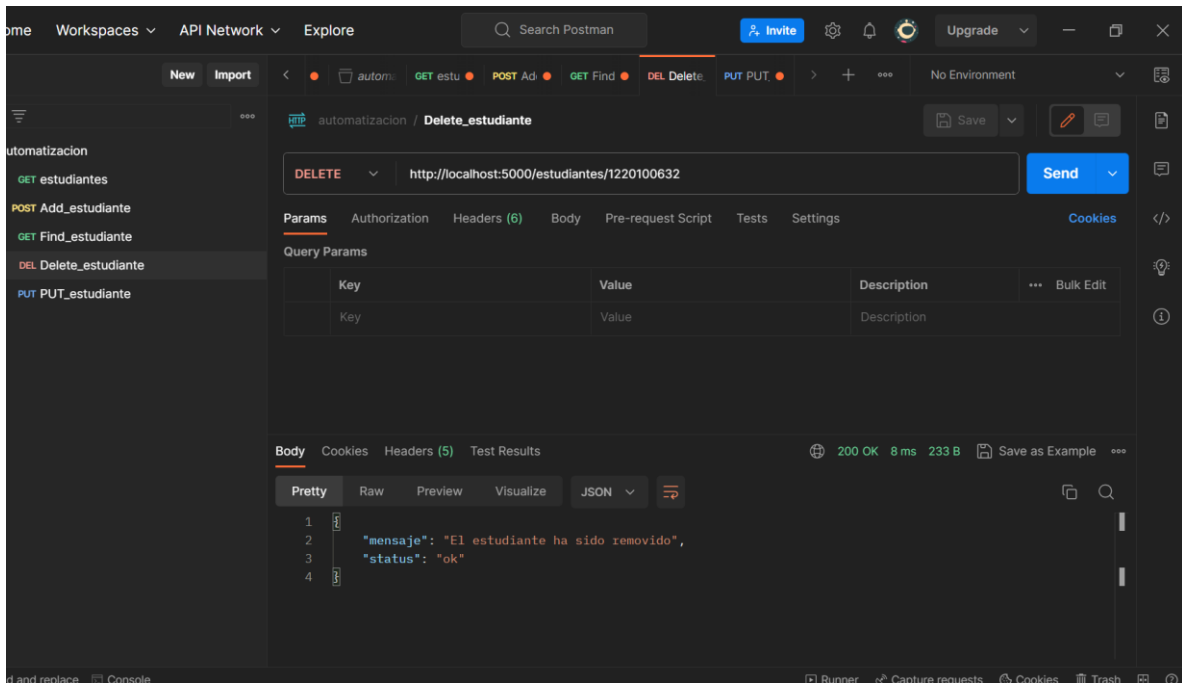


Agregar Usuario.





Eliminar Estudiantes



Modificar Estudiante

The screenshot shows the Postman interface for a PUT request to the endpoint `http://localhost:5000/estudiantes`. The request body is a JSON object with the following fields:

```
{  "numero_control": 1228100053,  "area": "TI",  "nombre": "Jazmin Gonzalez",  "edad": 20}
```

The response status is `200 OK` with a response time of `6 ms` and a body size of `232 B`. The response body is a JSON object:

```
{  "mensaje": "Se ha actualizado el registro",  "status": "ok"}
```

The interface includes a sidebar with a list of API endpoints, a top bar with search and navigation options, and a bottom bar with runner and capture request buttons.