

PARCIAL FINAL

Maicol Libardo Jossa Campaña  
Henry David Rosero

Instituto Tecnológico del Putumayo - ITP  
Tecnología en desarrollo de software  
Lenguaje de cuarta generación

## **Introducción e información general**

a continuación, daremos solución al problema planteado por el docente a manera de parcial final en el que aplicaremos los conocimientos adquiridos en el área de servicios web.

### **Planteamiento del problema:**

Ejercicio del parcial 2 (Asistencia) convertirlo a servicios.

Documentación es de como se utiliza el servicio

Múltiples endpoints

/estudiantes

Este endpoint, se encuentra habilitado con los métodos get, post, put, delete

en el método post, ud podrá crear un estudiante,

necesita enviar los datos identificación, nombres, apellidos, celular,

email, todos los campos son obligatorios.

## Aplicación de conocimientos y proceso de desarrollo:

### 1. PRIMER PASO, REGISTRAR MATERIAS

#### /materias

En este endpoint, se encuentra habilitado con los métodos GET, POST.

- En el método **GET** podremos ver que materias registramos.

Link → <http://127.0.0.1:5000/materias>

Ej:

```
1 = {
2 =   "Materias": [
3 =     {
4 =       "id": 1,
5 =       "nombre": "Base de datos 2",
6 =       "semestre": 5
7 =     },
8 =     {
9 =       "id": 2,
10 =      "nombre": "Lenguaje de 4G",
11 =      "semestre": 5
12 =    }
13 =  ],
14 =  "message": " Materias registradas."
15 = }
```

- En el método **POST** podremos registrar una nueva materia.

Link → <http://127.0.0.1:5000/materias>

<pre>1 = { 2 =   "nombre": "Lenguaje de 4G", 3 =   "semestre": 5 4 = }</pre>	<pre>1 = { 2 =   "Materia": { 3 =     "nombre": "Lenguaje de 4G", 4 =     "semestre": 5 5 =   }, 6 =   "message": "Materia guardada correctamente..." 7 = }</pre>
--	---

### 2. SEGUNDO PASO, REGISTRAR ESTUDIANTES

#### /estudiantes

En este endpoint, se encuentra habilitado con los métodos GET, POST, PUT, DELETE.

- En el método **GET** podremos ver que estudiantes están registrados.

Link → <http://127.0.0.1:5000/estudiantes>

Ej:

```
1 = {
2 =   "estudiantes": [
3 =     {
4 =       "apellido": "Jossa",
5 =       "celular": "3148129949",
6 =       "correo": "maicoljossa@gmail.com",
7 =       "id": 1,
8 =       "identificacion": "1006816052",
9 =       "nombre": "Maicol",
10 =      "semestre": 5
11 =    }
12 =  ],
13 =  "message": " Estudiantes registrados."
14 = }
```

- En el método **POST** podremos inscribir a un nuevo estudiante como en el siguiente ejemplo.

Link → <http://127.0.0.1:5000/estudiantes>

```
{
  "identificacion": "1006815687",
  "nombre": "Henry",
  "apellido": "Rosero",
  "celular": "320999102",
  "correo": "Henry@gmail.com",
  "semestre": 5
}
```

```
1 {
2   "estudiantes": {
3     "apellido": "Rosero",
4     "celular": "320999102",
5     "correo": "Henry@gmail.com",
6     "identificacion": "1006815687",
7     "nombre": "Henry",
8     "semestre": 5
9   },
10  "message": "Estudiante guardado correctamente..."
11 }
```

- En el método **PUT** podremos actualizar la información de un estudiante según su **ID**.  
Link → <http://127.0.0.1:5000/estudiantes/2> el numero al final del link es el ID del estudiante al cual le vamos actualizar la información, si se desea remplazar un solo dato basta con solo ingresar ese único valor y los demás campos dejarlos vacíos.

Ej:

<pre>1 { 2   "identificacion": "", 3   "nombre": "alexander", 4   "apellido": "", 5   "celular": "", 6   "correo": "alexander@gmail.com", 7   "semestre": 6 8 }</pre>	<pre>1 { 2   "estudiante": { 3     "apellido": "Campaña", 4     "celular": "318999102", 5     "correo": "alexander@gmail.com", 6     "identificacion": "68156054", 7     "nombre": "alexander", 8     "semestre": 6 9   }, 10  "message": "Estudiante editado correctamente ..." 11 }</pre>
---	---

- En el método **DELETE** podemos eliminar un estudiante en específico con su respectivo **ID**  
Link → <http://127.0.0.1:5000/estudiantes/2> , igual que con el método PUT al final tendríamos que ingresar el ID del estudiante.

```
1 {
2   "message": "Estudiante eliminado correctamente..."
3 }
```

### 3. TERCER PASO, ASIGNAR UNA MATERIA AL ESTUDIANTE

[/asignar](#)

En este endpoint, se encuentra habilitado con los métodos POST.

- En este método **POST** tendremos que asignar un estudiante con los **ID's** de la materia y del estudiante.

Link → <http://127.0.0.1:5000/asignar>

Ej:

```
{
  "materia_id": 2,
  "estudiante_id": 1
}
```

```
1 {
2   "Materia - Estudiante": {
3     "estudiante_id": 1,
4     "materia_id": 2
5   },
6   "message": "Materia asignada correctamente..."
7 }
```

### 4. CUARTO PASO, CREAR SESIONES

[/sesiones](#)

En este endpoint, se encuentra habilitado con los métodos GET, POST.

- En el método **GET** podremos ver que sesiones tenemos creadas.

Link → <http://127.0.0.1:5000/sesiones>

Ej:

```
1 {
2   "Sesiones": [
3     {
4       "fecha": "2021-07-21",
5       "hora_final": "22:40:00",
6       "hora_inicio": "19:00:00",
7       "id": 1,
8       "materia_id": 2
9     },
10    {
11      "fecha": "2021-08-21",
12      "hora_final": "22:40:00",
13      "hora_inicio": "20:00:00",
14      "id": 2,
15      "materia_id": 2
16    },
17    {
18      "fecha": "2021-08-21",
19      "hora_final": "22:40:00",
20      "hora_inicio": "20:00:00",
21      "id": 3,
22      "materia_id": 2
23    }
24  ],
25  "message": " Sesiones registradas."
26 }
```

- En el método **POST** podremos crear nuevas sesiones.

Link → <http://127.0.0.1:5000/sesiones>

Ej:

```
{
  "materia_id": 2,
  "fecha": "2021-08-21",
  "hora_inicio": "20:00:00",
  "hora_final": "22:40:00"
}
```

```
1 {
2   "Sesiones": {
3     "fecha": "2021-08-21",
4     "hora_final": "22:40:00",
5     "hora_inicio": "20:00:00",
6     "materia_id": 2
7   },
8   "message": "Sesion guardada correctamente..."
9 }
```

## 5. QUINTO PASO, ASISTENCIA ESTUDIANTES

[/asistencia](#)

En este endpoint, se encuentra habilitado con los métodos GET, POST.

- En el método GET podremos ver a todos los estudiantes que asistieron o faltaron a la clase.

Link → <http://127.0.0.1:5000/asistencia>

Ej:

```
1 {
2   "Asistencias": [
3     {
4       "Asistencia": "no",
5       "Fecha de sesion": "2021-07-21",
6       "Hora final": "22:40:00",
7       "Hora inicio": "19:00:00",
8       "Materia": "Lenguaje de 4G",
9       "Nombre": "Maicol"
10    },
11    {
12      "Asistencia": "si",
13      "Fecha de sesion": "2021-08-21",
14      "Hora final": "22:40:00",
15      "Hora inicio": "20:00:00",
16      "Materia": "Lenguaje de 4G",
17      "Nombre": "Maicol"
18    }
19  ],
20  "message": " Asistencias registradas."
21 }
```

- En el método **POST** podremos registrar la asistencia de cada estudiante que asistió a la clase o no.

Link → <http://127.0.0.1:5000/asistencia>

Ej:

```
{  
  "sesion_id": 3,  
  "estudiante_id": 1,  
  "asistencia": "si"  
}
```

```
1 {  
2   "Asistencia": {  
3     "asistencia": "si",  
4     "estudiante_id": 1,  
5     "sesion_id": 3  
6   },  
7   "message": "Asistencia guardada correctamente..."  
8 }
```